MNLP Homework 2 Report

Pisano Raffaele
pisano.1959863@studenti.uniroma1.it

1 Base Model

I chose to use the 'deberta-v3-base' model. The smaller version gave worse results, and the larger ones required too much memory and time to train. To address memory constraints, I used a gradient accumulation approach, as the GPU couldn't handle large batch sizes, and using smaller batches slowed down the training process. For similar reasons related to memory and speed, I implemented mixed precision training (fp16=True).
I saved the best model during training based on the lowest eval. loss, as this usually yields better results on test sets compared to saving based on the highest eval. accuracy.
The best results and hyperparameters found are highlighted in the Table 1 and used for comparison in the 'Extra' section. The v5 model have better performance on the validation set, probably it 'weel-fit' it but on both the test set the result of v2 are clearly better.

1.1 Test

As shown in Figure 1, the results on the adversarial test set and the base test set are quite different. The base test-set naturally has better results because the adversarial test-set is designed to be more challenging.
The distribution of samples for each label in both datasets is quite similar and balanced (Figure 1).
The mean and median lengths of the premises are very similar in both datasets, but the lengths of the hypotheses are different (Figure 2). This probably shows that the hypotheses in the adversarial test set are more complex and sophisticated, infact they are longer.

2 Adversarial test-set generation

The adv-set generation involves many different transformations. Here, they are explained in general terms, but in the code, each transformation detailed and has an associated test function.

2.1 Single word modification

The 'augment_sentence' transformation takes a sentence and modifies a word in it with a synonym, hyponym, hypernym, or meronym. It is also possible to specify if the word to modify should be a verb, noun, adjective, etc.
This function is used in combination with other transformations on the sentences to make them more challenging, and it is also used alone on sentences that can't be modified by other transformations. Some examples are shown in Figure 3.

2.2 Inversion of the verb

The transformation done by the 'flip_verb' function inverts the verb in the sentence using its antonym or through manual substitution in specific cases (e.g., "is" to "is not"). The label is also flipped ('CONTRADICTION' to 'ENTAILMENT' and vice versa), while sentences with a NEUTRAL label are skipped. This is because flipping the verb in NEUTRAL sentences could create ambiguity, making it unclear whether the label should be changed to 'CONTRADICTION' or 'ENTAILMENT'.
The 'flip_with_anty' function performs a similar transformation, but in addition to inverting the verb, it searches for the element that 'receives' the action and flips it with its antonym. This results in a double flip, making the sentence even more challenging. In this case, the label is not changed because of the double flip.
Some examples of both are shown in Figure 4.

2.3 Passive form

The transformation performed by the 'passive_form' function changes the sentence from active to passive voice. It checks several things to ensure the transformation is done correctly. For example, it verifies if there is a direct object us-

ing the SpaCy model, if there is only one verb, if the verb can be made passive, and considers the verb's tense and person. At the end of the process, a pretrained model is used to correct grammatical errors for the cases with irregular verbs.
The 'passive_generic' function performs a similar transformation, but it substitutes the agent or the direct object with "Someone" or "something" to generalize the sentence. In this case, the label is set to 'NEUTRAL' because the sentence loses specificity. Some examples in Figure 5.

2.4 Shuffle roles

The 'shuffle_roles' function changes the position of specific roles in the sentence, particularly 'Location', 'Time', and 'Cause', because shifting these roles doesn't affect the meaning of the sentence but makes it more challenging. Some checks are performed to ensure the result is correct.
Examples are shown in Figure 6.

2.5 Question

The 'make_as_question' function transforms the sentence into a question. Some checks are done to apply it only in appropriate cases. The function uses the tense and the person of the verb and builds the question based on that. At the end of the process, a pretrained model is used to correct any grammatical errors, including those with irregular verbs. In this case, the label is set to 'NEUTRAL' because the question, without an answer, doesn't assert anything.
The 'question_answer_modify' function performs a similar transformation but also allows you to add an answer 'Yes' or 'No', and apply the 'augment_sentence' transformation explained earlier. When 'No' is added, it negates the previous hypothesis, so the label is flipped. When 'Yes' is added, confirming the hypothesis, the label remains unchanged.
Some examples of both are shown in Figure 7.

