Event Detection – Homework 1

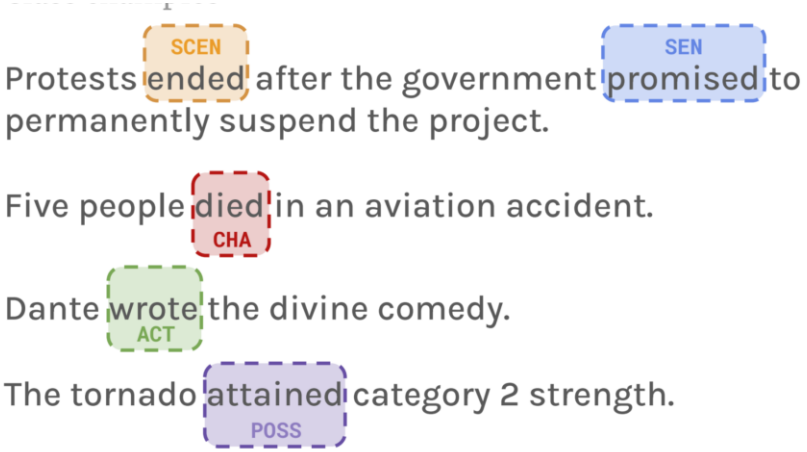
Multilingual Natural Language Processing

SAPIENZA
UNIVERSITÀ DI ROMA

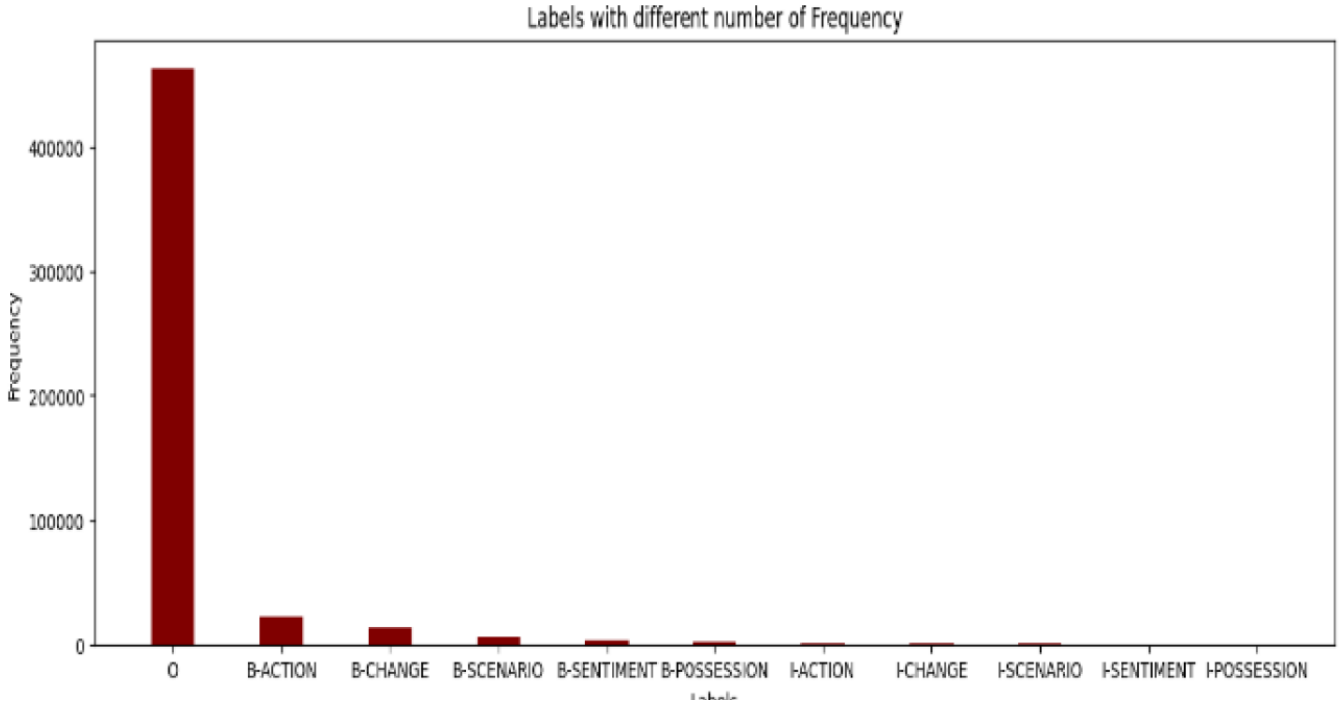
Presented To : Prof. Roberto Navigli
Presented By : Sameer Ahmed

Event Detection

- Event detection (ED) is a task in Natural Language Processing (NLP) that aims to identify event triggers and categorize events mentioned in the text.
- These event detection labels are based on the BIO format. The total number of labels according to BIO format is (B-Sentiment, B-Scenario, B-Change, B-Possession, B-Action, I-Sentiment, I-Scenario, I-Change, I-Possession, I-Action, and O).

