Modeling Semantic and Physical Plausibility Making Use of Binary Classification

Miriam Segiet and **Li Lin** and **Huirong Tan**
Institut für Maschinelle Sprachverarbeitung, University of Stuttgart
{firstname.lastname}@ims.uni-stuttgart.de

Abstract

Modeling plausibility is a common task in natural language processing even though natural language understanding requires commonsense reasoning. This reasoning ability is easier for humans than for models due to the lack of world knowledge and reasoning. We present three methods to model this plausibility based on binary classification which show that there are some difficulties for models to classify plausibility of events.

1 Introduction

Humans have the ability to discern properties of natural language such as classifying sentences as nonsensical or semantically plausible. This ability is useful for commonsense reasoning or knowledge acquisition which is not as straightforward for models since plausibility is part of world likelihood and human intuition is mirrored in plausibility as well (Porada et al., 2021). Therefore, it is more difficult for models to correctly model plausibility. There is a difference, however, between semantic and physical plausibility. Semantic plausibility relates to whether a statement or action makes sense given the semantics or meaning of the words. Physical plausibility, however, does not only relate to semantics but also takes into account whether the event is plausible in general. Therefore, it takes more commonsense knowledge to classify physically than semantically plausible sentences. Wang et al. (2018) state that *man swallow candy* is semantically plausible and also the preferred option while *man swallow paintball* is not even though it is physically plausible. This emphasizes that it is crucial for a model to learn modeling physical plausibility as well.

To address the issue of modeling semantic plausibility, in this work, we present three different models for the classification of binary plausibility. We furthermore evaluate and analyze the performances to point out future directions related to our work.

2 Datasets

For the classification task at hand, the two datasets PAP and PEP-3K are used for training and/or testing.

2.1 PAP

The PAP dataset is a dataset for physical and abstract Plausibility (Eichel and Schulte Im Walde, 2023), where *text*, *original_label*, and *label* are the main components. The original PAP dataset's train-dev-test split contains a total of 2160 items. In it, the text is composed of s-v-o (subject-verb-object) triples, the *original_label* is based on dataset construction, and the *label* is from human annotation. It shows that for the same text, the *original_label* and *label* may have different judgments regarding their semantic plausibility. However, to reduce the impact of unreliable annotations on training, we adopted the filtered dataset, which only contains 1733 entries.

In the raw annotation of the dataset, each word of the s-v-o triples was labeled with the abstractness of the event(Eichel and im Walde, 2023), such as a-m-c, which represent highly abstract, mid-range, and highly concrete respectively.

2.2 PEP-3K

The PEP-3K dataset, the Physical Event Plausibility ratings (Wang et al., 2018), is a crowdsourced, two-column dataset with more than 3,000 instances in total. The whole data is divided into a development, a test, and a training file with a distribution of 80% of the data used for training and 10% for development and testing each. In all files the first column represents the binary plausibility labels ('1' means *plausible*, '0' means *implausible*), annotated for physical plausibility, while the second column contains the labeled text. In total, there is almost the same amount of *plausible* and *implausible* labels. For further information on dataset statistics refer to appendix B.

Texts are represented as s-v-o (subject-verb-object) triples where no grammatical agreement (verbs appear in infinitive forms e.g., *man poison dog*) or punctuation is present and all words are presented in lower-case letters. Since the text is presented in this form, the binary part–of–speech (POS) tags are not correctly assigned by the nltk (Bird and Loper, 2004) library and most of the tag pairs are two nouns (for further information see appendix C).

2.2.1 Data Adaption

In order to investigate grammaticality effects on the prediction, the PEP-3K and binary PAP dataset are adapted in two configurations to see whether the abstractness of words in PEP-3K makes a difference.

The first configuration adds the third–person singular *s* to the each verb in the PEP-3K data. In order to adhere to the given grammar rules, verbs ending in *ch*, *sh*, *s*, *z*, and *x* will receive the appendix *-es* instead of *-s*. On the other hand, PAP is adapted such that the third–person singular endings are removed from each verb adhering to the same rules as in the first configuration.

3 Models

The binary classification in this work is performed with three different models[1] as described in the following subsections.

3.1 Keras

The entire classification of the PEP-3K dataset is based on TensorFlow's (Abadi et al., 2015) Keras API (Chollet et al., 2015) where a sequential model is built. The model itself consists of an embedding layer, two bidirectional Long Short-Term Memory (LSTM) layers, and a dense layer.[2] In the training step, both training and validation data are included for their intended use. In case of additional data, data from both PEP-3K and PAP is jointly used for training and validation by simply concatenating both values. The data is tokenized using the Keras Tokenizer base class and fit on the sentences from the training data only to ensure test instances are not contained in the embedding phrases.

