Module: CMP-6059B/7059B Advanced Artificial Intelligence
Assignment: CW1: Applying advanced AI methods for analysing text documents

Set by: Daniel Paredes-Soto (d.paredes-soto@uea.ac.uk)
Date set: Thursday 5 February 2026
Value: 30%

Date due: 15:00 Friday 13 March 2026 (Week 7)
Practical demonstrations in Week 8 (TBC)
Returned by: Friday 27 March 2026
Submission: Blackboard
Checked by: Wenjia Wang (wenjia.wang@uea.ac.uk)

Learning outcomes

- Gained deep understanding of classical NLP and NLU techniques for text analysis with focus on document classification and topic discovery.
- Gained practical experience in designing, training and evaluating neural networks for classification, including a deep understanding of model improvement techniques and performance assessment.
- Enhanced programming skills for data processing, and effective use of AI libraries.
- Gained practical experience on saving trained models and testing them in isolated environments for real time predictions.
- Enhanced communication skills through the effective presentation of technical work, clear justification of the choices made in the coursework and, clear question answering.

Specification

Overview

This coursework requires the implementation of natural language processing and artificial neural network models to perform text document classification and topic discovery. This assignment should be completed and presented individually.

This coursework uses a provided dataset, available on BlackBoard, containing text-based social media posts with the entries labelled as fake or real content. A description of the dataset has been included in a later section of this document.

In this coursework we are not going to provide any starter code. You will be writing code for data loading, text processing, text transformations, neural networks (shallow and deep learning) from scratch. You will also save your trained models and run them to test on unseen data during the demonstration.

To complete this work, you can use any code provided in both lectures and labs, though you should and will be required to write your own code to complete the necessary tasks. Reference any code that you use where appropriate. You may use the ScikitLearn, PyTorch, TensorFlow and NLP packages documentation for reference only. All code that you submit must be your own, excluding the code from lectures and labs.

Description

The data that must be used for this work and the specific tasks to be completed are described below.

The data

The social-media.csv file contains social media posts linked to headlines of news articles. The dataset has over 89,000 posts linked to 947 news headlines. The posts have been judged and labelled following a rigorous process using the Amazon Mechanical Turk platform, where participants evaluated the degree to which the post content aligns with the linked news headline, and the posts were filtered based on majority voting when consensus was reached.

The ground truth label of the news headline and the majority vote were then used to determine the truthfulness (class label) of the post as follows: if the news headline is real (True) and the majority votes “Agree”, the post is labelled real (True). If the headline is True and the majority votes “Disagree”, the post is labelled fake (False). If the news headline is fake (False) and the majority votes “Agree”, the post is labelled fake (False), and, if the headline is False and the majority votes “Disagree”, the post is labelled real (True). See Appendix A for an example.

Each social media post is described by six attributes: 1 id, 4 features, and the class label in the following order: *id, news_headline, news_headline_ground_truth, post, majority_votes, and class_label*. In this coursework, each social media post in the dataset is treated as an individual text document.

Preparations: Load the data and perform a descriptive analysis of the data before starting any processing, and report any relevant observations.

Task 1. Identification of fake text documents

In this task, you need to apply NLP techniques to preprocess the data, including but not limited to syntactic analysis. Then, you need to transform the text into numerical representations (feature vectors), suitable for use by neural networks (NN) classifiers to classify documents as real or fake.

The implementation of NLP techniques for this task is open-ended. Therefore, it is expected that you experiment different approaches and you will assess their impact on the performance of the document classification models to select the most suitable NLP approach for this task.

Reserve a test set containing documents that are never seen in the training of the NN classifiers. It is a crucial step since you want to estimate how well the NN classifiers will generalise to new data. Use a subset to train shallow and deep neural network classifiers. You need to explain how you split the data and justify your choice of such approach.

1.1) Design a multi-layer perceptron (MLP) network that takes the feature data you designed as input and predicts the class_label (truthfulness) of the documents as real (True) or fake (False). You are expected to understand and explain your choice of the MLP architecture, such as the number of inputs, layers and neuron units, activation functions, and the number of outputs.

1.2) Design a deep learning (DL) neural network to perform the same as in Task 1.1. You are expected to understand and explain your choice of the DL model, like the number of inputs, layers, types of layers, convolutional and fully connected layers, activation functions, and outputs. You also need to justify any other configurations.

Perform the following for both types of networks, MLP and DL:

1. Train the network and note down the performance accordingly to the network type.
2. Choose 3 hyperparameters you think are most worth refining to improve the performance of the network. You are expected to understand and explain how these hyperparameters affect the performance of the model. Tune the chosen hyperparameters with as many values as you need until you are satisfied with the performance, you should use the default values as the baseline. Create visuals to show, which should be shown in the demonstration, your tunning approach and the results during the experimentation.
3. Assess and evaluate the results with appropriate metrics from both the MLP and DL neural networks. You are expected to be able to explain the insights from your evaluation during the demo.
4. Choose the best network model from all the different models you trained and save it for demonstration. You need to justify your choice.
5. Note down the performance of your best model on the test, and training data.

Save the best models for both MLP and DL networks into files using a format of your choice so that you can run them during the demonstration with a given dataset.

