Natural Language Processing Coursework Spec (2026)

Submission Deadline: Wednesday, **4th March (7pm)**.
I encourage you to submit your work sooner :)

This coursework is designed to immerse you in the NLP research lifecycle.

A typical NLP research lifecycle follows these six stages:

1. Task Definition and Literature Review
2. Data Acquisition, Exploration, and Preprocessing
3. Baseline model and proposing a novel approach
4. Implementing your proposed approach
5. Evaluation and Error Analysis
6. Communication and Reporting

Before you begin working, please read this entire coursework spec carefully and let me (Chiraag) know on EdStem if you have any questions.

Stage 1
Task Definition and Literature Review

Progress in NLP is often driven by 'shared tasks', where teams compete to build the most effective models for a specific challenge. The shared task you will be competing in is about detecting Patronising and Condescending Language (PCL). This task was task 4 (subtask 1) in the SemEval 2022 competition. More information about the task is available in the task paper.

Stage 1 typically involves reading and reviewing multiple papers to survey existing research. This helps in understanding the NLP task and identifying the strengths and weaknesses of past approaches. For this coursework you are expected to review just one research paper.

Exercise 1: Critical Paper Review [6 Marks | up to 3 Hours]
Review the PCL paper and provide concise answers to the following:

- Q1. State the primary contributions of this work. (2 Marks)
- Q2. Identify the technical strengths that justify the paper's publication. (2 Marks)
- Q3. Highlight the key weaknesses or areas where the authors failed to provide sufficient evidence. (2 Marks)

To help you review NLP Research papers, please read the Appendix: Reviewing NLP Research Papers.

Stage 2
Data Acquisition, Exploration, and Preprocessing

The task data can be found here. More specifically, you will be using the dontpatronizeme_pcl.tsv file. An allocation of this data into **train** and the **official dev set** is provided here. The **official test set** (without the labels) can be found here. The test-set labels are held out and will not be shared with you. We will use it to evaluate your submitted model's performance after the coursework submission deadline.

Note: the task repository also contains a breakdown of the type of PCL language detected for each example (broken down into seven categories). You are welcome to use this additional label information if it is helpful but don't forget the task you are working on is task 4 (subtask 1) which is Binary Classification (PCL vs No PCL).

Stage 2 is mainly about exploring the data. It involves a deep dive into the dataset to identify linguistic patterns, class imbalances, and noise. If you identify noise that can be cleaned easily, you ensure higher quality inputs for your binary classifiers and a more reliable training process downstream.

Exercise 2: Exploratory Data Analysis (EDA) [6 Marks | up to 3 Hours]
Analyse the PCL dataset using two distinct EDA techniques (3 marks each). For each technique, you must provide:

- **Visual/Tabular Evidence:** A figure or table.
- **Analysis:** A brief description of the findings.
- **Impact Statement:** An explanation of how this specific insight influences your approach to the PCL classification task.

To help you do the above exercise, please refer to the Appendix: Exploratory Data Analysis for inspiration.

Stage 3
Baseline model and proposing a novel approach

In NLP research, a baseline model is a standard existing approach used as a reference point. Its primary purpose is to provide a 'floor' for performance. The PCL Shared Task organisers had provided the following baseline model: RoBERTa-base baseline model. This baseline model for the task 4 (subtask 1) achieved an F1 score of 0.48 on the official dev-set and 0.49 on the official test-set. [Note: These results are measured using the F1 score of the positive class which are the PCL examples. 'No PCL' is the negative class.]

While the ultimate goal is to propose an approach* resulting in a model that outperforms all the other models on the shared task, a first step is to propose something that outperforms the baseline.

*An approach refers to any justifiable deviation from the baseline, such as a novel model architecture, a refined training methodology, or a strategic modification of the data distribution, or fine-tuning of an existing model trained on another related task (transfer learning), etc.

Exercise 3: Describe your proposed approach [4 Marks | up to 2 Hours]
Clearly articulate your strategy to surpass the RoBERTa-base baseline. You may include figures or flowcharts or examples to explain your proposed approach.

- Proposed approach (2 marks)
- Rationale and Expected outcome (2 marks)

In case you are experimenting with multiple approaches, then only describe the approach that you eventually submit (BestModel). Thus, you may answer the above Exercise 3 after completing Exercises 4 and 5.1.

Stage 4
Implementing your proposed approach

Now you implement your proposed approach and train the model you would like to submit. (By training, I mean that it can be training from scratch or Hyper-parameter tuning or Fine-tuning an existing model or setting up a new pipeline or training on a different dataset, etc. Basically, full implementation of your proposed approach.)

Recall from Stage 2, you have access to the Training set, Official Dev set, and Official Test set without labels. Since the labels of the official test set are held out and not available to you, you could use the official dev set as your own test set. If you are experimenting with multiple approaches, you can compare the performance of the different approaches on this official dev set (your own test set). Thus, you may need to create your own internal dev set from within the train set for the purpose of hyper-parameter tuning.