3.2 RoBERTa

The first approach to binary classification on the PAP dataset is to utilize RoBERTa (Liu et al., 2019) as the cornerstone of the model. RoBERTa (Robustly optimized BERT approach) is a pre-trained language model built on the Transformers architecture. Given that our task revolves around semantic plausibility prediction and our input comprises text sequences, RoBERTa's specialized model variant, RoBERTaForSequenceClassification [3], is well-suited for this purpose. RoBERTaForSequence-Classification adds an additional classifier layer on top of the RoBERTa model, which is used to classify input text sequences. During training, the model adjusts its parameters to maximize the classification performance of the input text sequences, aiming to maximize classification accuracy or other metrics.

Models were trained using the standalone PAP dataset and the combined PAP plus PEP-3K dataset as training data respectively, and evaluated on the PAP test set. Both models achieved their optimal states at training epoch 3. For further experimental setup and details on code execution, please refer to README and code content for pap_RoBERTa at GitHub.

3.3 Llama2-7B

This model is based on Llama2 with 7 billion parameters on Hugging Face[4] and performs classification on PAP and PEP-3K datasets. Model implementation mainly includes the following steps:

1. *Load the model*: Llama2 is a text generation model. In order to use it for classification tasks, LlamaForSequenceClassification[5] is used to load the pre-trained model, which adds a linear layer on top of Llama2 as a sequence classification head.
2. *Quantitative model*: The GPU resource used to fine-tune Llama2 is GeForce RTX 4060 Ti. In order for Llama2 to run under limited GPU resources and speed up training, the model was quantized to reduce the precision to 8 bits.

[1] The code is available at GitHub via https://github.tik.uni-stuttgart.de/st185873/Modeling_Semantic_Plausibility.

[2] The detailed model configurations can be found in Appendix D.

[3] https://huggingface.co/docs/transformers/model_doc/roberta

[4] https://huggingface.co/meta-llama/Llama-2-7b

[5] https://huggingface.co/docs/transformers/en/model_doc/llama#transformers.LlamaForSequenceClassification

3. *Configure PEFT (Parameter Efficient Fine-Tuning) method*: Lora(Low-Rank Adaptation)[6] is applied in the query and value layers of the attention mechanism of the model transformer. The final number of fine-tuned parameters is 4,194,304, which only accounts for 0.063% of the original parameter number, making the fine-tuned model more efficient.
4. *Process data*: Load the tokenizer from Llama2 to tokenize the triplet text, and set it to the right side padding. Convert the datasets into dataloaders with a batch size of 3.
5. *Training parameter settings*: The learning rate decreases linearly from 1e-4 to 6e-5. The training epochs are 3.

4 Results & Analysis

In the following, the three models are evaluated and results are analyzed.

4.1 Keras

The Keras model is evaluated using precision, recall, and F1 scores on the labels predicted by the model. Results are obtained on the same model configurations independent of training or test data. Table 1 shows the results for different data configurations. The results show that additional data increased performance on the PEP-3K data significantly. Nevertheless, for all results, the additional data did not change the results significantly. In addition to including data from the binary PAP file, adding data from the multi–class PAP file was done as well. Since this incorporation did not change the results significantly, results are not displayed here.

Data		Metrics		
train	**test**	**P**	**R**	**F1**
P3	P3	.69	.75	.72
P3 + PAP		.75	.78	.77
P3 + a-PAP		.71	.78	.74
a-P3 + PAP	PAP	.53	.80	.64

Table 1: Results for different data configurations where all configurations contain the PEP-3K (P3) or adapted PEP-3K[7](a-P3) data and are combined with either PAP or the adapted version of PAP (a-PAP). Results are based on precision (P), recall (R), and f1-score (F1).

Analyses showed that, on average, there are 50 false positives and 35 false negatives predicted by the model for all configurations. Interestingly, the false negatives from the P3 + a-PAP configuration are almost identical to those of the P3 + PAP and only differ by four phrases. An in–depth analysis of the wrongly predicted phrases did not show any regularities where the model has problems in prediction. However, the false positives show that subjects in wrongly predicted instances are most often animate (e.g., *child, lion, or woman*) while the objects are inanimate (e.g., *pebbles, tree, pillow*). The same applies to subjects and objects in false negative predictions, where subjects are most often animate and objects most often inanimate. This is counter–intuitive since it is expected that sentences with an animate subject and an inanimate object (e.g., *child drink water* as opposed to *food poison bird* where the first intuition would dictate that the bird was poisoned by some*body* instead of some*thing*) are easier to classify since this conforms to common sentences.