Task 2. Topic discovery with natural language understanding (NLU)

In this task, you need to apply NLP and NLU techniques to explore and analyse the content in documents and their linked news headlines in order to automatically discover underlying topics across the text data

You need to implement standard syntactic analysis, including but not limited to parsing or tokenisation, stop word removal, and lemmatization or stemming. You must experiment with at least two different text representation strategies, such as but not limited to BoW, TF-IDF, LDA, Word Vector or Word Embeddings representations. You are encouraged to explore other classical NLP methods where appropriate.

You need to understand the models you implemented and experiments you have conducted, evaluate their performance or the interpretation of the results, analyse and discuss the discovered topics and their link to document content, news headlines, and both the document headline- and document - class labels. You should prepare to be asked for explaining what you have done in this task.

Save the best model(s) using a format of your choice, so that you can run them during the demonstration to analyse new text inputs and show the topic discovery findings.

Relationship to formative assessment

The formative part of the module is provided through the lab exercises and the feedback and discussion with ATs and lecturers.

Deliverables

The assessment is carried out with a bench-demonstration that is supplemented by PowerPoint slides (maximum 7 slides) to provide documentation of the design, model improvement regime, performance evaluation and discussion of results for both tasks, identification of fake documents, and topic discovery in text.

You must submit all your code and PowerPoint slides via BlackBoard by the deadline (3PM, 13/03/2026):

1. Create a .zip file named {your_student_registration}_code.zip containing only your .py files. Do not include any other files produced by your code, or files from the development environment.
2. Create a PowerPoint file named {your_student_registration}_presentation.pptx (or .pdf).

You must use the code and PowerPoint slides submitted to BlackBoard in the demonstration.

The demonstration is allocated up to 15 minutes (10 for presentation and demonstration and 5 minutes for Q/A and transitions). Specific guidance for demonstration will be provided in due time.

In the demonstration, you will be asked to run your pre-trained and saved models. Therefore, you must not retrain any models during your demo.

You are expected to deliver high-quality slides alongside a well organised demonstration that should contain high quality (e.g. well-structured and without obvious inefficiencies) and readable code. You are also expected to be able to conduct clear discussion and concise question answering in relation to the work produced.

Resources

Essential material is provided in the lecture slides and lab sheets and code (Weeks 1 to 5). Additional resources can be consulted using the Internet. However, all code that you submit must be your own or permitted code.

Plagiarism, collusion, and contract cheating

The University takes academic integrity very seriously. You must not commit plagiarism, collusion, or contract cheating in your submitted work. Our Policy on Plagiarism, Collusion, and Contract Cheating explains:

- what is meant by the terms 'plagiarism', 'collusion', and 'contract cheating'
- how to avoid plagiarism, collusion, and contract cheating
- using a proof reader
- what will happen if we suspect that you have breached the policy.

It is essential that you read this policy and you undertake (or refresh your memory of) our school's training on this. You can find the policy and related guidance here: https://my.uea.ac.uk/departments/learning-and-teaching/students/academic-cycle/regulations-and-discipline/plagiarism-awareness

The policy allows us to make some rules specific to this assessment. Note that:

In this assessment, working with others is *not* permitted. All aspects of your submission, including but not limited to: research, design, development and writing, must be your own work according to your own understanding of topics. Please pay careful attention to the definitions of contract cheating, plagiarism and collusion in the policy and ask your module organiser if you are unsure about anything.

The use of Large Language Models, such as ChatGPT or any other generative AI, to produce any part of your submissions is strictly prohibited. This include but not limited to generate code, process data, test models, writing or creating PowerPoint slides content. The use of generative AI is treated as a serious offence by the school and University, and it may be considered as a third-party with regards to contract cheating.

Marking scheme

Marks will be allocated as follows:

Task 1. Identification of fake text documents – (45%)
- Data descriptive analysis and pre-processing (20%)
- Shallow Neural Networks, design and improvement (10%)
- Deep Learning Neural Networks, design and improvement (15%)

Task 2. Topic discovery with natural language understanding (NLU) – (35%)
- Data preprocessing (5%)
- Topic discovery approach, design and improvement (15%)
- Analysis and interpretation of results (15%)

Structure, organisation, professionalism and Q&A of the demonstration – (20%)

Appendix A. Data annotation example

Example: social media post labelled as True

id	news_headline	news_headline_ground_truth	post	majority_votes	class_label
9999	End of eviction moratorium means millions of Americans could lose their housing in the middle of a pandemic.	TRUE	As many face backlogged rent payments, Americans can observe that the eviction moratorium was simply a bandaid, and not a solution, to the issue of affordable housing in the U.S. Moving forward, policymakers must consider expanding Section 8 housing and voucher options #padp8670	Agree	TRUE

Real* news headline (news_headline_ground_truth = True***):** "E*nd of eviction moratorium means millions of Americans could lose their housing in the middle of a pandemic.*"

Linked social media post example: "A*s many face backlogged rent payments, Americans can observe that the eviction moratorium was simply a bandaid, and not a solution, to the issue of affordable housing in the U.S. Moving forward, policymakers must consider expanding Section 8 housing and voucher options #padp8670*"

Annotation method: The majority of the annotators (Mechanical Turk participants) agreed that the content of the post was consistent with the information in the news headline. As a result, the post was labelled as **True** (*class_label = True*) in the dataset.