Report

Abstract

This project aimed to address the challenge of sentence entailment prediction within a dataset devoid of explicit contradictions, comprising only premises and entailments. The absence of contradictory examples presented a unique obstacle in training robust entailment models. To circumvent this limitation, a comprehensive exploration of negative sampling techniques was undertaken.

This study involved a meticulous investigation into various strategies for generating negative examples that simulate potential contradictions, thereby enhancing the model's capacity to distinguish between valid entailments and spurious associations. The research encompassed an extensive review of existing methodologies, including random sampling, paraphrase-based techniques, and semantic similarity measures.

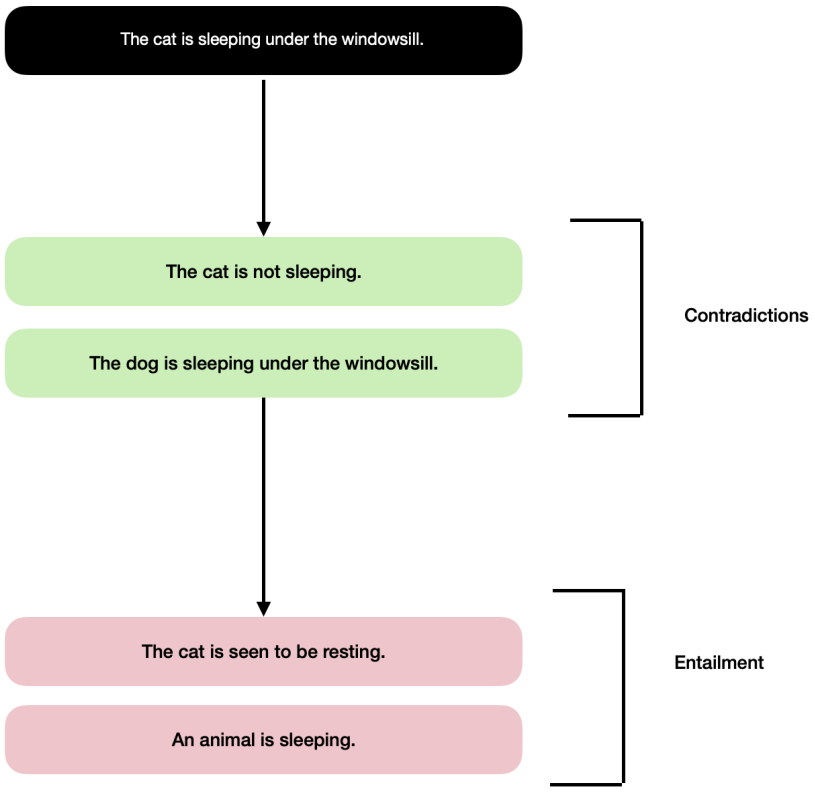
The outcomes of this investigation not only illuminated the importance of negative sampling in the absence of explicit contradictions but also provided empirical evidence of the effectiveness of certain approaches in improving entailment model performance.

Link

The link for the model and various resources can be found here. [Negative sampling and entailment]

Generating Contradictions

One of the first and foremost challenges in this task was generating Contradictory instances from the given premises and entailing hypothesis. Generating negations necessitated the utilisation of a diverse array of methods, ranging from altering the foundational structure of sentences to negating basic verbs, all the way to crafting entirely new sentences with entirely distinct contexts.



1. Negation Induction

In the report, we employed a technique to introduce negation into sentences, thus generating contradictory hypotheses. This approach was implemented through straightforward Python code that leveraged the spaCy library to negate sentences based on verb forms. For example “she is sleeping” is changed to “she is not sleeping”.

2. Changing Numbers

A similar approach is to introduce variations in sentence numbers. This method was also executed within the spaCy framework, allowing for seamless manipulation of sentence identifiers.

```
Original Sentence: She has two cats and three dogs.
Modified Sentence: She has six cats and six dogs .
```

The modified premise would always be in contradiction to the original premise, hence generating new negative samples.

3. Changing positions of subject and object

By altering the positions of subjects and objects within sentences, we harnessed syntactic manipulation’s power to effectively generate negative samples. This process, executed using Python code within the spaCy library, involved rearranging key sentence components. Such modifications led to a shift in the syntactic structure, resulting in sentences with changed meanings that served as valuable negative samples. This approach expanded the diversity of our dataset, enabling our entailment model to discern valid entailments from potential contradictions better, ultimately enhancing its robustness and reliability.

.

