CS7.501: Advanced NLP | Course Project

Instructor: Manish Shrivastava

Deadlines: Oct 6 | Oct 22 | Nov 16

General Instructions

1. This is a group project. Each group (team) will consist of **2** to **3** members only.
2. A mentor will be announced for each project. Please make sure to stay in touch with the mentor for any clarifications needed for the project.
3. Feel free to approach TAs if anything seems difficult to grasp. Make use of the TA Office Hours. Ask for tutorials on any common topics of concern if necessary.
4. The deadlines have been planned to allow adequate duration for each submission. Hence, divide the project into deliverables and plan their completion accordingly.
5. When including code in the submission (whenever applicable), it is necessary to include a **README** file explaining how has the code been divided into files and instructions for their execution, along with a **requirements.txt** file containing the list of necessary dependencies.
6. Please ensure there's no plagiarism for the parts that you have to implement. It's okay to use existing code for baselines and supplementary code that you'll be needing. However, your own implementation based on your understanding of the approach is needed for parts that you have to implement.
7. Any detected case of plagiarism would result in strict disciplinary actions, which could also include an **F** grade for the course, apart from **0** being awarded for the project component.

On the Implementation

1. This document has tried to scope each project and your expectations from it to establish a concrete ground. You are free to make any reasonable changes to the scope after discussion with the mentor & TAs.
2. The use of Ada (institute's GPU cluster) will be critical for implementing the architectures needed for the project. Hence, please make sure to include at least one member in your team with access to it.
3. Naturally, make sure you're acquainted with using the compute servers first, and some tips/tricks for working with it. This can be a helpful resource for the same along with the official guide (available on the institute network only).
4. Please start the implementation part of the project early after getting a grasp of the literature and the ideas involved. One typically keeps running into bugs, runtime errors which take days to solve.
5. Even with Ada, you may end up with approaches not being directly implementable due to GPU and runtime limitations. In this scenario, it's on your part to suitably play around with the parameters and make modifications to the approach with appropriate assumptions.

Project Submissions

There will be three submissions for the project, each bearing a considerable weight for the overall project marks.

1. **Project Outline**
   Deadline: October 6 2022, 23:55
   Weight: 15%

   Here, you will be explaining the niche problem that you will solve through the project. Describe the scope of the project properly. Explore the datasets available and their feasibility, metrics for evaluation, and write about your findings. Go through a few relevant papers and include a short literature review for the problem. This will help you identify the baselines that you can use and give you some grasp of the research area. Plan the implementation of the project over the period till final submission, and include a tentative timeline for the same. The overall length of the report is expected to be around 3-4 pages. Submit the report PDF, which should be named in the format **<TeamNo>-Outline.pdf**.

   Apart from this, it will be helpful if you go through an implementation of the approach available on GitHub and try running it to see its reproducibility. Not for the report, but this can help you for the later submissions.

2. **Interim Submission**
   Deadline: October 22, 2022, 23:55
   Weight: 25%

   This checkpoint is to see if you're on track with the project timeline. By this checkpoint, it is expected that you have done some exploratory analysis using your dataset, and checked the performance against the baselines that you'll be using. Remember that you're not expected to implement the baselines from scratch, and it only helps if you add more baselines for evaluating your final approach. You will thus also have an evaluation pipeline ready to check your model performance. Simultaneously, start implementing the approach that you're supposed to. Include all your progress in a report. Submit the code containing your implementation so far along with the report in a .zip file, which should be named in the format **<TeamNo>-Interim.zip**.

3. **Final Submission**
   Deadline: November 16 2022, 23:55
   Weight: 60%

   For the final submission, you should have completed the implementation along with sufficient analysis of results. This includes quantitative analysis showing the performance over the metrics and qualitative analysis demonstrating the expected output for a few test cases. As part of the final report, start with the introduction of the problem, explain selection of appropriate baselines from the literature study, dataset characteristics, your understanding & interpretation of the approach being implemented and a summary of your implementation. Importantly, along with the results, also include an analysis section at the end describing your findings - what works better and why, and what does not work.

   Along with the report, also include a well-designed, well-illustrated presentation which would include all the points covered above. Note that the presentation content should not be directly copy-pasted from the report, and should be presented properly. This will be used for the project walk-through during evaluation.

