Zero-shot classification and Named Entity Recognition for filtering and extracting information from large amounts of unprocessed text

Damian René Zink Andersen
dande20@student.sdu.dk

Marcell Klitgaard Sørensen
marcs20@student.sdu.dk

1 Introduction

Natural Language Processing (NLP) is the domain of analyzing and extracting information from natural languages such as daily human speech. Natural Language has many challenges because of complexities such as context and the lax structure. The paradigm of deep learning in machine learning provides many models and solutions for both old and new NLP tasks and problems. This project explores the practical use of these deep-learning-based models for filtering and extracting useful information from a large amount of natural language text data. The specific use is to filter articles with Text classification and then extract entities of interest using token classification for Named Entity Recognition (NER). For this preliminary data analysis was done, a pre-trained text classification model used for filtering, and transfer learning used to create a NER model for a custom task of recognizing soccer players and teams. The code can be viewed at https://github.com/Holycoffee1337/NLP_EXAM_PROJECT/tree/main/src[1]

2 NLP Tasks

2.1 Text Classification

Text classification is the problem of taking a piece of text and classifying it into one or more classes. The three main types are binary, multi-class, and multi-label. In this project, multi-class is the chosen type. This is because the classification is used as a filter, so the only articles of interest are the ones that can confidently be classified as soccer-related.

Traditionally, text classification using deep-learning models needs to be trained with known labels on annotated data. In this project, the articles of interest are those pertaining to soccer. Labeled data with such granular labeling are not widely available.

[1]The model could not be included due to Git's file size limit.

Therefore a more generalized labeling approach was used. In text classification, a sub-problem is called zero-shot text classification. This problem occurs when the text is to be labeled or classified with new labels not used during training. A solution based on the BERT (Devlin et al., 2018) language model was devised using text entailment (Yin et al., 2019), sometimes referred to as Natural Language Inference. This is based on how humans intuitively categorize or label information. They claim that humans construct some hypothesis and then apply them to the premise, in this context the text to be classified.

2.2 Token Classification

Token classification is the problem of assigning a label or some labels to tokens in a text. The three most popular sub-tasks are Named Entity Recognition (NER), Part-of-speech tagging, and chunking. For this project, NER is used to find and extract entities from text. These models are trained using human-annotated data where words in the text are marked with an entity label from a set of labels. The most common labels are things such as miscellaneous, person, organization, and location. A common standard for the labels is the BIO standard where each token is marked as the beginning of an entity, inside of an entity, or outside an entity. For this project custom labels will be used. This is done to extract more specific types of entities e.g. we are not interested in all persons, just players.

2.3 Transfer Learning

Transfer learning is the practice of using a pre-trained model and changing the output layer or further training it on task-specific data. This makes use of the great amount of general knowledge models can gather from pertaining on large amounts of data and then gaining task-specific accuracy and capability by further training the model on a smaller dataset with task-specific training data.

Further training a model can be computationally costly as all parameters will be updated. A way to minimize the amount of parameters to train is to freeze layers. The pre-trained models commonly have more task-specific learned parameters towards the end of the model. Therefore the early layers can be frozen to save the general language understanding of the model while the later layers can be trained for new tasks or domains.

2.4 Data annotation

To fine-tune the NER model data had to be created. For this, a machine-learning augmented annotation method was used. An initial annotation is done by a zero-shot NER model called GLiNER (Zaratiana et al., 2023). These labels are then manually verified and corrected by humans to create annotated NER data.

3 Results

3.1 Data

The first step of any data processing pipeline or task is to examine the data. For this project, the main dataset used was a Hugging Face dataset SetFit/bbc-news. The dataset contains 2225 articles published by BBC in the time frame of 2004-2005. Looking at the word frequency before filtration (Figure 1), general news terms such as "government", "year", and "time" are used. The Word "said" is much more frequent which is to be expected as news articles often cite statements of involved parties of the story. Whereas after filtration (Figure 2) all the most frequent words are soccer or sport-related, with "said" as an exception expectedly for the same reason as before.

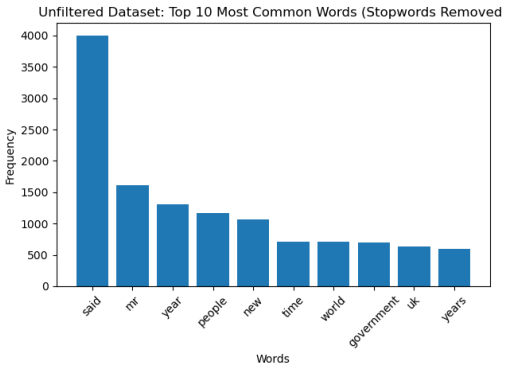

Figure 1: Word Frequency of unfiltered texts

The data used for manual annotation was obtained by searching for soccer news on Google and

