Sentiment Analysis Competition

Best way to learn NLP is to get your hands dirty, by actually doing it.

This is an individual assignment and not a group assignment. There will be a leaderboard on which your model will be ranked against others, so it is in your best interest not to share your code and model with friends unless you want to share your grades and career with them! Also we would be manually checking your code during a viva, if we find any two models/code are similar both would be honored with an F grade.

It is compulsory for everyone to take part in the competition (even during the dev phase), there should be at least five submissions per week for every individual. In case you fail to do so, it will be heavily penalized.

1. New Job

You recently changed to a new job and have been employed in the tech team of a social media company. One of the activities that the company engages in is understanding user behavior. One good way to gauge user behavior is by analyzing the sentiments of the user. The CEO of the company came to know about the wonders of deep learning and he wants to deploy these technologies in the company. Since you are part of the tech team and more importantly you have done the CS779 course at IITK, the company wants you to develop a system to predict the sentiment of the text written by a user on the social media platform. However, later someone else also approached the CEO and promised to develop a state-of-the-art sentiment prediction system. But since the CEO has faith in you and to be fair with all, he has organized a competition for developing the system.

From the course, you remember different types of neural models, e.g., RNN based models, sequence-to-sequence models, transformer models. You plan to use these to implement a system for predicting the sentiment of a given sentence. There 3 main

sentiment classes that need to be assigned to a sentence: Positive, Neutral and Negative.

This competition requires you to implement a deep learning based sentiment prediction system. In this competition, you have the freedom to choose any neural architecture e.g., RNNs, transformers, etc.

2. Task Definition

Given a sentence, automatically predict the sentiment expressed in the sentence, i.e., assign one of the three labels (positive, neutral and negative) to the sentence.

3. Background Tutorials

To help you get started we are providing some background tutorials that you might find useful:

1. Tutorial on implementing Transformer-Based models: https://pytorch.org/tutorials/beginner/transformer_tutorial.html
2. RNN based classification using GLOVE embeddings: https://colab.research.google.com/github/bentrevett/pytorch-sentiment-analysis/blob/main/2%20-%20Recurrent%20Neural%20Networks.ipynb#scrollTo=3a5e5883
3. The above can be extended with a transformer encoder which can be obtained from: https://pytorch.org/tutorials/beginner/torchtext_translation_tutorial.html

4. Dataset

We are providing the train set, dev test, test set and evaluation script. This will give you a good idea about different models that you would be experimenting with and help you in selecting the final model. Your final model will be evaluated using a separate test set that will not be provided to you during the development phase.

DATASET

The use of external datasets is NOT allowed in this competition.

5. Implementation and Code

All model implementations will be done in PyTorch. In case you are new to PyTorch, check out the following tutorials: https://pytorch.org/tutorials/ . Note in your implementation you should use only the PyTorch library and not any other higher level library built on top of PyTorch. You will be using Google Colab (https://colab.research.google.com/) for running models on GPUs. You are **NOT** allowed to use the HuggingFace library for transformers. **Use of external GPUs other than those provided by the free version of Google Colab is not allowed.**

USE OF PRE-TRAINED LANGUAGE MODELS (LIKE BERT, ROBERTA, etc.) IS STRICTLY PROHIBITED. USE OF SCIKIT-LEARN OR ANY OTHER LIBRARY TO OBTAIN PRE-BUILT MODELS IS PROHIBITED. YOU NEED TO TRAIN MODEL FROM SCRATCH. IF PRE-TRAINED MODELS ARE USED, YOU WILL GET ZERO! YOU ARE ALLOWED TO USE NON-CONTEXTUALIZED EMBEDDINGS: word2vec and GloVe. Use of LLM-based (or based on any other model) coding assistants is strictly forbidden! Use of LLMs is strictly forbidden!

Later you would be sharing your Google CoLab notebooks with us and we would be running those so see if code works or not.

Code needs to be very well documented. Document all the steps, model details, hyperparameters, etc. Please cite all the important references you have used during model development. You can refer to external tutorials and papers but you need to cite them properly in the code documentation. **But it should not happen that you simply copy the code from somewhere. If we found that you have copied, please expect an F grade for the entire course.**

6. CodaLab Leaderboard

You would be evaluating your model on a separate validation set on the CodaLab (https://codalab.org/). You will have to do a minimum 10 submissions during the development phase of the competition and a minimum of 3 submissions during the test phase of the competition. You can keep trying till 23rd November, 2024 11:59PM (end of development phase). On 24th to 26th November, we will have the test phase, when you would be evaluating your model on the test data. All the evaluations would happen on CodaLab. Your final submission at CodaLab till 26th November 11:59PM would be taken as your final model. Your model will be ranked on the leaderboard. This will give an idea about where your model stands with regard to others in the class. TAs will be contacting you regarding this on a regular basis.

To begin with, you need to create an account on CodaLab (https://codalab.lisn.upsaclay.fr) you are required to keep your username as “FirsLetterofNameInCaps_Rollno”, e.g., if your name is Ashutosh Modi and roll number is 123456 then your username is “A_123456”. On the leaderboard your

username (e.g., A_123456) would appear and we would use this to score your performance. If you fail to use this format for creating an account on codalab you will not be able to take part in the competition as your request will not get approved.