4.2 RoBERTa

Table 2 compares the performance of best models trained on PAP datasets against those incorporating both PAP and PEP-3K. Integrating PEP-3K improves precision but reduces recall, with the overall accuracy remaining unchanged. This indicates that while PEP-3K may help the model to be more accurate in its positive predictions, which could be due to PEP-3K enhancing the model's specificity, it slightly compromises the model's ability to identify all positive cases, as evidenced by a slight decrease in the F1 score.

Data		Metrics			
train	**test**	**Acc**	**P**	**R**	**F1**
PAP		.81	.86	.78	.82
PAP + P3	PAP	.81	.89	.75	.81

Table 2: Performance Metrics of RoBERTa Models. The results are evaluated based on Accuracy (Acc), Precision (P), Recall (R), and F1-score (F1).

A comparative analysis was conducted to investigate the misclassified sentences from two optimally selected models, as shown in Table 6. Each model produced 33 misclassified samples, with 19 samples being misclassified by both, suggesting a degree of ambiguity in these sentences that challenges

[6] https://huggingface.co/docs/peft/en/task_guides/token-classification-lora

[7] The adaption process is described in section 2.2.1.

the models' distinction capabilities. This issue particularly affects the models' ability to accurately identify positive examples with ambiguous characteristics. The analysis demonstrates significant challenges for both models in correctly identifying instances, with the PAP model showing a comparatively more balanced performance.

During the analysis of misclassified samples, it was noted that the degree of data abstraction impacts prediction outcomes. When data tend towards high levels of abstraction or concreteness, misclassification likelihood decreases. However, the presence of both highly concrete and abstract elements in data, while affecting accuracy, is not the main cause of predictive errors. The significant adverse impact on training outcomes is observed when training data cannot be clearly classified as highly concrete or highly abstract, highlighting the role of inherent ambiguity.

A thorough analysis of the PAP dataset revealed that approximately one-third of the data exhibit discrepancies between *original_label* and *label* (see Table 7). Despite using *original_label* for training, considering *label* values during evaluation showed that about half of the misclassified samples were deemed correctly classified within *label*. This suggests that the process of dataset construction and generating sentences marked as implausible might inadvertently produce sentences that are semantically plausible, identified during the human annotation process. The discrepancies between *original_label* and *label* might partly be due to certain s-v-o triples extracted from real-world information appearing semantically illogical in isolation, contributing to the differences observed.

4.3 Llama2-7B

The Llama2-7B pre-trained models were fine-tuned on three datasets: PAP, PEP-3K, PAP+PEP-3K. Only the data of the trian file was used for training and only the data from the test file was used for testing.

Classification performance As shown in table 3 , Llama2's classification performance on the PAP dataset is better than human annotation label(the consistency ratio with the original label is less than 70%). In terms of PEP-3K performance, the accuracy is better than the optimal result of 74% of the CONCEPTMA method in the 2021 paper(Porada et al., 2021), and the optimal result of 76% of the injecting world knowledge method in the 2018 paper(Wang et al., 2018).

Data		Metrics			
train	**test**	**Acc**	**P**	**R**	**F1**
PAP	PAP	**.868**	**.986**	.768	.864
PAP + P3	PAP	.862	.874	**.874**	**.874**
P3	P3	.863	.828	**.915**	.870
PAP + P3	P3	**.886**	**.888**	.882	**.885**

Table 3: Performance Metrics of Llama2-7B Models.

Adding additional training datasets impact The results show that adding PEP-3K did not obviously improve the accuracy of PAP prediction, but adding PAP to the PEP-3K test increased the accuracy by 2.3%. What emerged in both dataset tests was that adding additional training dataset slightly improved the F1-score, which shows that it makes the model more robust.

PAP abstractness impact To analyze the impact of word abstraction, we compared the distribution of unigram, bigram, and trigram abstractness tags frequencies in samples misclassified by the model with the original test data. The results show that the model fine-tuned with the PAP dataset is not sensitive to the abstraction of the word itself. It is also possible that the sample size was insufficient to show a difference. An interesting finding is that after fine-tuning the PEP-3K data, the frequency of highly concrete word in the samples misclassified by the model decreased. The possible reason is that PEP-3K mainly contains concrete words. Specific data and analysis can be found in Appendix F.

