NLI with Explanations / Presuppositions

NLP with LLMs - BGU CS - Michael Elhadad - Spring 2025 - HW2

This assignment covers one semantic task (NLI with explanations) and one pragmatic task (analysis of the presuppositions of a sentence).

The expected learning outcomes of this assignment are:

1. Practical understanding of the classification task of textual entailment (aka NLI) with the adversarial NLI dataset (ANLI).
2. Implementation of NLI with LLMs
3. Programming LLMs with DSPy and multi-step inference with LLMs with evaluation and self-consistency tests.
4. Empirical analysis of the effect of providing relevant explanation to improve the performance of NLI.
5. Practical understanding of presupposition and implicatures identification.
6. Benchmark analysis with verification of control parameters and consistency validation on the ImpPres dataset.

Examples demonstrating how to use DSPy are provided in the following notebooks:

- dspy-intro.ipynb
- ai-counter.ipynb
- person-extraction.ipynb

These are just provided as documentation to help get you started - no modifications are expected in these notebooks.

Please attempt to limit the cost of your experiments on x.ai to less than $10 which is the amount precharged per assignment.

1. NLI with Explanations

1.1. Execute the NLI Notebook

Install a local Python environment and run **the notebook anli_baseline.ipynb** to verify you can load the DeBERTa-v3-base model.

This notebook implements a baseline NLI model for the ANLI dataset. The metrics used for NLI are those for a classification task with three labels: entailment, contradiction, neutral and include accuracy, precision, recall and F1.

Implement the part of the evaluation on the ANLI samples that have a non-empty 'reason' field on the 'test' parts of the dataset (there are three such sections test_r1, test_r2 and test_r3).

The answer must be in anli_baseline.ipynb

1.2. Investigate Errors of the NLI Model

Sample 20 errors from the baseline model, and investigate the reasons the model made a mistake.

Report your observations in a table in **anli_baseline.ipynb**.

1.3 Create an LLM Baseline for ANLI

Implement in DSPy a classification model that takes as input the pair (premise, hypothesis) and returns an entailment label. You can optimize the DSPy program on the "dev_r3" partition of the ANLI dataset - with a sample of about 20-100 examples (depending on the budget you want to spend and the improvements you observe on smaller optimization runs). Evaluate the model on the "test_r3" partition of the ANLI dataset (1,200 samples).

Compare the results with the baseline and provide agreement metrics between the two models.

Your answers must be in the notebook "anli_llm_baseline.ipynb".

1.4 Explanation CoT LLM for ANLI

In https://aclanthology.org/2023.findings-eacl.162/ (Kavumba et al, EACL 2023), experiments demonstrate that when LLMs are prompted to classify a pair (premise, hypothesis) as entailment/contradiction/neutral, they perform better if the prompt also requires an explanation to justify the selected label.

In order for the explanation to be helpful, though, it must be a "relevant" explanation, that is, a sentence that is related semantically to the premise and the hypothesis.

You will reproduce this experiment, using two different strategies:

1. Joint prompt: prompt the LLM to produce at once a CoT explanation and a label.
2. Pipeline: prompt the LLM to produce a CoT explanation of the relation (premise, hypothesis) - then, given the explanation, produce a label.

We will compare the produced explanation with two other passages:

1. With the explanation provided by humans in the ANLI dataset (in the field "reason" of the samples).
2. With the (premise, hypothesis) (in order to verify that the produced explanation is relevant to the passage at hand)
3. We will also compare the (premise, hypothesis) passage with the human-provided explanation ('reason').

These comparisons will use the `sentence-transformers` method to rank the similarity between the 3 passages:

1. (premise, hypothesis)
2. human-provided explanation ('reason')
3. predicated explanation See the notebook **sentence-transformers.ipynb** for illustration of how to use this library.

In the DSPy optimization step, we will use these similarity measure to determine whether a predicted explanation is acceptable. You may need to learn a threshold to make this assessment. The DSPy `refine` module will be convenient to implement part of this question https://dspy.ai/tutorials/output_refinement/best-of-n-and-refine/

Compare the two methods - joint prompt and pipeline - on the dev_r3 section of ANLI.

Your answer must be in the notebook 'anli_llm.ipynb'.

2. ImpPres: Entailment for Presuppositions and Implicatures with LLMs

2.1 Explore the ImpPres Dataset

The ImpPres dataset is introduced in *"Are Natural Language Inference Models IMPPRESsive? Learning IMPlicature and PRESupposition"*, Jeretivc et al, ACL 2020, https://www.aclweb.org/anthology/2020.acl-main.768

The code for the dataset preparation and analysis is available in https://github.com/facebookresearch/Imppres/ and the dataset is available on Huggingface datasets under https://huggingface.co/datasets/facebook/imppres

The notebook in `imppres.ipynb` demonstrates how to load the dataset and explore the values.

Complete the notebook with code to create a new dataset that:

- Has all the lines from the presupposition sections of ImprPres
  - ['presupposition_all_n_presupposition', 'presupposition_both_presupposition', 'presupposition_change_of_state', 'presupposition_cleft_existence', 'presupposition_cleft_uniqueness', 'presupposition_only_presupposition', 'presupposition_possessed_definites_existence', 'presupposition_possessed_definites_uniqueness', 'presupposition_question_presupposition']
- Has one more column which is the name of the section:
  - ['premise', 'hypothesis', 'trigger', 'trigger1', 'trigger2', 'presupposition', 'gold_label', 'UID', 'pairID', 'paradigmID', 'section']

2.2 Non-LLM Baseline

In this question, we will explore the performance of the same DeBERTa-v3-base-mnli-fever-anli baseline model on the section of the ImpPres dataset which consists of presuppositions.

Implement this baseline evaluation and report on the classification metrics (accuracy, precision, recall, F1).

Present the results in a table format that shows the metrics for each section of the dataset (each of the 9 presupposition types): 'presupposition_all_n_presupposition', 'presupposition_both_presupposition', 'presupposition_change_of_state', 'presupposition_cleft_existence', 'presupposition_cleft_uniqueness', 'presupposition_only_presupposition', 'presupposition_possessed_definites_existence', 'presupposition_possessed_definites_uniqueness', 'presupposition_question_presupposition' and for all the results together.

The answer should be in imppres_baseline.ipynb.

2.3 LLM Baseline

Implement in DSPy an LLM baseline to perform the NLI classification task on the presupposition data. Show the same results in the previous question. Explore different prompting strategies and corresponding DSPy optimization methods (few-shot examples, CoT).

The answer should be in imppres_llm_baseline.ipynb.

2.4 Explanation CoT LLM for ImpPres and Consistency Validation

In this question, we attempt to improve presupposition identification by exploiting the signal of the paradigms encoded in the ImpPres dataset. For each presupposition, multiple variants of the same pair (premise, hypothesis) are listed where systematic transformations are applied on the sides of the pair (projection through negoation, interrogation, conditional, negation of the presupposition etc).

When optimizing the LLM model, we will use as a reward the measure of consistency across each paradigm - that is, we want the reward to be higher if the model makes consistent predictions across all transformations of the paradigms. This measure of consistency must be combined with the overall accuracy.

It helps to define DSPy examples that correspond to each paradigm (group of 19 samples) to perform this test. The metric on each paradigm is a combination of accuracy of the predictions and consistency over the whole paradigm. Consider whether the LLM should be given all the pairs in the paradigm at once for prediction, or one by one on separate LLM calls. Explain your approach for training and for inference. Make sure to shuffle the samples within each paradigm (because the positions in the paradigm predict the entailment label).

Report on performance on each section and for each of the 19 types of transformation in the paradigm.

Analyze the results of the experiment.

The answer should be in imppres_llm.ipynb.