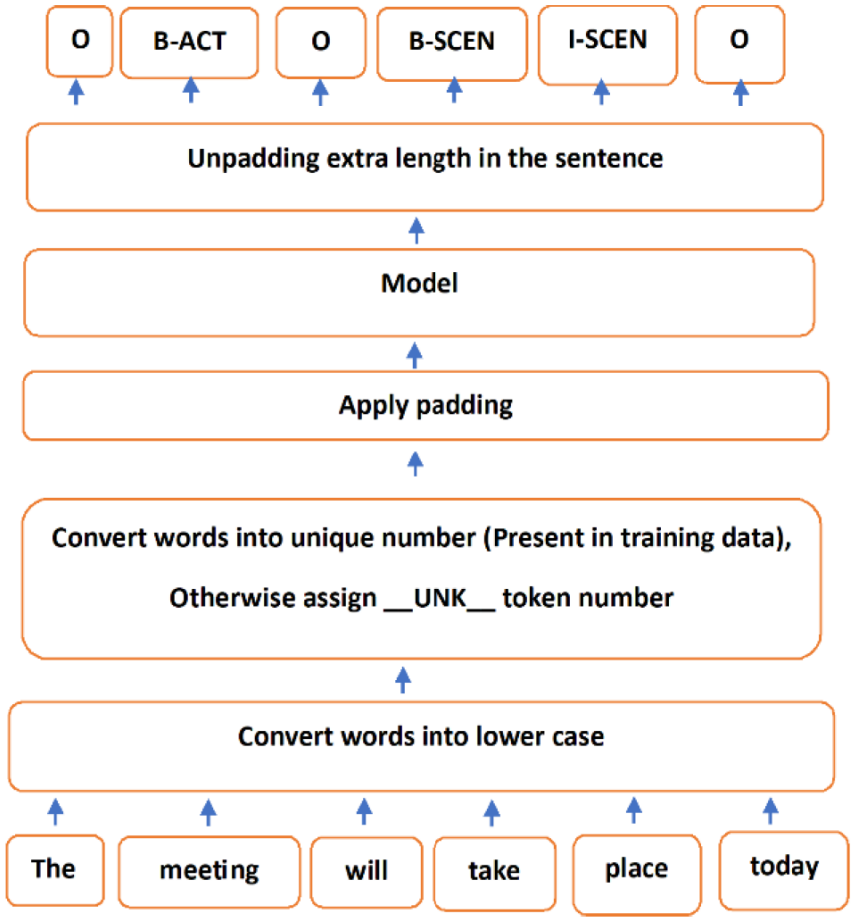


Data Imbalance



Flow Diagram

This flow diagram illustrates the complete process of the event detection task.



Preprocessing Data

Our dataset is not in the form that we give to our model. We need to preprocess our training, testing and development (Validation) data.

1. The first step is to convert (training, development, and testing) tokens into lowercase because the capitalized forms of the words will give different embeddings from the lowercased forms.
2. Then convert (training, testing, and development) tokens and labels into numbers. These tokens and labels need to be converted by taking training data (tokens and labels) unique values.
3. All tokens and labels are not the same length. So, for that, we need to apply padding.

