Learning from Data – Final Project
Hyperpartisan News Detection

General remarks

Differently than in the other assignments, for this final project we are not specifying exactly how you should do tackle the task you're given. In other words: the *way* you approach this problem is up to you, and it is an important part of the assignment itself, as it is normally the case that given a (learning) problem you will have to identify your best strategy to deal with it. Grading will be determined by your model, your report, and the final presentation you will give for everyone in class.

Please note that also differently than in the other assignments, you will work in teams. Teams will be made of **max three students**, and you can pair/group as you wish. It is important that work is spread as equally as possible, and that both the report and the presentation are prepared and given by all members. The report will also need to specify as much as possible the division of labour.

Deadline for submission on Nestor: Friday 26th of October 2018, end of day.
What you have to hand in by the deadline:

- script with your system. Your script should take a `.csv` training file and a corresponding `.csv` test file as arguments, as you've done for previous assignments.

  Please, remember that the output of your system should be a file with three space separate fields: `id hyperp-value bias-value`.

- report with system description (see Section 2 for details. Please, make sure to hand in a **pdf** file following the usual report template)

After submission, your systems will be run on test data which we have withheld, and you can report results on your presentation (hopefully we'll get them back to you in time), together with a description of what you have done and your system.

Presentations' week: October 29th. Schedule: TBA (most likely morning of October 29th, 2018)

1 Task: Hyperpartisan News Detection (HND)

Main reference page:
`https://pan.webis.de/semeval19/semeval19-web/`

Main reference paper:
Potthast et al. (2018)
`http://aclweb.org/anthology/P18-1022`
`https://github.com/webis-de/ACL-18`

Task Given a news article text:

- decide whether it follows a hyperpartisan argumentation, i.e., whether it exhibits blind, prejudiced, or unreasoning allegiance to one party, faction, cause, or person. This is either true or false
- decide which bias it has - one of left, left-center, least, right-center, right.

HND is organised as a *shared task* within SemEval 2019 (`http://alt.qcri.org/semeval2019/index.php?id=tasks`) and PAN 2019 (`pan.webis.de`).

1.1 Data

You will get a portion of the released training/development set for this task. The full dataset contains 1 million articles. It is split in training (200,000 left, 400,000 least, 200,000 right) and validation (50,000 left, 100,000 least, 50,000 right), where no publisher that occurs in the training set also occurs in the validation set. All articles are labeled by the overall bias of the publisher as provided by `BuzzFeed` journalists or `MediaBiasFactCheck.com`. Note: the trial data is not fully cleaned. Due to some encoding error, some characters are replaced by question marks.

Size and format You are getting 50,000 examples, balanced according to being hyperpartisan or not. Note that instances come from a limited number of sources, which can be biased in themselves. We have ensured that no source (`publisher`, see below) represented in the training is also represented in the test data, so that this information cannot (and should not) be leveraged.

The format is a `.csv` file, with the following fields:

```
id      hyperp  bias    url     labeledby       publisher       date    title   text    raw_text
```

- `id`: contains the id of the document, and must be used in the output file that your system produces
- `hyperp`: this field indicates whether the news is hyperpartisan or not.
  Possible values are `{true,false}`

- bias: this field indicates the stance of the news.
  Possible values are {left,left-center,least,right-center,right}. See Fig 1 to see how these values interact with the hyperp values
- url: this is an open field with the url of the news
- labeledby: this specifies who established the hyperp value (it's almost always the publisher)
- publisher: this is the source the piece of news is coming from. As said, there is no overlap in publishers between train and test
- date: the date the news was published
- title: the title of the news piece
- text: only text, any additional mark-up has been removed
- raw_text: everything that was in the origina textl, including XML tags, links, etc.

Note that not all instances are necessarily complete in all fields.

Structure of test data The structure of the test data is identical to that of the training data. In the test file, we will use the dummy XXX for the values of hyperp and bias.

1.2 Evaluation

We will evaluate three tasks:

- assignment of the hyperp label
- assignment of the bias label
- joint assignment of the hyperp and bias label for the same news piece

For each of the tasks we will use F-score for each label, macro F-score for the average, and also accuracy overall.

Please note that **these measures must be implemented in your script**.

The example that we provide, running a Naive Bayes classifier with bag of words with raw counts, is considered as baseline.

2 What you have to do

1. You have to produce **one HND final model**, that is a model that given a piece of news will classify it as hyperpartisan or not and will tell which bias it is, out of the five values specified above and that you find in your data. Note that as done for previous assignments we will run your model on test data which you haven't seen before. Your model can be a simple model or an ensemble. Please, make sure that you provide all necessary scripts to run it, and indicate any dependencies we might need to take care of when executing it.

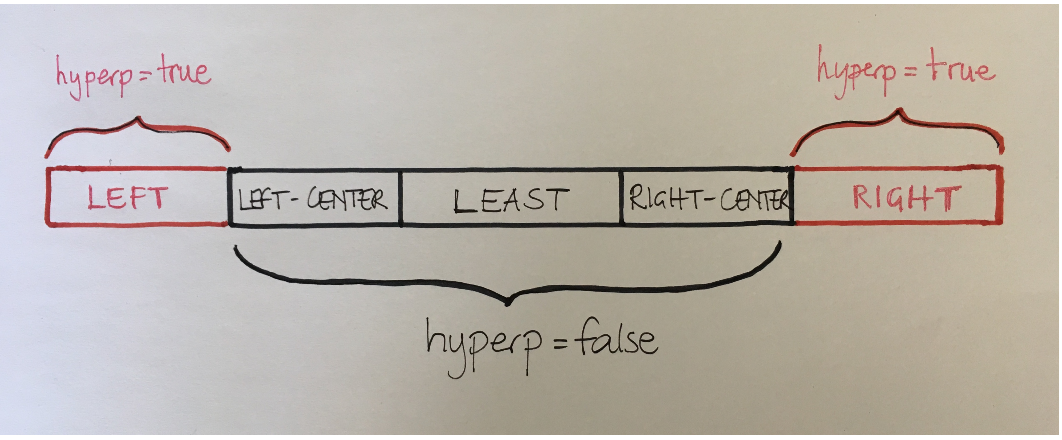

Figure 1: Interaction of `hyperp` and `bias` labes in the data/task

You have to make sure that:

- the output of your system is a space separated file with three fields, as follows:

  ```
  id hyperp-value bias-value
  ```

  for example:

  ```
  0318610 true right
  1469392 true left
  1240151 false least
  ...
  ```

- your script outputs the desired evaluation measures for each task (see above).
  - for hyperp: precision, recall, F-score per class, macro F-score, accuracy
  - for bias: precision, recall, F-score per class, macro F-score, accuracy
  - for joint labels: precision, recall, F-score per class, macro F-score, accuracy. Note that for this case you will have more "classes" ("true left", "true center-left", ...). For interaction of classes, see Fig 1.

You can decide to tune your system by setting aside a development set (using standard splits as we've done for assignments), or via cross-validation.

You can obviously use all the support from scikit-learn, Keras, and NLTK for this, and any other library you find useful. You can use any features and any learning algorithms you like. You are encouraged to experiment with several different features, you can make your own embeddings, use existing ones, etc. You are also welcome to incorporate more data, if you have it and wish to do so. Anything you use will have to be mentioned in your report.

2. You are also asked to produce a **report**. The report should contain the explanation of how you tackled this problem, a description of the features you used, any feature selection method you applied, the algorithm(s) you chose to learn your model, including parameter tuning and setting, any additional data/resources you incorporated, and how well you

do when developing (either via a separate dev set or via cross-validation) in terms of accuracy, precision, recall, f-score. You should also justify your choices explaining why you selected a certain approach, certain features, the learning algorithm, and so on. One important aspect of the report will be also specifying who did what in the team, how the labour was split, and whether there was any imbalance due to any reason you would like to mention.

3. Finally, you are asked to produce a **presentation** in which you will explain to the others what you have done, and why. You will have 15 minutes for this, including questions (think in terms of 10+5). Please, bear in mind that the presentation will contribute to the final grading as well, so all team members will have to contribute. Before the presentation you will be given your results on the test set (we will run your system as soon as we get it, so if there are no glitches, you will get results on test data right away after submission), so that you can refer to those as well.

References

Potthast, M., J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein (2018). A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 231–240. Association for Computational Linguistics.