2.6 Transformation loop

The transformation loop iterates over the samples in the dataset and modifies their hypotheses. Each samples is transformed at most once. The transformations are applied starting from the most challenging to the easiest, in order to guarantee that the most challenging have enough samples associated. Each one has a limit on the number of sentences it can modify to ensure a balanced transformation. If a sample cannot be modified by any of the functions, it remains unchanged in the dataset. The results in Table 4 are organized by listing the number of dataset samples modified by each transformation. For example, [Syn-Adj, Passive] refers to the number of samples that were transformed into the passive voice and had an adjective replaced with its synonym. Most of the samples have been modified; only 1,637 remain unchanged. All the transformation have many checks in order to guarantee the correctness of new sentences.

3 Extra

3.1 Architectural changes

I modified the model to also include the POS tags for each word. This because allows the model to better understand the syntactic structure, to better disambiguate word meanings and helps the model generalize better to unseen data. Normally, each token in the batch is passed to the model as:

```
[1, (premise ids), 2,
(hypothesis ids), 2, 0...0]
```

Where 1 and 2 are special tokens, and 0 represents padding. The POS ids are then passed as:

```
[1, (prem. pos ids), 2,
(hypo. pos ids), 2, 0...0]
```

The model is modified by adding:

- Ar layer of embeddings for POS tags of both hypothesis and premise.
- A linear layer that combines the POS embeddings with the hidden states.
- A classifier layer that takes the combination and outputs the logits of the 3 labels.

The results compared with the base model using the same parameters are better. (Table 2)

3.2 Results of adversarial set on model

I trained the base model on a combination of the base dataset and half of the adversarial dataset. This approach was chosen because training with both datasets together resulted in worse outcomes, likely due to overfitting on samples that appeared twice with only little modifications. I kept the entire base dataset to provide the model with a solid 'base' knowledge and added part of the adversarial to improve on challenging cases. The results show improvements compared to training only on the basic dataset, especially on the provided adv. test (Table 3).

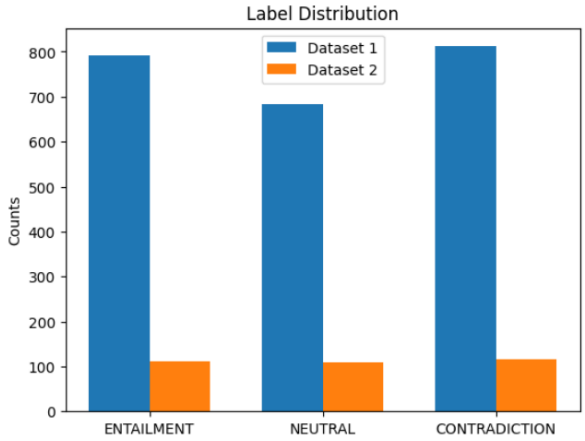

Figure 1: Labels distribution

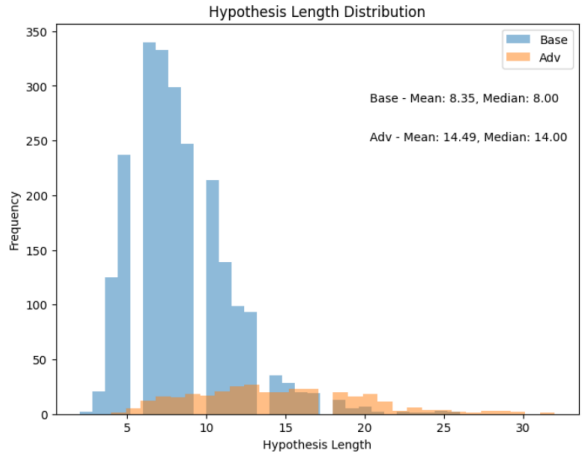

Figure 2: Hypothesis length distribution

```
HYPE - NOUN
There is a movie called The Hunger Games.
There is a product called The Hunger Games.
#####
SYN - NOUN
Ryan Seacrest is a person.
Ryan Seacrest is a individual.
```

Figure 3: Single word modification ex.

```
flip_verb:
   Avril Lavigne is not a voice actor.
   Avril Lavigne is a voice actor.
**FLIP LABEL**
#####
flip_with_anty:
   Caroline Blakiston is on a tv comedy.
   Caroline Blakiston is not on a tv tragedy.
```

Figure 4: Flip verb ex.

```
generalization-Agent:
   Sarah Paulson received a Golden Globe Award nomination.
   A Golden Globe Award nomination was received by someone.
#####
passive_form:
   Marion Cotillard portrayed someone.
   Someone was portrayed by Marion Cotillard.
```

Figure 5: Passive ex.

```
   VHS was released in late 1976 in Japan.
   VHS was released in Japan, in late 1976.
#####
   Tiger Woods has not competed in a golf tournament.
   In a golf tournament, Tiger Woods has not competed.
```

Figure 6: Shift roles ex.

```
question:
   Roman Atwood is a content creator.
   Is Roman Atwood a content creator?
#####
question_augmented-yes:
   The Boston Celtics play their home games at TD Garden.
   Do The Boston Celtics play their place games at TD Garden? Yes.
```

Figure 7: Question ex.