Model Approach

In this task, I am considering three different approaches:

- Bidirectional LSTM (BiLSTM) without pretrained word embedding
- Bidirectional LSTM (BiLSTM) with pre-trained word embedding
- Bidirectional LSTM with Conditional random field (CRF)

Bidirectional LSTM (BiLSTM) without pretrained word embedding

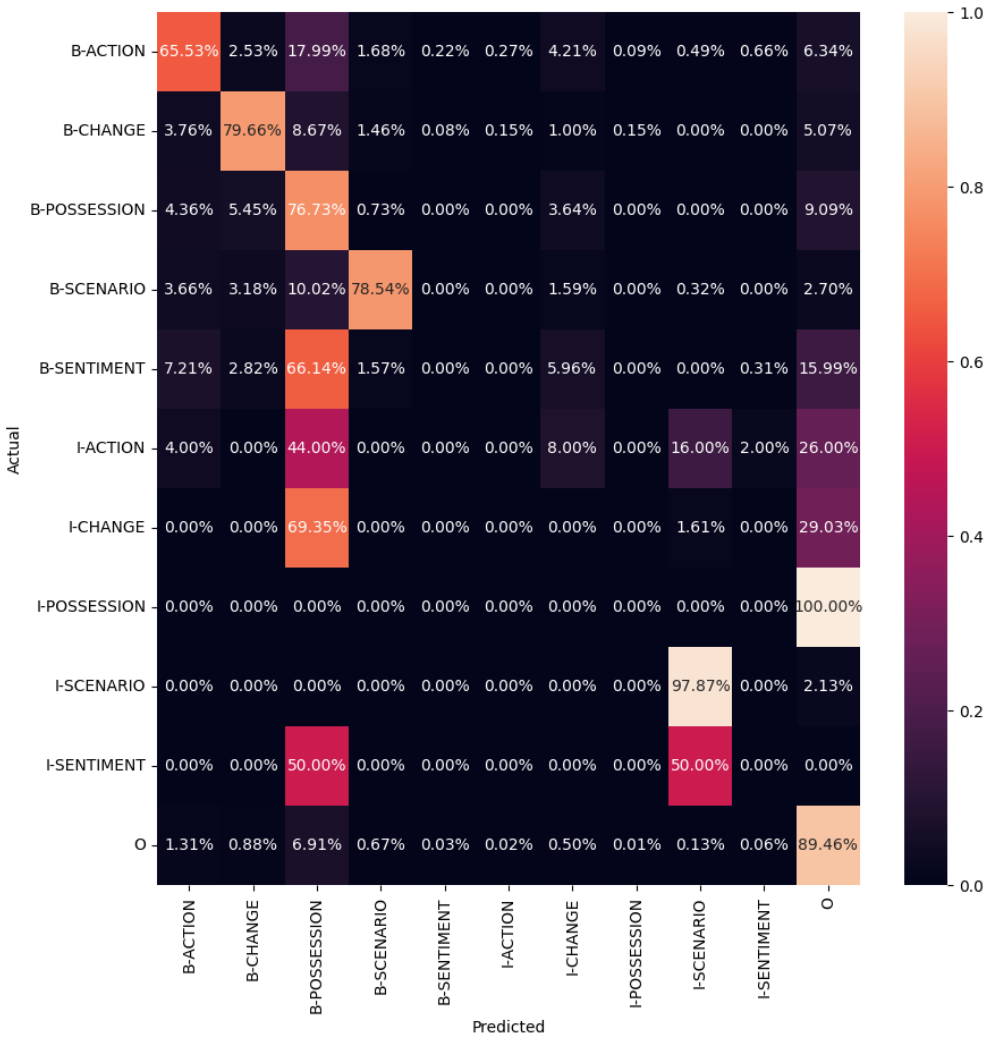
The first approach I have chosen for event detection is Bi-LSTM, and I selected it for this task because of its effectiveness in capturing sequential dependencies and contextual information, which are crucial in accurately identifying and classifying events.

Here I am using without pre-trained word embedding model.

The model arhitecture look like:

```
MODEL_1(
  (embedding_layer): Embedding(33081, 300, padding_idx=0)
  (lstm): LSTM(300, 128, batch_first=True, bidirectional=True)
  (dropout1): Dropout(p=0.4, inplace=False)
  (fc1): Linear(in_features=256, out_features=32, bias=True)
  (dropout2): Dropout(p=0.4, inplace=False)
  (fc2): Linear(in_features=32, out_features=12, bias=True)
  (softmax): Softmax(dim=1)
)
```

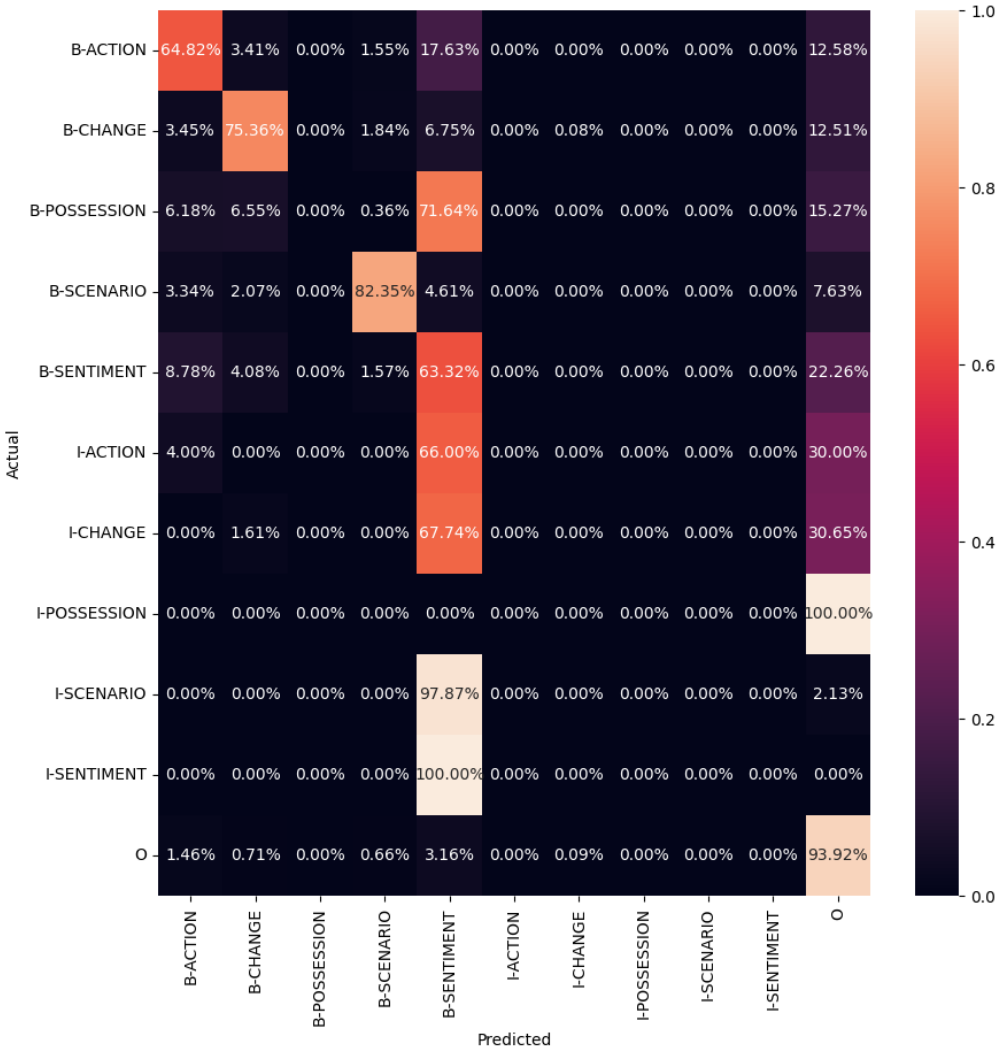
Confusion Matrix



Bidirectional LSTM (BiLSTM) with pre-trained word embedding

- Previous Model (BI-LSTM model without pretrained embeddings) may have limited ability to capture the contextual information present in the text. It relies solely on the training data to learn word representations, which might not be sufficient for effectively understanding and representing the complexities of the language.
- So for that, here the 2nd approach is (BI-LSTM model with pretrained embeddings). Pre-trained word embeddings provide a way to capture semantic and contextual information from large amounts of unlabeled text data. By leveraging pre-trained word embeddings, we can benefit from the knowledge and patterns learned from extensive training on vast text corpora. This can help improve the representation of words and their meanings within the event detection task.
- Here, I am using the same architecture of Bi-LSTM with 'GloVe' pre-trained word embeddings (300-dimensional vectors) for this task.

Confusion Matrix



Bidirectional LSTM with Conditional random field (CRF)

- The BiLSTM component of the model is responsible for capturing contextual information and dependencies in the sequence, which helps in making predictions for each word individually. However, it does not consider the global context or the relationships between neighboring words during the labeling process. As a result, it may lead to inconsistent classifications for the same word.
- Adding a Conditional Random Field (CRF) layer to the Bidirectional LSTM (BiLSTM) model can help resolve the issue of inconsistent word classifications. In event detection or any sequence labeling task, it is common for certain words to have different labels depending on the context or neighboring words.

