ANLP Assignment 2025:
Training and analyzing Transformer-based NMT models

due: Tuesday, 18 November, 12pm (noon), via Gradescope

1 Overview of task, motivation, and goals

What will you do?

In this assignment, you will train and evaluate Transformer models for German-to-English neural machine translation (NMT), and you will consider ways to improve their performance.

Since training a high-quality model would be too intensive, we will ask you to train a model on a relatively small dataset and compare its performance to a model that has the same architecture but is pre-trained on a much larger dataset. Concretely, you will be asked to:

- train an encoder-decoder Transformer model from scratch on a small dataset containing only short sentences.
- conduct an error analysis of its translations on both shorter and longer sentences, and compare them with those of an NMT model with the same neural architecture but pre-trained on significantly more data.
- implement some standard evaluation metrics and consider how they do (or do not) reflect the kinds of errors you observed.
- visualize the attention weights and analyze the extent to which they can be used to identify different kinds of errors.
- pick one of the possible strategies to improve one of the models, implement it and analyze the results in light of your observations above.

We provide you with a full task specification and helper code below. You will submit a written report of what you did and why. We will ask you to answer specific questions. The report is the only artefact that we will assess, and it must demonstrate your reasoning and explanatory skills.

Why?

We focus on NMT as this is a crucial real-world use case of NLP models. In completing this assignment, you will practice skills that are important for NLP, by demonstrating that you:

- understand concepts of seq2seq, attention, Transformers, and positional encoding.
- can work with support code to train and quantitatively evaluate NLP models.
- understand how to interpret NLP evaluation metrics in the context of a particular task; can evaluate the strengths and weaknesses of particular models based on detailed error analysis; and can critically reflect on the error analysis itself.
- can justify and implement possible improvements to a model based on the results of your error analysis.
- cad solve a new NLP problem whose solution requires the combination and generalisation of various concepts covered in class or during labs.
- can produce good scientific writing. This means clearly justifying your decisions, analysing your results, and drawing appropriate conclusions from them. We don't just care that you got the right answer, we also care that you are able to explain and justify your answer.

2 Working with others (and not)

We encourage students to complete the assignment for this class in pairs: by working with another student, you can discuss ideas and work things out together. Ideally, try to find a partner with a different skill set from your own, although this may not be possible in all cases. You may discuss any aspects of the assignment with your partner and divide up the tasks however you wish; but we encourage you to collaborate on each part rather than doing a strict division of tasks, as this will enable better learning for both of you. Independently of whether you will work on the assignment individually or as a pair, all members of your group will receive the same mark and you are all responsible for the work, so make sure you have agreed on the final submission before submitting it. You may also discuss high-level concepts and general programming questions with others in the class; but you may not share your specific solutions, answers, or code directly with other groups. Your report must be your own group's work. Important points include (but are not limited to) the following:

- **Any key ideas from outside sources should be appropriately cited**, and direct quotes must use quotation marks. If you re-use code snippets from outside sources, the source should be cited.
- **Be careful when using Piazza**; if in doubt, make your post private. Questions about the assignment can sometimes give away parts of the solution to other students. If you're not sure, use a private post and the instructors will make it public if appropriate.
- **You must take reasonable measures to protect your assessed work from unauthorised access**. For example, you may not store or post your work anywhere in a way that could be accessed by other students (except your partner).
- **You must not use AI except as noted in Section 4.2 below**.

See the School's guidance on academic misconduct for more details: `https://informatics.ed.ac.uk/taught-students/all-students/your-studies/academic-misconduct`

3 Submitting your assignment

- **Answer templates and page limits.** To prepare your submission, you must use either the .tex or .docx templates provided on Gitlab, in which you will fill in your answers to each question. Each answer should start on a new page and strictly observe the page limits for the question (given in the templates), or you will lose marks. This does not mean that answers must use all available pages; some answers can and should be shorter. You are allowed to use equations, tables, and plots in your answers, provided that they do not exceed the page limits. You should not include code or appendices. If you require references, they may go beyond the page limit.
- **How to submit.** You will submit your assignment through Gradescope, via Learn. You should submit a single .pdf file, and if you are working in a group, **only one of you should submit a file**. You can add your partner to the submission in Gradescope.
- **Late submissions.** This assignment follows Rule 2: Extensions are not permitted and Extra Time Adjustments (ETA) for extensions are not permitted. For more information, please consult:

  `https://informatics.ed.ac.uk/taught-students/all-students/your-studies/late-coursework-extension-requests`

4 Assignment specification and code

The full set of materials for the assignment is available at `https://git.ecdf.ed.ac.uk/anlp/course_materials/current/assignment`. This repository contains this very pdf file with assignment instructions; a Jupyter notebook containing helper code and detailed instructions/questions for the assignment; and the templates for your report.

4.1 Setting up a Notebook with GPU access

You will work on a single Jupyter notebook for all the assignment tasks. Noteable only supports CPU runtimes, so running the experiments for this assignment would be excessively slow and impractical even with small-sized NMT models. Hence, you are invited to run the notebooks on a platform that supports runtimes with Graphics Processing Unit (GPU) accelerators. The instructions below explain how to load and execute your .ipynb file on one such free platform, **Colab**:

1. Open `https://colab.research.google.com` on your browser
2. Click on `Upload` → `Browse`
3. Select a notebook from your computer

You are also allowed to run the notebook on other online (GPU-accelerated) platforms or on the University's MLP cluster, if you prefer.