5 Conclusion

In this work we present three different models for the binary classification of plausibility based on the PAP and PEP-3K datasets. The models are based on Keras, RoBERTa, and Llama2-7B. For both the Keras and Llama2-7B models the incorporation of more data improved the overall results while this is not the case for the RoBERTa model. The results suggest that it is still difficult for the model to correctly assign labels due to the lack of proper commonsense reasoning the world knowledge which is given for humans.

6 Future Work

Steps for future work in the keras set up include making use of pretrained embeddings to further investigate the effects of grammaticality where it is expected that sentences from PAP are easier to correctly classify since they adhere to given grammar rules. Furthermore, it would be interesting to see how the predictions change if sentences are presented in passive voice (e.g., *dog poisoned man*) where the plausibility classification needs to be done on a higher level since on the first glance the exemplary sentence seems implausible.

For future work on PAP dataset, it is worth considering utilizing *label* as the correct standard for training and prediction. However, this approach necessitates first addressing the issue of unbalanced label distribution we observed in Figure 4 across the dataset before further discussion.

Furthermore, to enhance the reliability and robustness of the model, it might be prudent to conduct repeated training sessions, such as five iterations, and calculate the average of these outcomes. This approach aims to mitigate the effects of training contingencies and ensure a more stable and dependable model performance.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. `https://keras.io`.

Annerose Eichel and Sabine Schulte im Walde. 2023. Amendment to eichel and schulte im walde (2023): A pap case study on amt vs. non-amt annotation.

Annerose Eichel and Sabine Schulte Im Walde. 2023. A dataset for physical and abstract plausibility and sources of human disagreement. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 31–45, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ian Porada, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2021. Modeling event plausibility with consistent conceptual abstraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1732–1743, Online. Association for Computational Linguistics.

Su Wang, Greg Durrett, and Katrin Erk. 2018. Modeling semantic plausibility by injecting world knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 303–308, New Orleans, Louisiana. Association for Computational Linguistics.

A Contributions

Miriam implemented all methods related to the PEP-3K dataset in the folder `pep-3k_classification` in the repository on GitHub. As general parts, she wrote the abstract, introduction, and conclusion. In addition, she wrote the introduction to the PEP-3K dataset and the model implementation and result analysis of the Keras model (part of the aforementioned folder) as well as the paragraph in the future work section related to Keras. She furthermore wrote the ‘introductions’ to the subsections.

Huirong was responsible for completing three data analyses on the PAP dataset (word frequency, annotation difference, and label distribution) as well as model training based on RoBERTa. She prepared the initial layout of the entire presentation slides and the display based on the above results, and wrote about the introduction to PAP dataset, the RoBERTa-related model and corresponding results & analysis, and future work in general and on PAP dataset in the report.

Li Lin completed part of the data analysis of PAP and PEP-3k, including basic analysis, tokens frequency analysis, part-of-speech distribution analysis (only displayed in the GitHub repository), and abstractness tags analysis of the PAP dataset (displayed in presentations and document). And completed all code, testing, result analysis, presentation and document writing parts related to the Lama2-7B model.

B Dataset Statistics PEP-3K

The general statistics of the PEP-3K dataset are displayed in table 4. This table shows the split of 80/10/10 for training, test, and development files, respectively. It furthermore shows that there is an almost equal amount of plausible and implausible labels.

File	num rows	PL	IMPL
dev	306	153	153
test	307	153	154
train	2449	1222	1227

Table 4: General dataset statistics. PL represents number of plausible instances, IMPL the number of implausible instances.

Table 5 displays further statistics related to the phrases in the file. Where it is visible that the average lengths of words and tokens per sentence is similar over files and therefore this property will not affect wrong predictions.

File	TW	UW	AWL	SWL
dev	918	330	4.79	14.4
test	921	315	4.75	14.2
train	7347	561	4.72	14.1

Table 5: Dataset statistics related to phrases and words. TW represents total number of words, UW the unique number of words. AWL displays the average length of words in each file (per word) while ASL displays the average length of tokens per sentence.

C POS–Tag Distributions PEP-3K

The POS–tag distributions show that the most appearing pair of POS–tags is the *NN NN* pair. This shows that the problem in the file is related to the infinitive form of the verb which is most often interpreted as a noun.