Exercise 4: Model Training [1 Mark | up to 8 Hours]
Create a GitHub or GitLab repository for your coursework. This repository can initially be private while you are working on your coursework but it must become public after the deadline for the GTAs to assess. Next, train your model(s). You must push your best performing model and its code or ipynb or

Colab notebook in a folder named **BestModel** in the repository. We will inspect the code manually to check if it is what you proposed in Exercise 3. You must include a link to the GitHub/GitLab repository on the front page of your report. Please check before submission that the link works.

Stage 5
Evaluation and Error Analysis

In a Shared Task environment, success is measured through two distinct lenses:

1. Global Evaluation (The Leaderboard)
2. Local Evaluation (The Researcher's Lab)

At the global level, the task organizers act as 'blind judges'. They evaluate every submitted model on a hidden test set using standardized metrics and release the leaderboard (For example, here is the official leaderboard of the PCL Shared Task).

In this coursework, the official test set labels are hidden from all students. We will be using F1 score to evaluate all the submitted models (PCL = 1 = Positive. No_PCL = 0 = Negative). We will publish a leaderboard on EdStem.

At the local level, individual teams investigate their own model's behavior. This internal validation is often more valuable than the final score because it reveals the model's logic. Some examples of local evaluation are as follows:

- Error Analysis: Manually inspecting samples where the model failed. Is it struggling with sarcasm, or perhaps with specific keywords? If you have the baseline model outputs as well, you can explore cases where a) both your model and the baseline model predicted correctly, b) both models predicted incorrectly, c) and d) where one model predicted correctly and other predicted incorrectly.
- Ablation Studies: Systematically removing parts of the model (e.g., removing a specific layer or data cleaning step) to see if performance drops. This proves which part of your 'novel approach' actually works.

- Custom Metrics different from the metrics used by the shared task organisers: Using metrics like Precision-Recall Curves or Confusion Matrices to understand the trade-offs the model is making or your own metrics that help you draw a useful insight.

Exercise 5.1: Global Evaluation [6 Marks | up to 30 mins]
Submit your model's prediction on the official dev set and the official test set. You must push the following two files in your GitHub/GitLab repository that you had created in Exercise 4:
– Dev set predictions as **`dev.txt`**
– Test set predictions as **`test.txt`**

Please make sure these files are easy to find in the repository for the GTAs marking your submission.

Recall, the official dev set evaluation is like a public test, i.e. the labels are available for all to see. The test set evaluation is a private test, i.e. the labels are not available to anyone. You will see your results on the test set when we release the leaderboard and in the feedback to your coursework.

The format of **`dev.txt`** and **`test.txt`** should be one output prediction per line. For example, the test set has 3832 lines of input text. So, **`test.txt`** should also have 3832 lines of predictions (0 or 1 in each line).
0 means No PCL.
1 means PCL.

Submit a name (upto 20 characters long) on the front page of the report that we can use to identify your work for the leaderboard. This can be your own name or some random name if you want to remain anonymous.

You will be awarded

- 1 mark for submitting **`dev.txt`** and **`test.txt`** in the correct format.
- 1 mark if **`dev.txt`** outperforms the baseline of 0.48 F1 score on the official dev set.
- 1 mark if **`test.txt`** outperforms the baseline of 0.49 F1 score on the official test set.
- 1 mark if you rank in the top 60% on the leaderboard.
- 1 additional mark if you rank in the top 30% on the leaderboard.
- 1 additional mark if you rank in the top 10% on the leaderboard.
- I will be lenient with borderline cases.

Exercise 5.2: Local Evaluation [5 Marks | up to 3 Hours]
Perform Error Analysis (2.5 marks) and any other local evaluation (2.5 marks) of your model. You may use the official dev set (your own test set) for this purpose. Clearly articulate using examples, tables, figures, etc.
Remember, a 'failed' experiment is still a good experiment if you can explain why it failed. A high-quality Error Analysis is often worth more than a high F1 score alone.

Stage 6
Communication and Reporting

An important research skill is to communicate and report your work clearly and effectively. NLP researchers typically do this by submitting well written research papers to NLP conferences, well written README instructions in their github/gitlab repositories, and sometimes via other means like website, blogs, tweets, videos, etc.

In this coursework, we will assess your communication and reporting skills from the quality of your submission.

Exercise 6: Coursework Submission [2 Marks | up to 30 mins]
You must submit a Report (pdf) with clear sections for each of the above exercises (except Exercises 4 and 5.1). For Exercises 4 and 5.1, you must submit a clickable working link to your GitHub/GitLab repository for this coursework on the front page of your report.

The 2 marks for this exercise are at the discretion of the GTAs marking your submission. If the report and repository are well written and well organised, you will get 2 marks. If these are OK, then you will be awarded 1 mark. If poorly written and difficult to navigate the repository, then 0 marks.

Workload Management

As you begin working on this coursework, you will realise that NLP Research is inherently open-ended. It is a marathon, but this coursework is a short structured sprint. The purpose of this coursework is to experience the NLP Research lifecycle; not necessarily to do groundbreaking publication worthy research which would ideally require larger teams, more compute resources, and longer duration. So, do not get discouraged if you don't hit SOTA; meaningful insights into why a model fails are often more scientifically valuable than a 'black box' that succeeds. Prioritise a clear, well-documented approach over endless experimentation. Notice how the marks are distributed in this coursework and strategise your time and effort wisely. Beyond a certain point, extra hours spent, for example extra hours for additional tuning of hyperparameters, yield diminishing returns in terms of marks.