{"idx": 18461,

"tokens": ["The", "storm", "was", "responsible", "for", "one", "death", "and", "$", "100,000", "in", "damage", ",", "mostly", "in", "North", "Carolina", "."],

"labels": ["O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O"]}

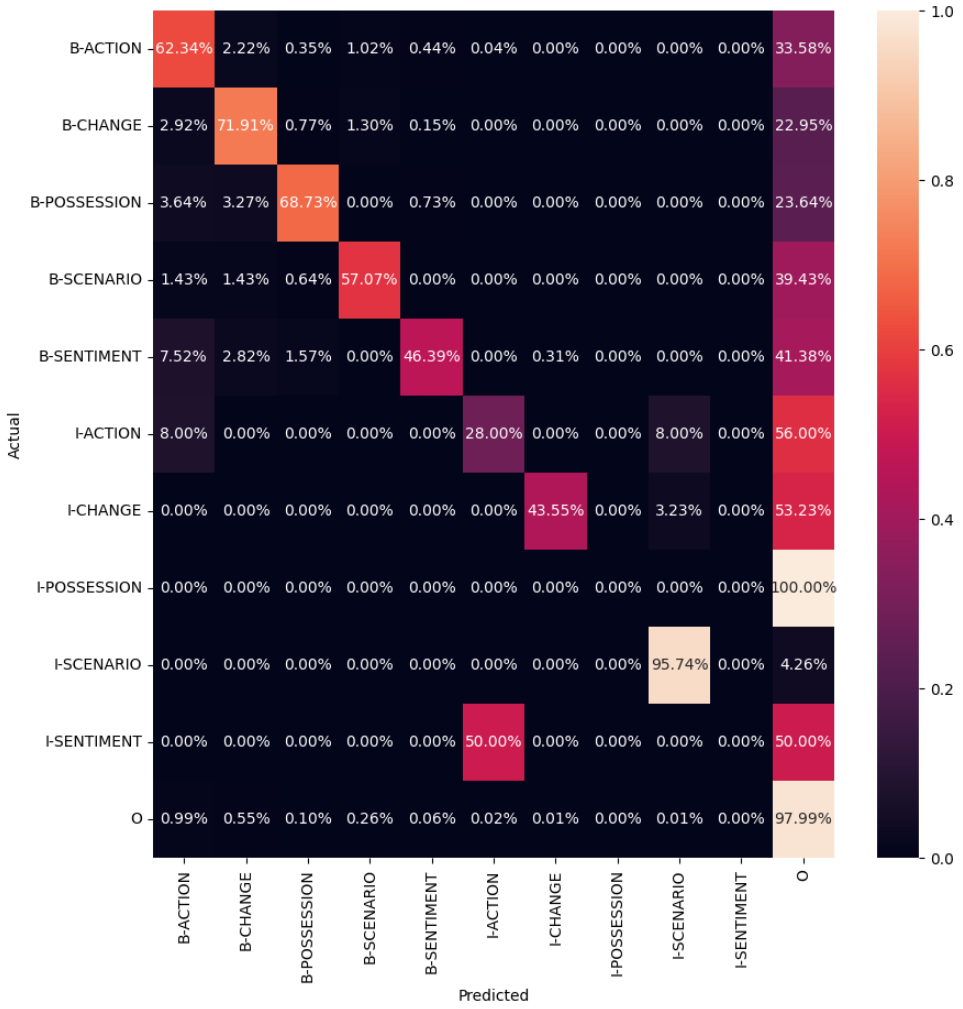
{"idx": 18494,

"tokens": ["The", "``", "Luftwaffe", "''", "formations", "were", "dispersed", "by", "a", "large", "cloud", "base", "and", "failed", "to", "inflict", "severe", "damage", "on", "the", "city", "of", "London", "."],

"labels": ["O", "O", "O", "O", "O", "O", "B-CHANGE", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "O", "B-CHANGE", "O", "O", "O", "O", "O", "O"]

}

Confusion Matrix



Results

	Epoch	F-1 score (macro)	Accuracy
BiLSTM with pretrained word embedding	30	0.39	87 .2%
BiLSTM with pretrained word embedding	25	0.42	91.0%
BiLSTM with CRF	**2**	**0.66**	**94.6%**
BiLSTM with CRF	3	0.63	94.4%

THANK YOU FOR YOUR KIND ATTENTION