**Assignment 3:**

**Fine-grained Entity Typing**

Last modified: October 16, 2022

1. **Overview**

In this assignment, you will develop a fine-grained entity typing system. Given a list of extracted entity mentions, you will need to classify each mention into one or more of the types defined in an ontology. The ontology consists of approximately 200 fine-grained entity types that are representative of news data.

1. **Introduction**

a. Entity Extraction

Entity extraction (named entity recognition) is an information extraction task that seeks to locate and classify named entities mentioned in unstructured text into predefined categories (e.g., person names, organizations, locations). For example, let’s consider the following simple sentence:

“Bill Gates founded Microsoft in 1975”

Then an entity extraction system needs to produce the following annotation:

“[Bill Gates]PERSON founded [Microsoft]ORGANIZATION in [1975]DATE”

We encourage you to try out [AllenNLP’s NER demo](https://demo.allennlp.org/named-entity-recognition/named-entity-recognition) to see more examples of the task.

Note that entity extraction has two sub-tasks: the “localization” and “classification” steps. **In this assignment, you will need to only focus on the “classification” step (also known as entity typing)**. We will give you a list of entity mentions extracted from some documents. And your task is to determine the type of each mention.

For example, let’s suppose your system needs to process the simple sentence shown above. Then, your system will be given the mentions “Bill Gates”, “Microsoft”, and “1975”; and it will need to determine the types of the mentions (i.e., PERSON, ORGANIZATION, and DATE).

**b. Recognizing Ultra Fine-grained EntitieS (RUFES)**

[Recognizing Ultra Fine-grained EntitieS (RUFES)](https://tac.nist.gov/2020/KBP/RUFES/index.html) is a shared task that extends entity extraction to a new fine-grained entity ontology that consists of approximately 200 fine-grained entity types that are representative of news data. Each entity type in the ontology has a one-sentence definition along with some examples. Some sample types can be seen in the table below.



Given an input document, the system is required to automatically identify an entity as a cluster of name, nominal, and/or pronominal mentions, and classify the entity into one or more of the types defined in the ontology.

In fact, the full RUFES task can be divided into 3 sub-tasks: mention extraction, coreference resolution, and entity typing. Again, **in this assignment, you will need to only focus on the entity typing subtask.**

1. **Data**

We will be using the data provided by [RUFES 2020](https://tac.nist.gov/2020/KBP/RUFES/index.html). You may want to visit the website and also read the [task overview paper](https://blender.cs.illinois.edu/paper/kbp2020overview.pdf) to get some familiarity with the task.

* For training, we have [50 annotated documents](https://drive.google.com/file/d/1h_GJfELg9EdJRAiJFRHwtRSrNz_Pi43z/view?usp=sharing) provided by RUFES 2020.
* For testing, we have [106 annotated documents](https://drive.google.com/file/d/1vnbqjJyGKXPJFEGxQZxvNvq7NE2ez1E3/view?usp=sharing).

Please refer to the readme.txt files in the downloaded zip files for more information.

For building an entity-typing system, you will likely need to pay attention the most to the folder data/ltf and the tab file in the folder annotation:

* LTF files consist of tokenized documents. A simple script for reading ltf files is available at: <https://github.com/wilburOne/ACE_ERE_Scripts/blob/master/ltf2sent.py>
* The tab file consists of ground-truth annotations.

Note that the data provided by RUFES 2020 is not that large. So you may also want to make use of external data for fine-grained typing. For example, refer to Assignment #1 of one of our previous courses: <http://blender.cs.illinois.edu/course/fall20/ie.html>

We also run our mention extraction system on the test set. Here is the [file](https://drive.google.com/file/d/1PWhr-8Q2JS11VYyVWoMf6gjY_dTEKxAQ/view?usp=sharing) containing the mention extraction results. The file is a tab-separated file. Each row corresponds to one extracted mention:

* The first column is the mention string.
* The second column is the mention justification.
* The third column is the mention type: “NAM” (for name mentions), “NOM” (for nominal mentions), or “PRO” (for pronominal mentions).

Finally, the ontology of RUFES 2020 is available at <https://tac.nist.gov/2020/KBP/RUFES/guidelines/RUFES2020OntologyV1.1.xlsx>

1. **Assignment Instructions**

We recommend you to follow the following steps to finish the assignment:

1. Get familiar with the RUFES data files mentioned in Section 3.
2. Get familiar with the scorer <https://github.com/shahraj81/rufes>. The author of the scorer also provided a sample output file. Please refer to it as well since your final prediction file needs to have the same format.
3. Implement and train a fine-grained entity typing system using the training data provided by RUFES 2020.
	1. Refer to Section 5 for some code of baseline models.
	2. Please use the **gold mentions** in the tab file of the training set for training
	3. You may want to leverage some additional data (mentioned in Section 3) for better performance
4. Evaluate the performance of your system on the test set.
	1. You need to use the **extracted mentions** that we provided. In real-world applications, gold mentions are typically not given.
	2. Generate a prediction file and use the [official scorer](https://github.com/shahraj81/rufes) to compute a score.\
5. Submit your prediction file to an official leaderboard provided by NIST.
6. **Helpful Resources**
* A script for reading the LTF files: <https://github.com/wilburOne/ACE_ERE_Scripts/blob/master/ltf2sent.py>
* RUFES type scorer: <https://github.com/shahraj81/rufes>
* An example LSTM-based model for entity typing: <https://drive.google.com/file/d/1jfVpUyoRheTlXj54zvrzmhBH66kTf_Uf/view?usp=sharing>
* Other tools for entity recognition and typing:
	+ Fine-grained entity recognition: <https://github.com/xiaoling/figer>
	+ Ultra-fine grained entity typing: <https://homes.cs.washington.edu/~eunsol/open_entity.html>
	+ Neural Entity Typing with Knowledge Attention: <https://github.com/thunlp/KNET>
1. **Submission**

Please name your submission as `netid\_assignment2.zip`, including:

* A report describing your methods, results, and findings.
* The code. The code should include a README.md with the environment and running instructions and at least a “predictions.tab” file for your predictions on the test set.
1. **Grading**

1. (10pt) Build a fine-grained entity typing system.

* [5 pt] Code for training the system on the training set of 50 documents.
* [3 pt] Code for generating the predictions on the test set of 106 documents using our **provided extracted mentions**.
* [1 pt] A valid “predictions.tab” file that has the predictions on the test set. Basically, we need to be able to directly evaluation your file using the scorer: <https://github.com/shahraj81/rufes>
* [1pt] Submit your prediction file to an official leaderboard provided by NIST.

2. (2pt) Performance Ranking:

2. (2pt) Write a clear and informative report about your methods, results, and findings.

* [Error analysis] Please also look into the prediction errors and analyze the possible reasons for the wrong predictions (and also think of how to improve the model).

3. (1pt) Write a clear README of the submitted code, including the environment and running instructions.

Bonus

4. (2pt) Implement a new idea to improve your system. For example, one potential idea is to first train your system using a larger typing dataset provided in Assignment #1 of <http://blender.cs.illinois.edu/course/fall20/ie.html>. After that, you can fine-tune on the training set of RUFES 2020.