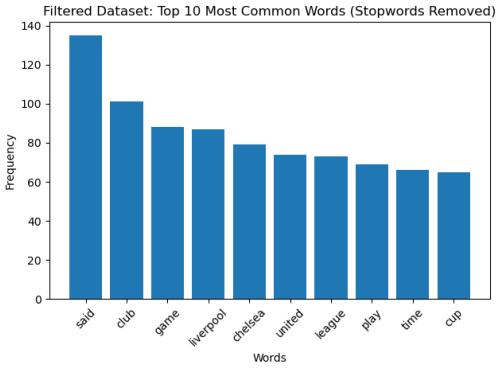

Figure 2: Word Frequency of filtered texts

picking sentences from articles that contained a team name or player name. A few sentences without any entities of interest were added to help with generalizing the data. Around 200 sentences were picked. There is no overlap between this training data and the other data as the training data is contemporary and the the other data is from 2004-2005.

3.2 Text Classification

A zero-shot text classification model from Hugging Face, deberta-v3-large-zeroshot-v2.0, was used with the transformer library for Python. The code was set up so it takes a list of candidates that it passes to the classifier. The classifier gives the probability, which can be understood as confidence, score of each candidate for the given text data. A second list containing the topics of interest, this list must be a subset of the full candidate list, is then used in combination with a threshold of the topics' confidence to filter the articles that are sufficiently likely to contain soccer-related content. In 1 It can be seen which candidates and thresholds are used and an accuracy score was found by manually checking the output of the zero-shot model and marking whether it is soccer-related or not. During this annotation, a significant amount of the non-soccer articles were found to be rugby-related articles. Therefore a more advanced filter was created that contained rugby-related candidates thus removing the likelihood they would be marked as one of the topics of interest. This resulted in higher accuracy and shows that including closely related candidates not of interest can improve the model's accuracy.

3.3 Data Annotation

For data annotation, a zero-shot NER model, GLiNER, was used for the initial annotation. Using the knowledge gained from the zero-shot text classifier an excess of candidate labels was passed to the GLiNER model [2]. Here a threshold was not used to only get the most likely classifications. Instead, only two labels from the full candidate list were saved to a JSON format that could be used to load the data into the manual annotation software Label Studio. The model-annotated data was loaded into the software and manually verified and corrected. The initial annotation using machine learning was implemented to save time on annotating data. To what extent it helped is hard to quantify. The annotator noted that it made annotation easier even with little domain knowledge.

3.4 Transfer Learning

Using transfer learning the plan was to fine-tune an existing pre-trained NER model, dslim/bert-base-NER, to be able to accurately recognize custom entities such as players and teams. This is where the manually annotated data was to be used. Based on the code provided by Tariq Yousef through Google Colab[3] fine-tuning code was implemented. To load the data from the annotated a custom data loader was implemented creating a Dataset object from the data exported from Label Studio.

Layer freezing was not implemented as the technical difficulties of implementing the existing fine-tuner were underestimated. Because of the low amount of data the first 6 layers would have been frozen.

3.5 Token Classification

Using the fine-tuned NER model on the text yields highly questionable results such as "##ikos" showing that a subtoken is classified but not correctly combined to a whole word. The NER model can accept data and classify tokens the correctness and accuracy of the classifications have not been tested.

4 Conclusion

A text classification filter is created using a zero-shot classifier increasing accuracy by including exclusion labels and tuning thresholds. Data was collected and annotated using a machine learning augmented method in which a zero-shot token classifier was used for initial classification, then followed by manual review and correction. The annotated data was used to further train a pre-trained model for NER with custom labels. The model can run but correctness has not been tested. A comprehensive combination of the methods implemented here is lacking due to technical difficulties with implementing fine-tuning code using the export data from Label Studio.

[2]The labels used can be seen in the code on the GitHub

[3]A link is provided in the relevant code.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*.

appendix

Table 1: Zero-Shot Classifier Results with Different Thresholds

Threshold Configuration	**Articles Kept**	**Accuracy (%)**
Threshold 50	54	64.08
Threshold 70	142	61.11
Threshold 50 (Advanced)	87	97.70

Candidates Used:

- For Threshold 50 and Threshold 70:
  - **Topics:** *Soccer, Football*
  - **Candidates:** *Soccer, Football, Technology, Business*
- For Threshold 50 (Advanced):
  - **Topics:** *Soccer, Football*
  - **Candidates:** *Soccer, Football, Association Football, Rugby, Rugby Football, Technology, Business*

Labels Looked For:

- Soccer-related labels: *Soccer, Football, Association Football*
- Rugby-related labels: *Rugby, Rugby Football*

Note: The dataset initially contains 2100 articles. Filtering was applied based on the defined thresholds, topics, candidates, and labels.