   The final submission will consist of these three components:

   (a) Code folder
   Since you will be implementing a SoTA approach on a problem, it will be good to also showcase the same on GitHub.
   (b) Report
   (c) Presentation

   Include all these deliverables in a .zip file named in the format **<TeamNo>-Final.zip**.

We have provided a list of 15 project topics along with their description & expected scope below.

1 NER With Weakly Labeled Data

Reference:
Named Entity Recognition with Small Strongly Labeled and Large Weakly Labeled Data, ACL 2021

Strongly labeled supervised data achieves quality performance, however, owing to practical considerations, models are trained using weakly labeled data which results in underperforming systems. The referred work explores this problem due to weak supervision in the field of NER by proposing NEEDLE - a three-stage training procedure in the face of small supervised and large weakly labeled data. The first stage involves usual continual pretraining on the domain-specific unlabeled data. Crux of the NEEDLE framework lies in the second stage which introduces task-specific pretraining with the aid of weak label completion and noise-aware loss function. In the third stage, finetuning on a strongly labeled dataset achieves good results on the task.

In this project, you'll be implementing this multi-stage NEEDLE training framework. The paper establishes results on two datasets in the E-commerce and Biomedical domain. You can work on any of these datasets. Contrast the performance gains due to the NEEDLE approach against a generally followed BERT+CRF model as the baseline.

2 Adversarial NLI

References:

1. **Adversarial NLI: A New Benchmark for Natural Language Understanding, ACL 2020**
2. **InfoBERT: Improving Robustness of Language Models from An Information Theoretic Perspective, ICLR 2021**

Although Natural Language Inference (NLI) is a well-studied problem in NLP with transformer-based models achieving great results on various benchmarks, it is found that models tend to learn spurious statistical correlations in data and thus do not fully learn the task. The Adversarial NLI (ANLI) dataset was introduced for robust training in the NLI task by introducing a large, difficult dataset created through human intervention. Current SoTA on this dataset is achieved by InfoBERT (the second reference paper) which introduces two regularizers for robust finetuning of models against adversarial attacks.

The project involves implementation of the InfoBERT training objective followed by training and evaluation on the ANLI dataset. Compare the performance against simple BERT-based baselines to show the effectiveness of the robust finetuning and the difficulty of the ANLI dataset.

3 Contract NLI

Reference:
ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts, Findings of EMNLP 2021

In this domain-specific NLP problem, you will be solving the problem of Natural Language Inference (NLI), but on a document-level. Here, your premise is a given contract at hand. Given a hypothesis making some claim about the contract, find out if it is entailed by, in contradiction to, or is neutral to the contract. Also, the next step is to find out the span in the contract which acts as the evidence for judging the hypothesis in case of entailment or contradiction. The work places a novel problem towards the automated review of legal contracts and establishes benchmarks on their publicly available dataset consisting of 607 Non-Disclosure Agreements (NDAs).

Just like the work referred here, show the failure of an existing method in NLI to solve the task. Achieve competitive performance on the dataset by either implementing the SpanBERT approach introduced in this work, or proposing your own modeling solution for the problem.

4 FEVEROUS

Reference:
FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information, NeurIPS 2021

FEVEROUS (Fact Extraction and Verification Over Unstructured and Structured information) is the 2021 edition of the popular FEVER shared task. The dataset introduced with the task contains claims to be verified against pieces of evidence which may be from structured sources such as tables or unstructured sources such as text in Wikipedia articles. The paper introduces a RoBERTa-based baselines for evidence retrieval and verdict prediction, showing the difficulty of the task of evidence retrieval and the relative simplicity of the later task.

The team can implement the baseline approach introduced in the paper for evidence retrieval, and focus on the verdict prediction task for further study. In this task, given the oracle evidence information, the task is to predict the verdict (Supports / Refutes / Not Enough Information). The challenge lies in the imbalanced distribution of the classes. Demonstrate respectable performance of your model over the three classes in the dataset.