4.2 Generative AI Policy

We also remind you that ANLP has a **strict policy on using Generative AI systems** (such as ChatGPT) for assessed work, as outlined in the course guidance document. Importantly, note that Colab (as well as other online notebook platforms) has some integrated Generative AI features: an autocomplete for your code and a conversational AI assistant.

For the purpose of the assignment:

- it is **allowed** to use the Colab code **autocomplete**;
- it is **forbidden** to use any conversational AI assistant (including the one on Colab).

This also implies that **any usage of Generative AI is forbidden for writing your report**. Failure to adhere to these guidelines constitutes a case of academic misconduct.

4.3 Recommended Preparation

Before you get started, you should review the 4 labs in our course, as each of them contains concepts relevant to his assignment:

- Labs 1 (for data analysis)
- Lab 2 (for text generation)
- Lab 3 (for sequence-to-sequence training and evaluation)
- Lab 4 (for attention and Transformers).

Also, make sure to (re)read the mandatory readings listed on Opencourse for weeks 6 and 7, which include selected parts of the following chapters of J&M 3:

- Chapter 8.1-8.8.2 (Transformers)
- Chapter 12.0-12.3 (Machine Translation)

The assignment assumes that you are familiar with these concepts.

4.4 Tasks

There are five sub-tasks in this assignment, as outlined in the overview at the top of this document. For the details of each subtask, and the questions you will need to answer in your report, please see the Jupyter notebook.

5 Marking and how to do well

5.1 The marking scale

Assignments will be marked on the usual British scale:

Numeric mark	Equivalent letter grade	Approximate meaning
< 40	F	fail
40-49	D	sufficient for Diploma
50-59	C	good; sufficient for MSc
60-69	B	very good
70-79	A3	excellent/distinction
80-100	A1, A2	outstanding/high distinction

Please note typical specifications for marks above 70 across the University:

A1 90-100 Often faultless. The work is well beyond what is expected for the level of study.

A2 80-89 A truly professional piece of scholarship, often with an absence of errors.
As 'A3' but shows (depending upon the item of assessment): significant personal insight / creativity / originality and / or extra depth and academic maturity in the elements of assessment.

A3 70-79
Knowledge: Comprehensive range of up-to-date material handled in a professional way.
Understanding/handling of key concepts: Shows a command of the subject and current theory.
Focus on the subject: Clear and analytical; fully explores the subject.
Critical analysis and discussion: Shows evidence of serious thought in critically evaluating and integrating the evidenced and ideas. Deals confidently with the complexities and subtleties of the arguments. Shows elements of personal insight / creativity / originality.
Structure: Clear and coherent showing logical, ordered thought.
Presentation: Clear and professional with few, relatively minor flaws. Accurate referencing. Figures and tables well constructed and accurate. Good standard of spelling and grammar.

5.2 What are we looking for?

Write your answers to each question separately in your report, as indicated by the template. Different people will mark different parts, so please try to make each answer stand alone.

We are interested in you demonstrating your critical thinking and communication skills and your understanding of NLP concepts, not just your coding prowess or the amount of time you spent tuning models. Therefore, a simple implementation or analysis that is clearly reasoned and explained will receive higher marks than a complex implementation or analysis that is poorly reasoned or badly explained, even if the complex implementation gets better quantitative results.

To do well on this assignment (70-79), focus on doing the following:

- Answer the questions correctly, clearly and concisely. We do not penalize for minor spelling and grammar mistakes, but to do well, your arguments must be clearly presented in good scientific style, as described further below.

- If using figures or tables, ensure they are clearly labelled and legible without zooming in (using font sizes similar to the main text), and that key points are easy to see.
- Use examples and/or quantitative results as appropriate to support your arguments.
- Describe the results of your work, not a history of it. So, avoid descriptions like this one:

  > "We implemented [Method 1], which is formulated as [equation], because [reason]. [Considerably more explanation of Method 1 here]. However, we then realized that [Method 1] wouldn't work [reasons here], so instead we ended up using [Method 2]. [Explanation of Method 2]."

  This is not appropriate style for a scientific report, and it is confusing. The reader initially thinks you used one method, but then later discovers you actually did something else. The reader (whether a marker reading your assignment or a scientist reading a research paper) does not want to know the history of your ideas. The reader wants to know what you actually ended up doing, and why. The above explanation should be rewritten as follows:

  > "We used [Method 2] to implement attention in this assignment, because [Method 1] does not work [reasons here], and [other reasons for using Method 2]. [Explanation of Method 2]."

- If using equations, explain all the terms in the equation, and say what they mean within the context of this report (if appropriate, give actual values). When you are doing an assignment, this shows the marker that you actually understand the equation yourself. When you are writing a longer paper or thesis, in many cases the reader will not actually know ahead of time what the terms in your equations mean so explaining them also helps the reader understand.
- Please do not include code or appendices to the answers to each question. A marker should not need to look at these to understand what you did and why. Also avoid detailed explanations of *how* you implemented things (in terms of data structures, iteration, etc). Focus on *what* you implemented (what does your code do) and *why*. You should demonstrate that you understand the conceptual issues, have done something reasonable where there are choices involved, and can justify why your choices are reasonable. You may also have some arbitrary or trivial choices to make, and in this case it is probably not worth mentioning those in the report. Honing your judgment about what is important and what isn't is an important academic skill and this is a good chance to start working on it.