Coursework specification: 70016 - NLP (2021)

Your task will be to develop a regression model to predict how funny an edited news headline is after one word in the headline has been replaced.

You should choose to do **one of the following two tasks**:

- **Task 1 (Regression)**: Predict the score for how funny the edited headline is
- **Task 2 (Predict the funnier headline)**: Predict which of two edited headlines is funnier

For the task you choose, you are required to address the task using two different approaches:

- **Approach 1**: Where you are allowed to use pre-trained representations or models
- **Approach 2**: No pre-trained representations or models allowed

More information about the two tasks can be found here: https://competitions.codalab.org/competitions/20970.

Coding instructions: You should use the ***train***, ***eval*** and ***model_performance*** functions provided for you, although you may edit these as necessary. We ask you to work with these functions to discourage copy/pasting model code from other sources. You may change the remaining code as much as required. You can also use any external libraries where needed.

Here are your initial notebooks for Task 1 and Task 2. Make a copy of the files to your Google drive or locally if you prefer. You can access the data from this link.

Teams: You should form groups of up to 3 students (we can help - google form). Submissions with fewer students are possible but will *not* be assessed differently. Please choose a team leader to create an account on codalab and download the data. Specify on the profile the team name and team members (names of students). It is your responsibility to request access to join the competition with enough time for the organizers to add you - we strongly recommend doing this at the beginning.

Data and evaluation: This website will provide a training set to build models, a development set for model selection or hyperparameter tuning, and a test set. We recommend re-splitting the training data to create a separate dev set for preliminary experiments, but for the final evaluation we recommend using the provided splits. To evaluate your system you can download the true labels for the test set from the competition website.

Report: You will need to submit to CATe a written report (pdf format) by February 26th (by 19:00), which should include the results on the competition's test set. The report should include your insights and approach from both approaches on your chosen task, including:

1) **Details on the design of models**: architecture, hyperparameters (and their selection process), algorithms used for learning, optimisation)
2) **Type of input data / any preprocessing**: word embeddings (which type? pre-trained? contextualised?) vs bag of words vs pre-extracted features, etc.
3) **Performance obtained for the blind test set**
4) **Analysis/insights/discussion** on the results, the challenges encountered in designing your models, and any insightful findings you have found along the way (for example, what didn't work, and how did that lead you to your chosen method).

Your report should not have more than 4 pages of content and 1 additional for references. It should follow the following latex style: `https://www.overleaf.com/read/xhxbhtgjgbxv` Make sure you download or make a copy of the overleaf project as this one is shared with all of you.

Code: Finally, you will need to submit your code: please add the url to a colab file with your Jupyter notebook in your pdf report. Your code should be self-contained and should work without intervention when we try to run these. You will need to add code to download any libraries required in your notebook, and also add code to download any word embeddings that may be required.

Marking: Your mark will be based on the diversity and creativity of ideas used to devise your models (including pre-processing), not on the performance of the models on the task. The code will be checked and marked according to whether it follows basic coding good practices, such as organization, readability and documentation and comments. You are allowed to use any libraries for NLP/ML, but make sure to add some of your own code (see coding instructions).

The marking of the code and reports will be based on the following:

- Clarity and insightfulness of the written report
- Organization and documentation of code (make sure to add a readme and comments)
- Creativity of linguistic pipeline (pre-processing, analysis in report, etc.)
- Appropriate use of machine learning methodology

look at data, where it fails, try to find out why it fails
preprocess