**1.1**

The first 8 labels are all propaganda techniques and are a subset of those identified in the Propaganda Techniques Corpus (Da San Martino et al., 2020). The final label not propaganda indicates that no propaganda has been identified in the text. The second column contains a sentence or chunk of text where the propaganda technique has been identified (or no propaganda has been identified in the case of not propaganda). Note the use of additional tokens <BOS> and <EOS> which indicate the beginning and end of the span of text (within the sentence) which is actually annotated with the given propaganda technique. In the first example above, the span of text “our soldiers” has been identified as an example of flag waving in the context of the sentence “I want to get our soldiers out.”

1. Build and evaluate at least 2 approaches to classify whether a sentence contains propaganda or not.

2. Given a snippet or span of text which is known to contain propaganda, build and evaluate at least 2 approaches to classifying the propaganda technique which has been used.

In this assignment you are expected to complete both tasks above and investigate at least 2 different approaches to making classification decisions. The approaches used for task 2 may be the same or different to the approaches used in task 1. Your solution does not need to be novel. You might choose to investigate 2 of the following approaches or 1 of the following approaches and 1 of your own devising.

It does not matter how well your method(s) perform. However, your methods should be clearly de- scribed, any hyper-parameters (either fixed, varied or optimised) should be discussed and there should be a clear comparison of the approaches with each other — both from a practical and empirical perspective.

**1.2 Report**

Your report should be in the style of an academic paper. It should include an introduction to the problem and the methods you have implemented. It might contain a brief discussion of related work in the area but the focus here should be on your practical work rather than producing a comprehensive literature review. Also, make sure you describe your solution and not just the theoretical background of the approach. For example, the theoretical background on how word embeddings are learnt using word2vec might be useful to motivate your approach but does not constitute a description of your method to solving the task using word2vec — there are many ways word2vec can be used to provide a solution and it is this that you should focus on in the description of your method. You should also make sure you discuss any hyper-parameter settings - both those which you have decided to fix and any which you are investigating. Justify your design decisions. You should discuss and justify the method of evaluation. You should provide your results and compare them with any baselines. You should also provide some analysis of errors — do the approaches make the same or different mistakes and can you comment on the types or causes of errors being made? You should end with your conclusions and areas for further work. You should also submit your code as an appendix. Your report (including figures and bibliography but not including code appendix) should be no longer than 8 sides (3000 words of text plus figures and bibliography). Your code in the appendix should be clearly commented.

**A white sheet with black text

Description automatically generated**

**Abstract**

This report investigates two different approached to classification in the context of propaganda detection tasks. The investigation will be carried out through experimentation and analysis of each task, with performance analysed with refrence to wider empiracle considerations.

Owing to their ubiquity and importance in contemporary natural language application [CITATION] both approached explored will utilise word-embeddings as inputs to a multi-layer perceptron (MLP) classifier model. The first approach will use pre-trained static word embeddings trained using the word2vec algorithm [CITATION]. The second will use contextualised word embeddings encoded by the BERT transformer model [CITATION].

The differences between these appraoches, their performance on the different tasks and comparison between them shall be used as a basis for discussion of both techniques as well as representation and classification in natural language processing more generally.

**Problem Outline (10)**

The broad aim of the assigment is to build and evaluate 2 approaches to two sepatate but similar classification tasks. Both tasks relate to propoganda detection, a problem which is well introduced and discussed by Da San Martino et al [CITATION], a paper which is also the origin of the data of which a subset was used as the data for this assignment [citation].

A more detailed desciption of the data, and how it was used in the solutions to the problems can be found below, but in brief; the data provided was in the form of a table of label and text pairs, where the label either described a type of propoganda found in the text (for example ‘flag-waving’ or ‘loaded language’) or labelling it as ‘not propoganda’. Each text item was a passage of text which contained segment start and end tags (‘BOS’ and ‘EOS’). As described by [CITATION], the segments where to what the labels referred. As such, if the lbael was ‘not propoganda’ then the segment tags were arbitrary as there was no propogrande in the text to segment.

The aim of task one was to build a classifier model which could take in the entire text passage, and classify whether it contained a propoganda segment or not. As such it was a binary classifiction task, with a passage either containing propoganda (with a binary class of ‘1’) or not (a binary class of ‘0’) This shall be referred to as task 1 of T1 from this point. The second task was a multi-class classification problem. In this task all non-propoganda instances were removed, and the task was to apply the correct specific propoganda-type-label to the given propoganda snippet.

**Propoganda**

As pointed out in the paper [CITATION] propaganda detction in an important, but also a particularly challenging task for NLP.

Important why:

* Fake news
* Stirrring of hatred
* People unaware

Challenging why:

* People unaware so it is subtle
* Subjective!
* other

Many of these challenges are mitigated in this particular assignment because we are using prelabeled data. As such any problems are issues are built in and despite good performance on this data, one would have to rigorously any models built on this or even a similar dataset to be confident that they are in fact able to detect ‘propaaganda’

**Method (25)**

In this section a general introduction to the data, its distribution and the preprocessing required for both task shall be described. More detail will then be given on the two approaches used.

As the same two approached were used for both tasks, the approaches shall be described in detail here, grouped by approach, whereas comparison in performance – both practically and with empiracl reflection, shall be grouped by task, as each task brings out interesting nuances of each approach for disucssion.

* **Data**

Format – how it looked, what the classes were, dimensions, val train etc

Distibution – just show that they are balanced

Pre-processing – how it looked after preprocessing – label transformation

* **Word2Vec**

Intro to word to vec and some general chat

Process

Flow chart maybe? Sentence –> token -> embedding -> max pool -> MLP -> class

Options – tokenising/stemming /normalise/max/mean pool

* **Bert**

Intro to bert and some general chat (model structure?)

Process

Flow chart?

Sentence = BERT EMBEDDINGS->MLP->Class

Options – padding (get max len), bert direct???? Or extra layer

* **MLP**

MLP for classification

Process – how it works

Hyperparams – batch\_n, lr, epochs, dropout, optimiser, loss function

Flow chart / print of structure

Explain how loss works to drive gradient descent

**Hyper parameter Settings (10)**

Exeriments of hyperparams in each modality and what ended up being best?

**Evaluation (20)**

* HOW IT WAS DONE – explain accuracy and loss calculation and why useful
* Explain confusion matrix and why useful
* Present the results

Performance of different models in the two tests – take best performing one from each and compare in terms of ‘results’ but then also maybe chuck in some other results for balance.

**Analysis (15)**

Actual chat about different representational ability of the different models and how it relates to their performance

Analysis of results

**Further Work (5)**

Reiterate how shit this is, further work would be unsupervised somehow – maybe to detect balance?

**Style (5)**

**Code (5)**