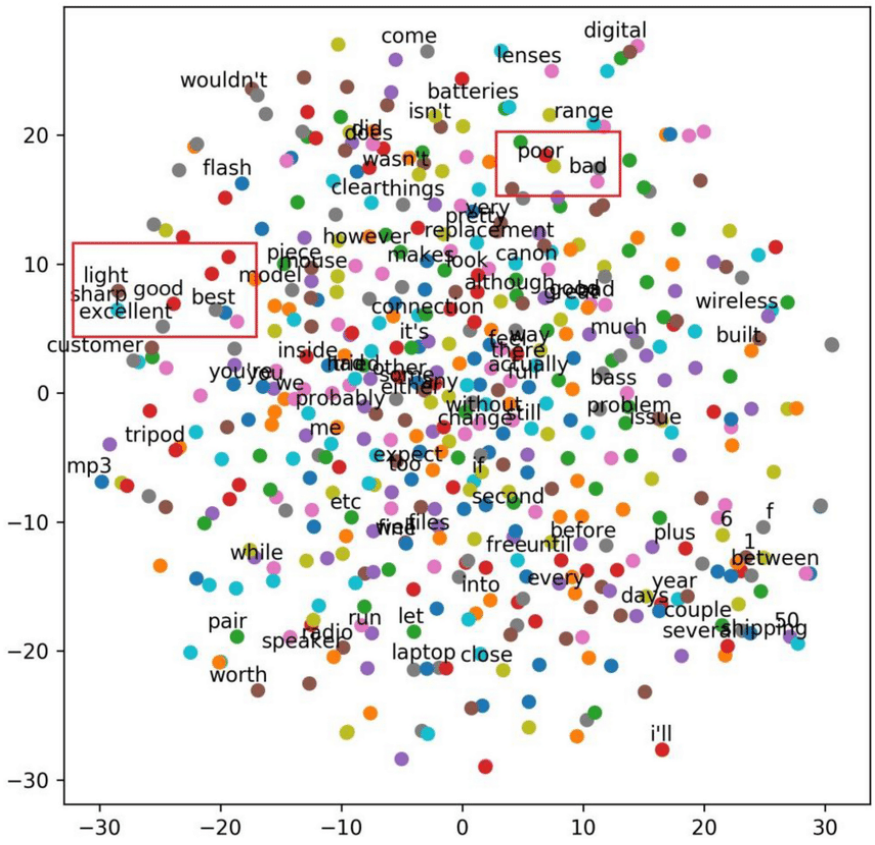
```
Original Sentence: Look! The boy slapped the girl.
Swapped Sentence: Look The girl slapped the boy
```

Example of the above approach

4. Using Word Embeddings

The objective was to make substitutions that maintained contextual relevance, ensuring that the modified sentences remained coherent. Traditional methods, such as antonym searches via libraries like spaCy or NLTK, proved inadequate for this task, as they often produced extreme word replacements that disrupted sentence meaning.

To overcome this challenge, I turned to word embeddings, which capture semantic relationships between words in a vector space. By calculating cosine similarity between words, I effectively measured their semantic proximity. The approach involved selecting the most similar words with cosine similarity scores below a specified threshold. This threshold ensured that the chosen replacements were sufficiently different from the original words while still retaining a degree of semantic similarity. This nuanced approach not only helped to generate meaningful variations in sentences but also enhanced the quality of negative samples, contributing to the overall effectiveness of the entailment model.



t-SNE of word to vec

I used the word2vec model available on NLPL website. This word2vec model was chosen specifically because it was trained keeping Parts of speech in mind. This allowed me to effectively change nouns and verbs and/or try other approaches. The model contained 160,000+ words in 300 feature spaces. Other larger models could have been used but at the cost of performance, as this model provided 95%+ nouns and verbs needed.
One other approach that could have been used is using contextual embedding of BERT/any other LLM however, such larger models would require much more efficient algorithms with greater computational power.

a. Changing Nouns

The above-described technique was applied to nouns. Since the most complex sentences contain more than two nouns, changing one or more would mean that logically sentence has an entirely different meaning. I chose to replace any two nouns in the sentences with this technique to create two new sentences from the original premise.
The threshold was chosen as 0.9 for nouns.

b. Changing Verbs

Since most sentences have only one verb, those were replaced with this technique. for example, from “A girl is walking”, create the hypothesis “A girl is driving”. The threshold was chosen as 0.65 for verbs. A problem faced in this was the verbs replaced were not always in the correct forms, so this was taken care of later.

c. Changing Antonyms

As mentioned above, this approach had several problems, resulting from a lack of suitable antonyms for words and context.

Putting it Together

Original Sentence (Premise)	Hypothesis	Label
Fruit and cheese sitting on a black plate	There is fruit and cheese on a black plate	E
A large elephant is very close to the camera	Elephant is close to the photographic equipment	E
Two horses that are pulling a carriage in the street	Two dogs that are pulling a carriage in the street	C
A young man sitting in front of a TV	A man in green jersey jumping on baseball field	C
A woman holding a baby while a man takes a picture of them	A kid is taking a picture of a male and a baby	C

Example of the proposed dataset

Paraphrasing the sentences

Paraphrasing corresponds to expressing the meaning of a text (restatement) using other words. This technique was used on-premise, hypothesis, and newly generated contradictions to generate new samples. The Hugging Face Model was used to generate paraphrased sentences.