5 Answer Sentence Selection

References:

1. **A Study on Efficiency, Accuracy and Document Structure for Answer Sentence Selection, COLING 2020**
2. **TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection, AAAI 2020**

Answer Sentence Selection (AS2) entails selecting the right answer given a candidate set of answers for a question and is a critical problem in the question answering pipeline. The first reference paper looks at a solution engineered solution for a low-cost, quickly trainable approach based on CNNs and RNNs. While this problem has been traditionally modeled as a pointwise binary classification task, this work ranks the candidate answers globally through careful feature interaction while achieving competitive results to a computationally heavier BERT-based baseline. The second referred work introduces TANDA: a simple 2-stage finetuning pipeline for AS2 involving task-specific training (Transfer) on a large corpus for AS2 followed by finetuning on a small domain-specific corpus (Adapt). The pipeline also shows its benefit in providing a stable training over the smaller dataset while also being robust to noisy samples.

This project involves implementing and criticizing the two approaches based on their relative merits and comparing against a BERT-based baseline. You may use the WikiQA dataset for experimentation.

6 Exemplar-Guided Paraphrase Generation

Reference:
Contrastive Representation Learning for Exemplar-Guided Paraphrase Generation, Findings of EMNLP 2021

A work blending the two long-studied problems in NLP of style transfer and paraphrasing, here, we look at paraphrasing a sentence by taking the help of an 'exemplar' for guiding the generation. In other words, the target sentence has to be a paraphrase which matches the same style as the exemplar sentence. Apart from using usual generation loss, the work adds two contrastive learning losses to preserve the content from the source and to match the style of the exemplar. Because the existing datasets for paraphrase generation do not involve exemplars, the authors use an ad-hoc algorithm for obtaining the same.

The project would involve implementing the pipeline in this paper from scratch. The paper describes the architectures made use of for modeling the encoders and decoders. An ablation study involving replacement of the constituent models with reasonable alternatives and their impact on performance can be added for completion.

7 Controllable Text Simplification

Reference:
Controllable Text Simplification with Explicit Paraphrasing, NAACL 2021

Simplification of text has often been attempted by throwing in seq2seq networks to directly obtain the simplification for a given 'complex' source sentence. These methods have turned out to focus on deleting parts from the source to trick around the simplicity by falsely eliminating complex words at the cost of loss of meaning. The referred work proposes an interesting way to achieve text simplification by an intermediate module inspired from the application of linguistic rules. They show the effectiveness of this strategy on the basis of commonly used metrics in simplification as well as qualitative results showing the good quality of simplified outputs.

In this project, you will be implementing the approach on any text simplification dataset of your choice. Contrast it against any simple seq2seq-based modeling.

8 Unsupervised Text Detoxification by Style Transfer & Paraphrasing

Reference:
Text Detoxification using Large Pre-trained Neural Models, EMNLP 2021

This project looks to explore the task of detoxifying content in texts as an interesting combination of textual style transfer and paraphrasing. You'll be implementing the ParaGeDi approach proposed in the reference paper, which combines a generative paraphrase model (for content preservation) with a discriminative model for style transfer (for detoxification). The crux of the modeling lies in combining the supervised training of the paraphraser with the unsupervised training of the style conditioned generator.

After implementing this approach, demonstrate its superiority over baselines which involve pure paraphrasing and pure style transfer.

9 Unsupervised Text Detoxification by Conditional MLM

Reference:
Text Detoxification using Large Pre-trained Neural Models, EMNLP 2021

The task of text detoxification is modeled using a token-by-token replacement approach. Masked Language Modeling (MLM) is a ubiquitous pretraining task for transformer encoder models. Here, following the CondBERT approach in the reference paper, you'll be making an interesting application involving using an encoder architecture for a generative purpose by finetuning the encoder to replace tokens of high toxicity. In masking and replacing, multiple tokens can be replaced in place of a single token. Care needs to be taken that there is no loss of meaning in the process.

There have been previous pointwise editing approaches used for style transfer in general. Use any two of these works as baselines to compare against your implementation of CondBERT.

10 Factual Consistency in Abstractive Summarization

Reference:
Improving Factual Consistency of Abstractive Summarization via Question Answering, ACL 2021

Abstractive summarization has been traditionally evaluated using ROUGE which fails to capture the consistency of facts in the generated summary while purely focusing on n-gram overlaps. The referred work proposed QUALS - a question-answering based efficient framework for automatic evaluation summaries analogous to human evaluation. The work also proposed a contrastive learning based solution for

improving the factual consistency. The approach achieves a good margin of improvement over factual inconsistency metrics (the proposed one as well as previous) while being competitive in performance based on ROUGE scores.

