Report for task-2: NERMuD

Leonardo Colosi
Sapienza Università di Roma
1799057
colosi.1799057@studenti.uniroma1.it

1 Dataset Description

The original dataset was intended to perform extraction and classification of named entities, such as persons, organizations, and locations, on multidomain documents. More information about the task can be found on the official website. The texts used are taken from: Wikinews (WN), some Italian fiction books (FIC) and the writings and speeches of Alcide De Gasperi (ADG). The provided dataset is partitioned in three datasets: "ADG", "FIC", and "WN", already split in train, dev and test.

2 Format

The data is annotated using the Inside-Outside-Begining (IOB) tagging scheme, which subdivides the elements of every entity as begin-of-entity (B-ent) or continuation-of-entity (I-ent). Person are marked with PER, organizations with ORG, and locations with LOC. Fields are tab separated and, as claimed on the website, sentences are empty line separated. An example is reported in figure [1].

```
Silvio  B-PER
Berlusconi  I-PER
e O
Mario B-PER
Monti I-PER
a O
Milano  B-LOC
```

Figure 1: A random sample extracted from the WN partition of the dataset.

3 Methodology

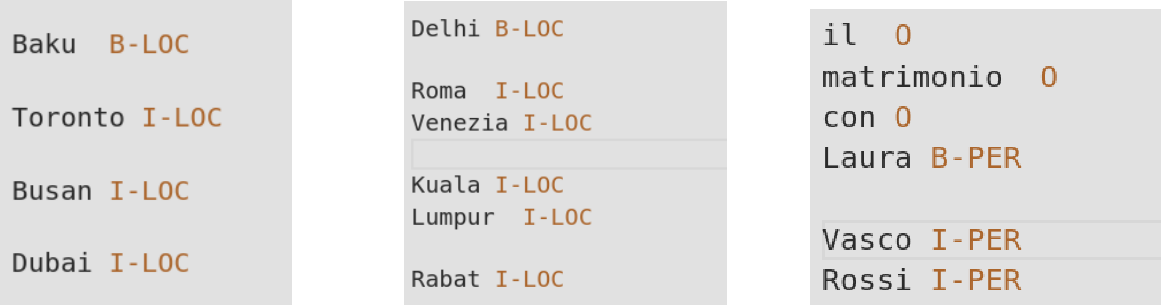
In order to produce the desired samples structure some required actions has been performed on the partition of the original dataset. Since the separation of the sentences in the original dataset is not consistent, the manipulation applied required to go over all the different possibilities. In particular there are two cases in which a sentences could be considered terminated. The first one is when a termination token such as "." or ";" is found. The second one is when an empty line is preceded by a token that is either a word or a termination token. The case in which is a word represent the separation between what has to be considered as a title and the related text following the title. Only three exception to this conditions exists in the original dataset and are all inside the test split of WN partition. Those has been handled separately (see Figure[3]). By performing this handcrafted split it was possible to achieve a better precision in the sentence separation, avoiding the case in which two chuck of text separated by an empty line, but conceptually linked by a ":" character would end up belonging to two different sentences. An important thing to notice is the mapping between the original tagging and the generated choices as can be seen in the following table [1]. The final output is are *jsonl* file for each split and partition, those file contains the samples in *json* format [2].

Original Tag	Choice
PER	Persona
ORG	Organizzazione
LOC	Luogo

Table 1: Mapping between original tag and possible entries of the choices list.

```
{
    "sentence_id": int, # an incremental integer (starting from zero)
    "text": str, # the input sentence,
    "target_entity": str,
    "choices": List[str],
    "label": int, # the correct answer
}
```

Figure 2: Standard template for a json object in the dataset.



```
Baku   B-LOC

Toronto I-LOC

Busan I-LOC

Dubai I-LOC
```

(a) In this case (bullet point list) all the tokens belongs to the same sentence and represent 3 different entities so they have been re-tagged with the format B-xxx.

```
Delhi B-LOC

Roma  I-LOC
Venezia I-LOC

Kuala I-LOC
Lumpur  I-LOC

Rabat I-LOC
```

(b) This case is completely analogue to case 3a so it has been handled in the same way.

```
il  O
matrimonio  O
con O
Laura B-PER

Vasco I-PER
Rossi I-PER
```

(c) In this case the named entity "Vasco Rossi" is tagged as a internal to the entity "Laura". Also in this case the two entities has been re-tagged as separate entities.

Figure 3

4 Prompts

In the context of this task three different styles of prompt has been used. The first one consisting in a a simple straight forward question where the model provided with a text containing a target name entity, and is asked to identify such entity given a set of options to chose from. The second style is specific with respect to the partition of the dataset from which the sample came from. This means that not only the text but also additional context is provided to the model. even in this case the prompts contain a list of possible option for the classification of the entity. The final one can be consider as free text generation prompt style. In this case the model is directly ask to classify the named entity in the text without any constrain on the possible classification options.

5 How to run

In order to execute the code it is required to install some standard python libraries which are listed in the *requirement.txt* file. After this initial set-up will be possible to directly run the code inside the python file: *task_2.py*. This script will automatically download the original dataset from the online source directly into a *data/* directory. All the formatted partition of the new dataset will be saved into a *results/* directory, after the termination of the execution. It is also possible to test the correct generation of prompts by executing *prompt.py*. Run this script is by using the *-d* option, provide a number between 1 and 3 to identify the dataset partition. The code will then chose random samples, from the newly generated train test and development splits, to be substituted inside the prompts templates.