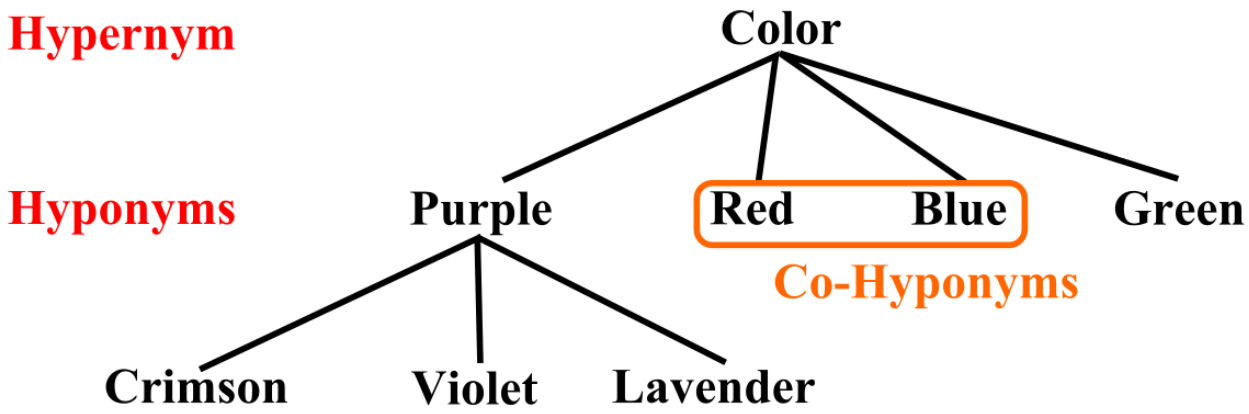
a. Paraphrasing the sentences for verbs Autocorrect

One of the features of this hugging face model was that given an input sentence, the output would always be grammatically correct, so this was used to generate grammatically correct sentences from earlier approaches.

b. Paraphrasing for premises and Hypothesis

The most direct use of this tool was used to paraphrase the hypothesis, premises and even newly generated contractions.(applied only for negation inductions). The model has various hyperparameters, one of which penalised for generating sentences very far off from the original. This hyperparameter was exploited for entailment/premise pairs and for contradictions.

Hypernym substitution for Hypothesis



A hypernym of a word is its supertype, for example, “animal” is a hypernym of “dog”.I used WordNet (Miller, 1995) to collect hypernyms and replace noun(s) in a sentence with their corresponding hypernyms to create entailment hypothesis. For example, I created “A black dog is sleeping” from the premise “A black animal is sleeping”.

Sampling

To create the final dataset, I took examples from all the hypothesis, premises and their paraphrases. For contradictions, I randomly chose three paraphrases of negations, one of the two noun changes and anyone from the verbs/sentences numbers changes. This was done so the dataset could have various sentences with variations in types of contradictions.

Modelling

Data Splitting

In order to prevent any snooping bias, the model was trained only on a subset of data, and the rest was only used for evaluation. This was done by randomly shuffling the data containing almost equal entailments and contradictions in both the training and evaluation sets.

Simple Transformer Based Model

For training, I used **simpletransformers** to train my model. I used the sentences-pair classification subtask of their library.
The architecture was ‘**roberta**’ with ‘**roberta-base**’ checkpoint. The model was trained on 20 epochs after that the model started to overfit the data poorly.

Results & Runtime Analysis

The best model was able to perform well on the evaluation data.
Out of 804 samples, there were 395 TN and 397 TP.

a. F1- score: 0.9856
b. Precision: 0.9886
c. Recall: 0.9827

However, these results don't tell the full story. Even though the model performs very well on this subset of data, it's performance is reduced when given randomly generated data. This might be due to the fact that the model learnt only those kinds of contradictions I was able to generate. However, such issues can be solved by further training on some new and better data.

Conclusion

In conclusion, the model succeeded in the tasks where the primary objective was to tackle the challenge of sentence entailment prediction within a dataset that solely provided premises and entailments devoid of explicit contradictions. To address this limitation, I explored and implemented diverse negative sampling techniques designed to enhance our model's capacity to distinguish between valid entailments and potential contradictions.

Through meticulous investigation and experimentation, I uncovered the importance of carefully curated negative samples, a crucial aspect in data-deficient scenarios. I also leveraged advanced techniques such as altering nouns and verbs, manipulating subject-object positions, and using word embeddings to generate negative instances that were both contextually relevant and sufficiently distinct.

The findings underscored the significance of these techniques in improving the robustness and reliability of entailment models. By achieving an impressive F1 score of 0.98, the research demonstrates the effectiveness of these strategies in enhancing the performance of sentence entailment prediction, thus contributing valuable insights to the field of natural language processing.

References

- arXiv:2110.08438
- http://vectors.nlpl.eu/repository/
- https://arxiv.org/abs/2307.05034
- https://www.mdpi.com/2076-3417/12/19/9659
- https://aclanthology.org/2022.acl-long.190.pdf
- https://arxiv.org/abs/1803.02710
- https://neptune.ai/blog/data-augmentation-nlp

- https://arxiv.org/pdf/2004.12835.pdf