You will be implementing the contrastive learning based approach for improving factual inconsistency in this project. For evaluation, you may use QALS as well as QAGS (a previous work cited in the paper) without implementing them from scratch. Compare against any SoTA abstractive summarization baseline to show improvement in factual consistency while maintaining respectable ROUGE scores.

11 Query-guided Multi-perspective Answer Summarization

References:

1. **AnswerSumm: A Manually-Curated Dataset and Pipeline for Answer Summarization, NAACL 2022**
2. **Exploring Neural Models for Query-Focused Summarization, Findings of NAACL 2022**

This project deals with generating an abstractive summary to provide a single comprehensive answer given multiple possible answers to a question on a community question-answering forum. To break down the problem, query-guided refers to using the question as a guide for summarization, while multi-perspective refers to using the perspective from each of the multiple available answers to produce a single summarized answer. A quality publicly available supervised dataset is provided in the first reference paper.

You will work with this freshly introduced dataset and implement any one of the summarization techniques discussed in the second reference paper on the dataset. Compare the performance of this implementation over any suitable pre-existing baseline (Note: We do not expect you to implement the RL-based approach introduced in the first paper).

12 Consistency-Enhanced Story Generation

Reference:
Consistency and Coherency Enhanced Story Generation, ECIR 2021

Story generation by pretrained architectures such as GPT-2 often struggles to maintain consistency, and often ends up generating content incoherent with the given prompt for generation. The referred paper treads on a 2-step approach to story generation involving generation of a plot outline based on the prompt followed by actual generation of the story. To enhance the quality of sentence flow, the work also proposes using an objective based on discourse markers between adjacent sentences.

Implement the 2-stage story generation with discourse relation modeling (Note: You may not add the coherence consistency-based objective.). Ablate the performance obtained using this pipeline against single-stage story generation and 2-stage generation without using discourse relation modeling. Traditional metrics for evaluating generation won't be of use here. Hence, make use of metrics devised in this work to provide an adequate quantitative analysis. Add some qualitative analysis of your own to the evaluation.

13 Causal News Corpus

Reference:
The Causal News Corpus: Annotating Causal Relations in Event Sentences from News, LREC 2022

Event identification is an important aspect for downstream tasks like question answering and summarization. There exists a temporal relation between events based on their timeline. Finding causality between these events is where the challenge lies since causality is a more psychological than a linguistic concept. However, based on lexical cues and strict annotation schemes, people have developed guidelines for finding causal relations between events. The Causal News Corpus (CNC) is one such dataset containing relations between events that have occured on the news.

The task is to improve the efficiency of the baseline CNC model by pre-training on different Causal corpora like EventStoryLine, PDTB, Because 2.0 etc.

14 Cross-Lingual Question Answering

Reference:
MLQA: Evaluating Cross-lingual Extractive Question Answering, ACL 2020

The referred work introduces MLQA - an evaluation benchmark in question answering for 7 languages along with training data for question-answering in English. The highly parallel nature of the dataset facilitates evaluation of cross-lingual transfer using two tasks as denoted in the paper: XLT and G-XLT. The paper establishes baselines by training on SQuAD 1.1 data (English) and using MLQA for development and testing. BERT and XLM are used for obtaining the cross-lingual representations.

Implement the training followed by zero-shot evaluation for cross-lingual transfer as explained in the paper. Try to improve over the results established. You can experiment over any 2 languages of your choice apart from English.

15 Code-Mixed Machine Translation

Reference:
CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences, Fifth Workshop on Computational Approaches to Linguistic Code-Switching, ACL 2021

Code-Mixed data is massively used by multilingual people in their everyday lives. However, code mixed data can create a problem for another person who is familiar with only one of the two languages. This is where machine translation comes into play. While this is a highly relevant problem for those who speak multiple languages, not much research has been done in the field of code mixed machine translation.

In this specific task, we take Hinglish data and convert it into English language. The baseline approach uses mBART and achieves a BLEU score of 12.2. The challenging part of this task lies in improving the results on the baseline model. You can use pre-processing, post-processing or an altogether different model to produce better results on the same dataset.