

NLP 2024

Homework 2 instructions

Adversarial Natural Language Inference

Slides provided by:

- Roberto Navigli
- Luca Giofrè
- Lu Xu
- Luca Moroni
- Tommaso Bonomo
- Alessandro Scirè



What is NLI

Natural language inference (NLI) is the task of determining whether a *hypothesis* is true (**entailment**), false (**contradiction**), or undetermined (**neutral**) given a certain *premise*.

Premise: A man is waiting at the table of a restaurant.

- **Hypothesis 1:** A man waits to be served his food → **Entailment**
- **Hypothesis 2:** A man is looking to order a sandwich → **Neutral**
- **Hypothesis 3:** A man is waiting in line for the bus → **Contradiction**

What is NLI

Note that this task is **sensitive to the sentence order!**

The *premise* P (i.e., all the available context from which we can make assumptions) shall always go first, and the *hypothesis* H (i.e., the claim) will go second.

Moreover, NLI is not necessarily pertained to evaluate the truthfulness of the premise and/or hypothesis, but to **evaluate the soundness of the hypothesis given the premise.**

- H must be in the right relationship with P (the pair must be correctly *labelled*)
- No assumption are made on the truthfulness of P or H

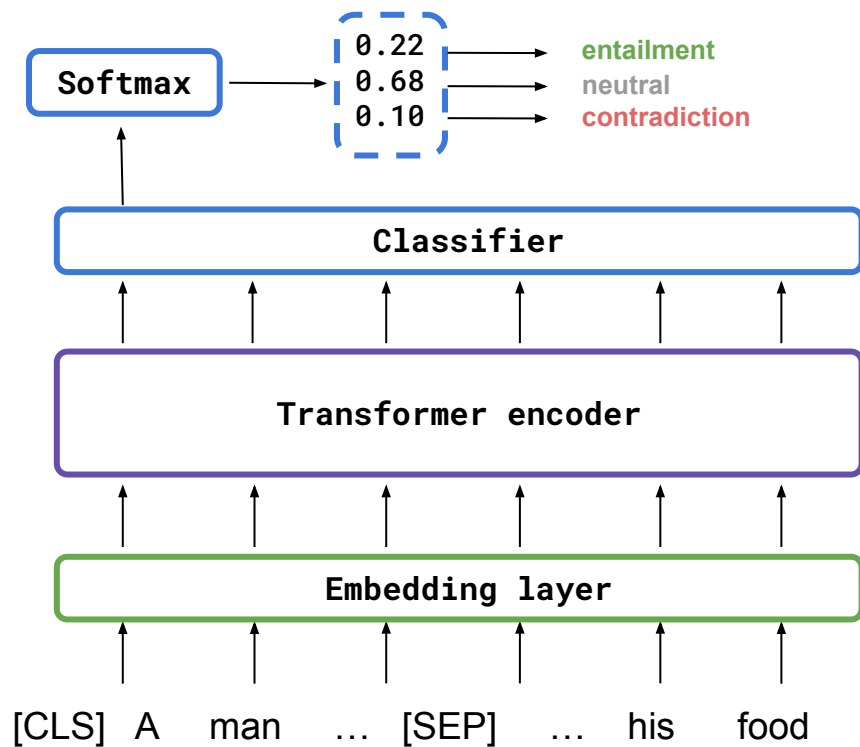
Standard Transformer Approaches to NLI

The standard approach is to treat NLI as a 3-class classification task.

Transformer-based architectures ([DeBERTa](#), [RoBERTa](#)) are the *de facto* standards in NLI due to their performances.

However, on datasets such as [FEVER](#), different approaches have been successful, for example leveraging Semantic Role Labeling or paraphrasing the sentences – have a look at the [task leaderboard](#).

NLI with a Transformer



The model computes softmax probabilities of each of the three classes.

You should format the input premise-hypothesis like so:

“[CLS] *P* [SEP] *H*”

Remember to add [SEP] token between Premise and Hypothesis.

FEVER Dataset: NLI as fact verification

FEVER is a hand-curated dataset for Fact Extraction and Verification built from Wikipedia

It evaluates **claims** against a certain **context** and manually annotates them into true (**Supported**), false (**Refuted**), or undetermined (**Not Enough Info**). For **supported** and **refuted** classes, the annotators recorded the sentence(s) considered as evidence.

[Nie et al. \(2019\)](#) were among the first to use FEVER as a NLI dataset where (context, claim) pairs were considered as (premise, hypothesis).

Data Format

```
{  
  "id": The ID of the sample,  
  "premise": The context to be used in making the inference,  
  "hypothesis": The claim that is made concerning the  
premise,  
  "label": The annotated label for the claim.  
           One of ENTAILMENT | CONTRADICTION | NEUTRAL,  
}
```

Note that the labels have been renamed compared to the original FEVER:

SUPPORTS → **ENTAILMENT**, **REFUTES** → **CONTRADICTION**, **NOT ENOUGH INFO** → **NEUTRAL**

Example of Training Data

```
{  
  "id": 79578,  
  "premise": "Uzbekistan. Uzbekistan is a member of the Commonwealth of Independent  
              States (CIS), Organization for Security and Co-operation in Europe (OSCE),  
              UN, and the SCO.",  
  "hypothesis": "Uzbekistan is a member of the SCO",  
  "label": ENTAILMENT,  
}
```


Adversarial NLI

In this homework you are asked to deal with **Adversarial-NLI**, so modifying the data in a way that the NLI task becomes more complex.

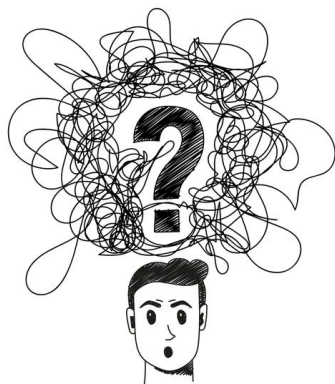
An adversarial sample is a sample generated to be **particularly complex** and intended to **fool models trained on the normal distribution** of the original NLI task.

- Note that adversarial samples are still sound and compliant to the task descriptions

Adversarial test set

Besides a custom version of FEVER, we will provide you an extra test set (the *adversarial test set*) manually curated by a human annotator.

- To perform well on the extra test data, you need a more robust system!



<https://www.facebook.com/photo/?fbid=124431166355362&set=a.124431153022030>

Robustness – where it comes from?

To build a more robust system there are two directions:

1. Change the architecture
2. Change/augment the data



In this homework, you will try both approaches – we will provide you:

1. [a notebook you can use as a starting point for your code](#)
2. [a downsampled FEVER with semantics annotations \(WSD and SRL\)](#)

You can leverage these additional information to create additional data and to find new ways to improve your model on the adversarial test set.

Explicit semantics for more robust NLI



SAPIENZA
NLP

Training set augmentation

As part of this homework, you are asked to generate new samples from a subset of the *training data* of the NLI dataset.

Such new examples have to be **adversarial**, so more complex than the original samples.

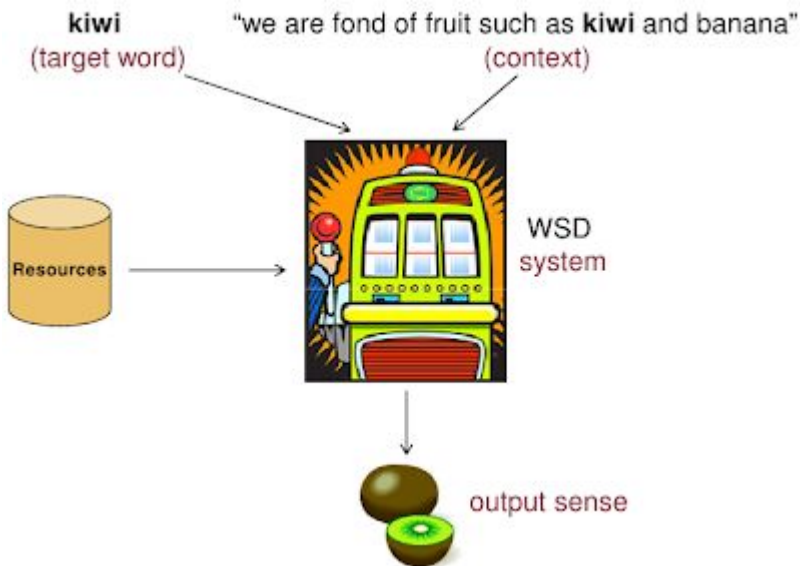
- When altering the original samples, do check that the resulting samples are sound!

In the following slides, we will give you two examples of how you can use semantics to augment the training data, but you are free to devise your own solution

Word Sense Disambiguation

Word sense disambiguation (WSD) is the process of identifying the sense of a word in a sentence.

We will rely on [AMuSE-WSD](#) (Orlando et al., 2021) to provide you with the word sense disambiguation of the senses in the sentences of the training subset that you have to augment.



Semantic Role Labeling

Semantic Role Labeling is the process that assigns labels to words or phrases in a sentence that indicates their semantic role, such as that of an **agent**, **goal**, or **result**.



We will provide you with the annotation of the sentences present in your NLI training subset.

To do this we will rely on [InVeRo-SRL](#) ([Conia et al., 2020](#))

Possible approaches to augment the data

- Given the WSD annotation, you can change the pointed sense with its *hypernym*, *hyponym* or *synonym* (usually, no label change on sample), or by one of its *antonyms* (usually, label do change on sample)
 - A word *w* is the *hypernym* of a target word *x* if the meaning of *w* includes the meaning of *x*, which is more specific (e.g., *animal* is a hypernym of *elephant*)
 - A word *w* is the *antonym* of a target word *x* if the meaning of *w* is the opposite of *x* (e.g., *hot* and *cold*)
 - A word *w* is a *synonym* of a target word *x* if it means exactly or nearly the same as *x* (e.g., *small* and *little*)

You can find these and other information on [WordNet](#), a lexical-semantic resource that organizes nouns, verbs, adjectives, and adverbs in a graph.

- You are free (and encouraged!) to use also other relationships to augment the data.

Possible approaches to augment the data

- **Hypernym substitution:**

- P: The *cat* is running. H: The *cat* is moving quickly. L: **ENTAILMENT**
- P: The *cat* is running. H': The *animal* is moving quickly. L: **ENTAILMENT**

Animal is the hypernym of *cat*, by substituting *cat* with *animal* we obtain another valid pair, which preserves the **ENTAILMENT** relationship. **Be aware that the instances you generate do not invalidate the provided NLI label**

- **Synonym substitution:**

- P: The *kitten* is running. H: The *kitten* is moving quickly. L: **ENTAILMENT**
- P: The *kitten* is running. H: The *kittie* is moving quickly. L: **ENTAILMENT**

Here, *kittie* is a synonym of *kitten*. Again, replacing *kitten* with *kittie* preserves the entailment relationship among the two sentences.

Possible approaches to augment the data

- **Antonym substitution:**

- P: The *tall* man is dancing. H: The *tall* man is moving rhythmically. L: **ENTAILMENT**
- P: The *tall* man is dancing. H': The *short* man is moving rhythmically. L: **CONTRADICTION**

Short is an antonym of *tall*, by substituting *tall* with *short* we obtain a pair that is not entailing anymore. For this reason, we changed the label to **CONTRADICTION**.

Feel free to design other semantic-based transformations that lead to valid NLI instances.

Ideas that are both creative and working will be rewarded!

Possible approaches to augment the data

- **AGENT-PATIENT swap (SRL):**

Given the SRL graph, you can invert the semantic roles (such as `AGENT` and `PATIENT`) to change the meaning of the sample.

If the sentence is complex you can rely on the SRL graph structure, modifying it to create a more complex example, maintaining or changing the class of the training sample.



You are free (and encouraged!) to experiment with other semantic roles to augment the data

Possible approaches to augment the data

While we gave you some possible approaches with WSD and SRL, the adversarial test set is made using **a mix of different techniques**.

- This means that, using semantics alone may or may not prove beneficial; you are encouraged to try other approaches and see what works best!

We suggest you start by modifying the hypotheses, as they are shorter than the premises and so, simpler to modify.

We will give you some other examples in the next slides.

Possible approaches to augment the data

You are encouraged to use a **mix of changes** in combination to make a more complex training set!



- You may need to use external datasets – that's fine, as long as you use them only for data augmentation purposes (cannot train your main model on them)
- You may need to use extra tools, like a POS tagging library, to implement your ideas – that's fine too

Possible approaches to augment the data

- **Numerical inferences:** e.g., inferring dates and ages from numbers
- **Reference inferences:** e.g., coreferences between pronouns and forms of proper names
- **Names inferences:** e.g., leveraging the gender information in the proper names
- **Syntax inferences:** e.g., conjunctions, negations, cause-and-effect, comparatives, superlatives
- **Tricky inferences:** e.g., wordplay, linguistic strategies such as syntactic transformations/reorderings

You are free to experiment, as long as you give reasons for your choices

Architectural changes for a more robust NLI



SAPIENZA
NLP

Experiments

In this homework, you will fine-tune a model to solve plain NLI task over a custom downsampled FEVER dataset and see how the performances changes when testing on the extra test set.



Then, you are asked to augment part of the training data and then check whether this leads to improved performances or not.

Model Architecture

While you can use the additional information provided by the semantic annotations for *anything* you may think of, you could also try to include it to the model by changing the architecture.

- You are not supposed to rethink the Transformer architecture (obviously)...
- ...but you can, for example:
 - add some modules to leverage those information
 - use additional embedding coming from WSD, SRL, POS, etc.
 - use an ensemble of models
 - search the literature for adversarial NLI to get inspiration
 - ...experiment!

When I improve model accuracy by copying from Github an architecture I don't understand.



Submission guidelines

What you will receive

- A subset of FEVER, formatted as previously shown, splitted into `train.jsonl`, `dev.jsonl`, `test.jsonl` and including WSD and SRL information;
 - Available as a Hugging Face dataset at this link:
huggingface.co/datasets/tommasobonono/sem_augmented_fever_nli
- The adversarial test set (`adversarial_test.jsonl`);
- The corresponding WSD and SRL annotations, provided together with the FEVER samples.

WSD example structure

```
"Damon Albarn . Raised in Leytonstone , East London and around  
Colchester"
```

```
{  
  "index": 3,  
  "text": "Raised",  
  "pos": "VERB",  
  "lemma": "raise",  
  "bnSynsetId": "bn:00084108v",  
  "wnSynsetOffset": "2539788v",  
  "nltkSynset": "rear.v.02"  
}
```

SRL example structure (a bit complex)

```
{  
  "tokens": [list of tokens],  
  "Annotations": [  
    "tokenIndex": int,  
    "verbatlas": ...,  
    "englishPropbank": ...,  
  ]  
}
```

What we expect from you

***: don't use a fine-tuned model on NLI datasets!**

Mandatory (up to 30/30)

1. Train a transformer-based model* to solve a plain NLI task on the provided dataset
2. Test your model on the provided adversarial test set
3. Write a comparison on the performances of the model on the two test sets
4. Create an automatic pipeline to generate an adversarial training set

Extra (+1 point)

- 1.A Add architectural changes to point 1 (using the semantic or other stuff inside the training pipeline)

Extra (up to +3 points)

5. Train another model on both provided standard **AND** your adversarial training data and test it **once** on the original test set and **once** on the adversarial test set, comparing the performances as in point 3

Leaderboard!

As we did for homework 1b, the best scoring models and dataset will be given up to +2 points!

Metrics to use:

- Accuracy
- Precision
- Recall
- F1-score



Feel free to add any other metric if they can be useful for your work!

Report Delivery

- You will write an **individual report**, up to **2 pages long** excluding tables, figures and references.
- You should explain your **methodological choices**, both in **modelling** and **data augmentation** strategy, and **analyze** the performance of the two different models, both **quantitatively** and **qualitatively**.
- Delivery deadline: **23:59ish June 18th Italian time (CET)**
- **Late submission penalty**: we will deduct **1 point for each day** after the deadline, up to a maximum of -5 points.
No submissions are accepted beyond 23:59 June 23rd (CET)

Report format

- Use **Latex** to write the report, we suggest [Overleaf](#).
- Use the current **ACL template** (available as [Overleaf](#) template or in [GitHub](#))
 - We penalized some submissions in HW1 for not using this template...
- Each report should consist of **up to 2 pages**; tables, figures and references are not counted in this limit, but they **have to be at the end** of the report
 - Again, we penalized some submissions in HW1 for putting figures and tables between the text, resulting in too many pages

Report format

- We are interested in your reasonings and why you did X instead of Y – this must be your priority when writing the report
 - Any unsupported claim (e.g., saying that the dataset is unbalanced without providing stats) is considered an **error**
 - Technical details are important, but less important than your reasoning, which is the **priority**
 - Plots on data distribution, train metrics etc. are **more than welcome** (and again, please put them at the end of the report!)
 - How to plot? Try [matplotlib](#) or [seaborn](#)
 - Don't want to lose the head on the runs? Try [wandb](#)
- **Remember:** a well-written report with okay results will give you more points than a poorly-written report with stunning performances

Code and data

You must also deliver all the code needed to perform this homework.
You should separate your code into:

1. **Model definition, training and evaluation:**

A single Python script or Jupyter Notebook that implements, trains and evaluates your model on a given dataset

2. **Dataset augmentation:**

A single Python script or Jupyter Notebook that takes the original training dataset and semantically augments it

Code and data

Python files will be automatically launched, so please follow this convention:

1. File for training/testing the model*:

*: use the [argparse library](#)

```
<matricola>-main.py [train,test] --data [original,adversarial]
```

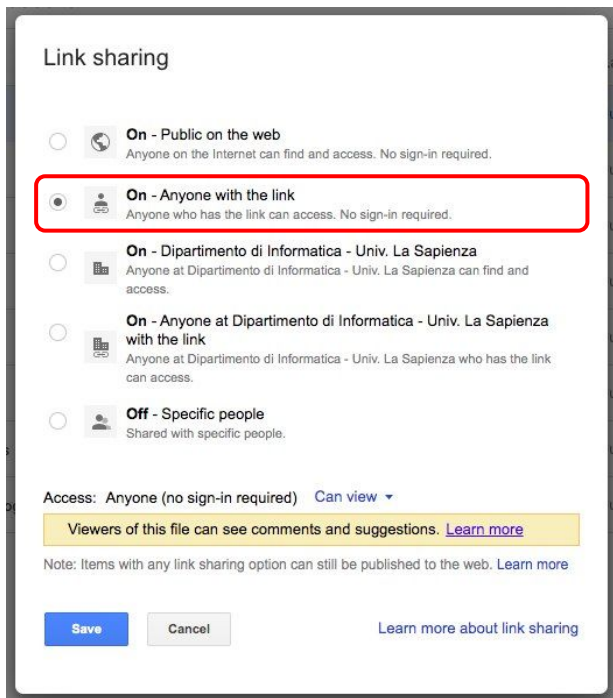
2. File for augmenting the data:

```
<matricola>-augment.py
```

If you need extra flexibility, e.g., you want to submit three different data augmentation pipelines, you surely can!


- Make the main one compliant to the convention and add a README and include a brief description in the “notes” field when submitting


Submission Instructions





The screenshot shows the 'Link sharing' settings for a Google Drive file. The 'On - Anyone with the link' option is selected and highlighted with a red rectangle. Below the options, it states 'Access: Anyone (no sign-in required)' and 'Viewers of this file can see comments and suggestions.' There are 'Save' and 'Cancel' buttons at the bottom.


Link sharing

☐  **On - Public on the web**
Anyone on the Internet can find and access. No sign-in required.

☒  **On - Anyone with the link**
Anyone who has the link can access. No sign-in required.

☐  **On - Dipartimento di Informatica - Univ. La Sapienza**
Anyone at Dipartimento di Informatica - Univ. La Sapienza can find and access.

☐  **On - Anyone at Dipartimento di Informatica - Univ. La Sapienza with the link**
Anyone at Dipartimento di Informatica - Univ. La Sapienza who has the link can access.

☐  **Off - Specific people**
Shared with specific people.

Access: Anyone (no sign-in required) [Can view](#)

Viewers of this file can see comments and suggestions. [Learn more](#)

Note: Items with any link sharing option can still be published to the web. [Learn more](#)

[Save](#) [Cancel](#) [Learn more about link sharing](#)

- Upload the zip on your **institutional** Drive and make it **link-shareable** and **public** to anyone (an automatic script will download it).
- Make sure it is accessible via an incognito page of your browser!
- You have to submit the homework through this [submission form](#) on Google Classroom. You will be asked to fill a form with the requested information and the **link** to the zip you uploaded on Drive.

Plagiarism

We will check for plagiarism both manually and automatically.

It is **not allowed** to:

- Copy from other students;
- Share your code with other students;
- Use ChatGPT or similar systems **for report writing**.

Projects violating any of the above conditions will be desk-rejected

Any doubts?

Use the [Classroom group](#) if you have any questions. Only after you cannot solve your issues in the group, write in English to **ALL the TAs**, so to have a higher reply rate ;)

- Luca Moroni: moroni@diag.uniroma1.it
 - Luca Giofrè: gioffre@diag.uniroma1.it
 - Lu Xu: xu@diag.uniroma1.it
 - Alessandro Scirè: scire@diag.uniroma1.it
- Tommaso Bonomo: bonomo@diag.uniroma1.it