CodaLab Competition Link:
https://codalab.lisn.upsaclay.fr/competitions/20488?secret_key=12445231-ec10-421d-a6f1-d414fdf269b4

Note that performance on the dev set would not count towards final evaluation, this activity is only for improving your models. So this means no point in trying to cheat on the dev set by including it in your train set.

Final evaluation on the test set will be done on CodaLab and we will have a leaderboard on the test set as well.

NOTE: The time mentioned on CodaLab is in UTC, you will have to convert the timezone to IST (UTC+05:30 hours) at your end.

7. Evaluation

This competition has to be attempted INDIVIDUALLY and NOT in GROUPS.
The model developed by you will be tested using a separate test set. TAs will be carrying out that evaluation. You need to provide your code and scripts to the TAs and they will run those on the test set. During testing, you are required to submit only one final model. Of course, during the development and training you can experiment with as many models as you like.

Competition will be evaluated on the following parameters:

1. Implementations and Code Documentation: You would be experimenting with dozens of models and finally select one final model for testing. Your code should be very well documented. Since so many models are involved, it would be better to follow standard software project development practices. For example, data preprocessing code may be in a different file, each model might have a separate file on its own, common functions might be part of a utility library, there might be one main file that could be used to call different models in different files, etc.
2. Analysis of the experiments and results. Since so many models are involved it would make sense to do a thorough analysis to understand what works and what doesn't and why. For example, you could use attention maps or other visualizations to explain model performance, etc. This will also help you to select the best model.
3. Novelty and Creativity: Since it is an open problem we are looking for out-of-the-box solutions. Let your ingenuity and creativity take the lead. You

can be creative about each of the points mentioned above. We value unconventional and novel model architectures, something that has never been done before. You could also take hints of existing research papers but don't forget to cite your inspirations.

4. Performance of the model on the final test set. Performance will be relative i.e. your model performance will be ranked against others. Models are going to be evaluated using the F1-Score metric.
5. VIVA: We will conduct an in-depth VIVA about how you developed the models and about the code that you have submitted and the models you have experimented with. Basically the viva is going to be about everything you did in the project.
6. Project Report: In a brief report, outline all your attempts and experiences. See details below.
7. Minimum Submissions: You are required to do a minimum of 10 submissions during the development phase of the competition and a minimum of 3 submissions during the test phase of the competition. A penalty will be imposed if you do not satisfy the minimum submission requirements.

It is needless to say that the course has a VERY STRICT policy about cheating. If we found that you have cheated from another student or you have copied from an external source, you and all those involved would be honored with an F grade. If you think pragmatically, it is better to get less marks in this competition than failing the course. The idea is you learn by doing, it's OK even if you do not have the best performing model. I honor authentic efforts.

8. Report and Code Submission

You are required to submit a report about your solution(s). The report has to be written in latex and compiled into pdf. The details of the report are given in this template: ReportTemplate

In addition to the report you also need to submit code for your solution. Since you have been training on Google CoLab or Kaggle Notebook, you can download it as a Jupyter notebook, zip it and submit it.

9. Timeline

Competition Start Date: 7th November 2024
Development Phase: 7th November 2024 to 23rd November 2024 11:59 PM
Test Phase: 24th November 2024 Midnight to 26th November 2024 11:59 PM

Test Data Release: 24th November, 2024 12:00 Midnight
Final Test Submission: 26th November, 2024 11:59 PM
Report and Code Submission: 27th November, 2024, 6:00 PM.
Viva: 28th November, 2024 (Tentative)
These will be submitted via google form (Note you can submit only once):

https://forms.gle/jBx8HsdYzUNUwe1N6

NOTE: The time mentioned on CodaLab is in UTC, you will have to convert the timezone to IST (UTC+05:30 hours) at your end.

All deadlines are strict and if you do not submit on time you will get a zero in this competition. TAs will be coordinating with you regarding the evaluation.

10. Lost? Whom to contact?

We understand some of you are new to the topic, but the idea is to learn while doing this competition. We also understand that the MT is a difficult task but don't lose hope, seek help. We are here to help you out and conquer MT! If you are able to develop an MT system you can boast about it in your CV and this will give a boost to your career!

For any queries/questions/doubts please contact TAs. Please contact TAs if you are having trouble understanding some concepts or need help with some tutorials on relevant topics, or you are stuck somewhere. Instead of cheating, it is better to seek help from us. If you have any question specifically for the instructor, email to nlp.course.iitk@gmail.com

TAs are available on Slack and these are the contact emails:

Name	Email
Divyaksh Shukla	divyaksh@cse.iitk.ac.in
Sanjeet Singh	sanjeet@cse.iitk.ac.in
Dhananjay Ghumare	dhananjayg@cse.iitk.ac.in
Vaibhav Sharma	svaibhav@cse.iitk.ac.in
Yashwant Holla	yashwant@cse.iitk.ac.in
Junth Basnet	junthb24@iitk.ac.in