Homework 8:
Type Prediction

Dr. Benjamin Roth
Computerlinguistische Anwendungen

Due: Wednesday July 3, 2019, 14:00

In this homework you will implement type prediction for entities using pre-trained word vectors. Download the file `enitities_types.tar.bz2` from the lecture homepage, and unpack it into your `src/data` directory. You can check your progress using unit tests:

```
python3 -m unittest -v hw08_entity_types/test_entity_types.py
```

Exercise 1: Reading word vectors [4 points]

Complete the function `read_word2vec_file(filename)` in the file `utils.py`. It reads a word vectors file (such as `word_vectors.txt`) and returns a matrix, stored in a Numpy array (rows: words, columns: features). The first line of a word vectors file contains the number of rows and columns (white-space separated). All following lines contain a word and its features (again, white-space separated).

Exercise 2: Representing Entities [4 points]

The task of this exercise sheet is to predict types for named entities.
So, for example the entity: **Office of the Vice President**
... can be characterized with two types: **government_agency** and **organization**

The function `read_entity_types_file` in `utils.py` reads entities and types, and encodes entities by the average of their word vectors, and types as a 1-0 matrix. Read the docstring of `read_entity_types_file`, and understand how it is meant to work.

Your tasks is to complete the part which computes the average of the word vectors for an entity. If an entity contains tokens, which are not in the embedding matrix, ignore those tokens. If none of the tokens of the entity are in the embedding matrix, use the vector with all 0 for that entity.

Exercise 3: Predicting Types [4 points]

Complete the function `train_evaluate_type_prediction` in the file `predict_types.py`. Your need to solve two tasks:

1. Train a multi-label classifier (using the training data), and predict the labels (for the test data). Use OneVsRest classification and LogisticRegression (**not** SVC).
2. Compute and return Precision, Recall and F-Score for your prediction. Remember, how precision and recall are computed [1], and how to compute the f-score from that.

If you are confident that your implementation is working, you can test it with the large training and test file (may take between 5 and 10 minutes):

```
python3 -m unittest -v hw08_entity_types/test_entity_types_large.py
```

[1] `https://en.wikipedia.org/wiki/Precision_and_recall`