H.par.	batch	epochs	w-dec.	lr	token-len
v1	256	3	0.01	3e-5	512
v2	**256**	**3**	**0.001**	**3e-5**	**512**
v3	256	3	0.001	3e-5	700
v4	128	3	0.001	3e-5	512
v5	256	3	0.001	4e-5	512
Val	Loss	Acc	F1	Prec.	Recall
v1	0.675	0.777	0.770	0.774	0.777
v2	**0.671**	**0.776**	**0.770**	**0.773**	**0.776**
v3	0.693	0.777	0.769	0.774	0.777
v4	0.708	0.767	0.756	0.766	0.766
v5	0.674	0.780	0.774	0.777	0.781
Test	Loss	Acc	F1	Prec.	Recall
v1	0.699	0.763	0.756	0.76	0.763
v2	**0.691**	**0.767**	**0.760**	**0.762**	**0.767**
v3	0.699	0.763	0.755	0.758	0.763
v4	0.706	0.760	0.752	0.756	0.760
v5	0.697	0.762	0.756	0.757	0.763
ADV	Loss	Acc	F1	Prec.	Recall
v1	1.115	0.629	0.630	0.630	0.629
v2	**1.110**	**0.629**	**0.630**	**0.631**	**0.629**
v3	1.138	0.620	0.620	0.620	0.620
v4	0.991	0.623	0.624	0.625	0.623
v5	1.105	0.617	0.618	0.619	0.617

Table 1: Base Model on Validation (Val), Basic test-set (Test) and Adversarial test-set (ADV)

H.par.	batch	epochs	w-dec.	lr	token-len
pos	256	3	0.01	4e-5	512
Val.	Loss	Acc	F1	Prec.	Recall
pos	**0.722**	**0.780**	**0.774**	**0.778**	**0.780**
Test	Loss	Acc	F1	Prec.	Recall
pos	**0.745**	**0.762**	**0.754**	**0.757**	**0.762**
ADV	Loss	Acc	F1	Prec.	Recall
pos	**1.155**	**0.614**	**0.614**	**0.620**	**0.614**

Table 2: Results with pos

H.par.	batch	epochs	w-dec.	lr	token-len
2sets	256	3	0.01	2e-5	512
Val.	Loss	Acc	F1	Prec.	Recall
2sets	**0.647**	**0.779**	**0.774**	**0.776**	**0.779**
Test	Loss	Acc	F1	Prec.	Recall
2sets	**0.666**	**0.765**	**0.760**	**0.761**	**0.765**
ADV	Loss	Acc	F1	Prec.	Recall
2sets	**1.021**	**0.650**	**0.648**	**0.653**	**0.650**

Table 3: Results with adv. set

-	Augment
Syn-Adj	907
Syn-Noun	4492
Syn-Verb	4060
Syn-Adv	236
Hype-Noun	1616
Mer-Noun	68
Tot	11379

-	Flip verb
Normal	5389
Syn-Adj	1102
Syn-Noun	4500
Hype-Adj	9
Hype-Noun	2500
Antyom	5500
Tot	19000

-	Passive
Normal	270
Syn-Adj	179
Syn-Noun	1013
Hype-Noun	212
Hypo-Noun	21
Agent generaliz.	1000
D.O. generaliz.	1000
Tot	3695

-	Shuffle
Normal	1226
Syn-Adj	327
Syn-Noun	1146
Hype-Noun	826
Tot	3525

-	Question
Normal	5044
Syn-Adj-Yes	355
Syn-Noun-Yes	1000
Syn-Adj-No	400
Syn-Noun-No	1000
Syn-Noun	1000
Answer No	1500
Answer Yes	1500
Tot	11799

TOT	49398
Not Modified	1688

Table 4: