CS6320 Fall 2022: Final

Due: Dec 16, 11:59pm

Report page limit: 6 pages, you are encouraged to make the report brief. You need approval from the TA to have extra pages for the report, via emails.

Work on this project is individual. No group work is allowed. Your goal in the final is to develop a name-entity recognizer.

GitHub Classroom Setup Follow the *starter repository invitation* link. GitHub Classroom will provide a link to your private repository for you to clone to your computer. All team code should exist in the repository. *Only the code that is present in the master branch of your repository by the due date will be evaluated as part of your submission!.*

Starter Repository (Individual):

`https://classroom.github.com/a/Q52PvC39`

Leaderboard:

`https://www.kaggle.com/competitions/cs6320final`, maximum daily submissions is 10 times (kaggle constraint). After submission, your model's test set F1 score will be published/updated on the leaderboard.

Report Template (including Rubrics):

`https://www.overleaf.com/read/swknmtkffvnx`

Data Download:

eLearning. After downloading, please refer to the notebook for more details.

Introduction In this project, you will implement a model that identifies named entities in text and tags them with the appropriate label. Particularly, the task of this project is Named Entity Recognition. A primer on this task is provided further on. The given dataset is a modified version of the CoNLL-2003 [1] dataset. Please use the datasets that we have released to you instead of versions found online as we have made simplifications to the dataset for your benefit. Your task is to develop NLP models to identify these named entities automatically. We will treat this as a **sequence-tagging task**: for each token in the input text, assign one of the following 5 labels: ORG (Organization), PER (Person), LOC (Location), MISC (Miscellaneous), and O (Not Named Entity). More information about the dataset is provided in the ipython notebook.

Named Entity Recognition Let us now take a look at the task at hand: Named Entity Recognition (NER). This section provides a brief introduction to the task and why it is important. What is NER? As we've covered in the lecture, NER refers to the information extraction technique of identifying and categorizing key information about entities within textual data. In the example, we can see that the text has multiple named entities that can be categorized as LOC (location), ORG (organization), PER (person), etc.

Input:	ZIFA	said	Renate	Goetschl	of	Austria	won	the	World	Cup	down	hill
	race	in	Germany	.								
Output:	ORG	O	PER	PER	O	LOC	O	O	MISC	MISC	O	O
	O	O	LOC	O								

Your system is expected the generate the outputs like above, where several named entities are extracted.

Implementation For this project, you will implement **any approach at your will**, which is covered in the lecture (e.g. HMM, MEMM with feature engineering, RNN, pretrained-language model based approaches)[1] and even not limited to machine learning techniques. You may use any public libraries and resources (e.g. `nltk, nltk.classify, math, numpy, operator, sklearn, torch, transformers`[2]). Make clear which parts were implemented from scratch vs. obtained via existing packages or resources. You may use any publicly available data, including additional training data and taxonomies (e.g., lists of named entities) if needed.

Data and Evaluation Code Please refer to details in the notebook.

Performance Grading Part of the grading will based on your test performance on the kaggle leaderboard. For details on submitting to leadearboard, please refer to the notebook. In the report, you will add a screenshot of your entry of leaderboard performance.

Submissions Guidelines

- Update your code in the github repo.
- Submit your report to Gradescope. The report should be brief.

Posting of this final on public or private forums, services, and other forms of distribution is not allowed. **We thank Heather Zheng for the data and inspiration.**

References

[1] Erik Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[1] Hint: for linear models, you can try do feature engineering with stemming and what we covered the lectures. For unknown word handling, you can refer to previous assignments and what we have covered in the lectures.

[2] E.g. https://github.com/huggingface/notebooks/tree/main/examples