Appendix: Reviewing NLP Research Papers

When you are asked to review an NLP research paper, you are essentially playing the role of a **technical judge**. You aren't just summarising the paper; you are evaluating its quality and honesty.

Most formal reviews ask for these five specific components:

1. The Summary (The "What")

The review usually starts with a brief, objective summary.

- **What to answer:** What problem are they solving? What is their main contribution (a new dataset, a better algorithm, or a clever experiment)?
- **The goal:** Prove to the editors/instructors that you fully understood the technical core of the work.

2. Strengths and Weaknesses (The "How Good")

This is the heart of the review. You need to look at:

- **Originality:** Is this a new idea, or did they just "re-package" an old method?
- **Soundness:** Are the math and the experimental setup correct? For example, did they use a fair baseline to compare against?
- **Significance:** Does this paper actually matter to the NLP community?

3. Evaluation and Results

You must scrutinize the "Experiments" section.

- **Metrics:** Did they use the right tools to measure success? For example, using **Accuracy** on a dataset where 99% of the labels are the same is misleading.
- **Ablation Studies:** Did they prove that their specific change is what caused the improvement?
- **Error Analysis:** Did the authors look at where their model *failed*? A paper that only shows wins is usually hiding something.

4. Clarity and Reproducibility

- **Writing:** Is the paper easy to follow, or is it buried in unnecessary jargon?
- **The "Recipe":** Could a stranger recreate this exact model using only the details in the paper? If they didn’t provide hyperparameter settings (like learning rates or batch sizes), the answer is likely "No."

5. Recommendation (The "Verdict")

Finally, you are asked to give a score or a recommendation.

- **Accept:** The paper is solid, new, and well-explained.
- **Weak Accept:** Good idea, but needs better experiments or clearer writing.
- **Reject:** The math is wrong, the results aren't significant, or the authors ignored existing work that already solved the problem.

Comparison of a Good vs. Bad Review

Feature	A "Bad" Review	A "Good" Review
Tone	"This paper is bad and I didn't like it."	"While the idea is interesting, the lack of an ablation study makes it hard to verify the results."
Specificity	"The results are weak."	"The model only outperforms the baseline by 0.2 BLEU points, which may not be statistically significant."
Advice	"Fix the grammar."	"I suggest the authors clarify the transition between Section 3 and 4 to improve flow."

Appendix: Exploratory Data Analysis

In NLP, **Exploratory Data Analysis (EDA)** focuses on the linguistic properties, patterns, and potential biases hidden in the text.

Here is a breakdown of the typical NLP EDA workflow:

1. Basic Statistical Profiling

Before looking at the words, you look at the structure. This helps you determine your model's constraints (like maximum sequence length).

- **Token Count:** What is the average, minimum, and maximum sentence length?
- **Vocabulary Size:** How many unique words exist? This dictates the size of your embedding layer.
- **Class Distribution:** Is the dataset balanced? (e.g., In a hate speech task, if 98% of the data is "Non-Toxic," your model might achieve 98% accuracy just by guessing "Non-Toxic" every time).

2. Lexical Analysis (The "Word" Level)

This involves digging into the actual language used in the dataset.

- **N-gram Analysis:** What are the most common pairs (bigrams) or triplets (trigrams) of words? This reveals common phrases or domain-specific jargon.
- **Stop Word Density:** How much of the text is "filler" (the, is, at)? High density might mean you need more aggressive cleaning.
- **Word Clouds & Frequency:** A quick visual check to see if the most frequent words actually align with the task.

3. Semantic & Syntactic Exploration

Modern NLP requires understanding the "meaning" behind the statistics.

- **Part-of-Speech (POS) Tagging:** Are there more verbs than nouns? (e.g., in instruction-following tasks, verbs are dominant).
- **Named Entity Recognition (NER):** Does the dataset focus on specific people, locations, or organizations?
- **Embedding Visualization:** Using techniques like **t-SNE** or **UMAP** to project high-dimensional word vectors into 2D space. This allows you to see if similar concepts are naturally clustering together before you even train a model.

4. Identifying "Noise" and Artifacts

The most important part of EDA is finding the "trash" in your data:

- **Duplicates:** Repeated entries can lead to data leakage (the model seeing the same sentence in both training and testing).
- **Special Characters/HTML:** Finding hidden tags like `&` or `\n` that could confuse a tokenizer.
- **Outliers:** Extremely long or short sequences that might be errors in data collection.

Why is EDA critical for your coursework?

If you skip EDA and go straight to training, you are flying blind. EDA tells you:

1. **If you need to augment your data** (if the classes are imbalanced).
2. **What your `max_length` should be** (to avoid cutting off important info).
3. **If your task is "too easy"** (e.g., if the model can guess the answer just by looking for a specific keyword).