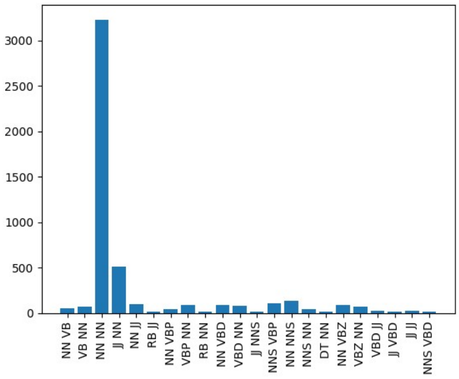

Figure 1: Distribution of bigram POS–tags appearing at least ten times in train.csv.

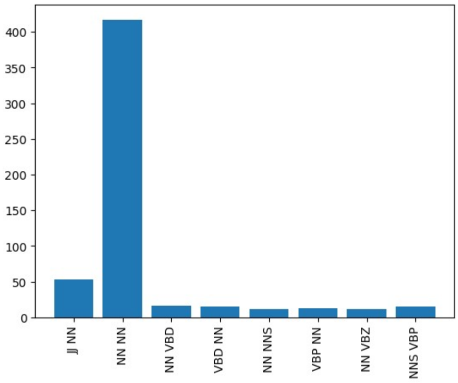

Figure 2: Distribution of bigram POS–tags appearing at least ten times in dev.csv.

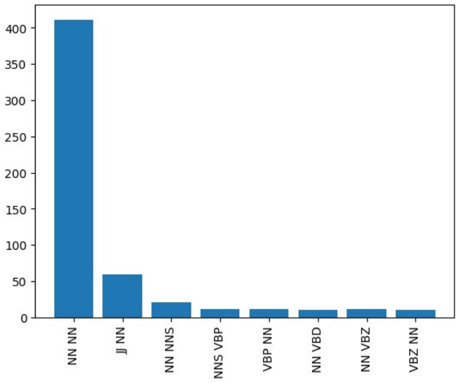

Figure 3: Distribution of bigram POS–tags appearing at least ten times in test.csv.

D Keras Model Configuration

The best–performing model configurations are based on an embedding layer with a 32–dimensional output, a first bidirectional LSTM layer with 256 dimension, a second bidirectional LSTM layer with 64 dimensions, and a uni–dimensional dense layer with sigmoid activation function, an L1L2 kernel regularizer, and an L2 bias regularizer. Between all layers, a dropout of 0.3 is used. The model is furthermore compiled with a binary cross entropy loss and the adam optimizer. Moreover, 35 epochs and a batch size of 64 are used.

E Misclassification Analysis, Annotation Inconsistency and Label distribution result - PAP

	TMis	FN	FP	OH	WM	NI	O	WMR
PAP	33	21	12	3	16	10	4	.70
PAP + P3	33	24	9	2	21	5	5	.75
Shared Mis	19	15	4	1	10	5	3	.71

Table 6: Misclassification (Mis) metrics for RoBERTa models. The data in the table from left to right represents: total misclassified entry (TMis), False Negative instances (FN), False positive instances (FP), instances that only labeled with highly concrete or highly abstract (OH), instances that labeled with one or more mid-range (WM), instances with no information on abstractness (NI), other instances that labels are combined both highly concrete and abstract (O), and Ratio of instances that labeled with mid-range (WMR). When calculating WMR, O was excluded from TMis.

Dataset	Entries	Incons	Incons Ratio
Train.csv	1386	477	0.344156
Dev.csv	173	57	0.329480
Test.csv	174	57	0.327586

Table 7: Inconsistencies (Incons) in Original Label vs. Label on PAP dataset. Around 1/3 of the data have discrepancies in labeling.

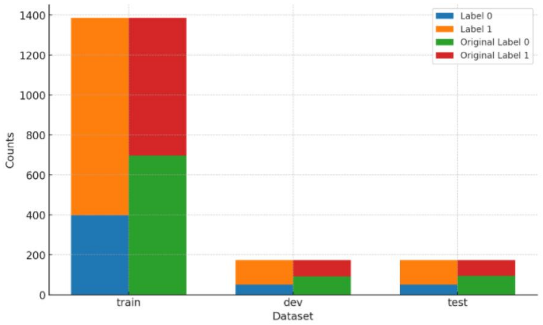

Figure 4: Label distribution in Train, Dev, Test Dataset on PAP dataset. Across all three datasets, the count of label 1 in *label* is generally higher than that of label 0, which suggests a potential bias towards label 1. The distribution of *original_label* is more balanced across the three datasets.

F Abstractness Impact Analysis - PAP

Take the Llama2-7B models that were last fine-tuned with the PAP and PAP+PEP-3K datasets and test it on the PAP dataset. There were 23 and 24 misclassified results respectively. To observe whether fine-tuned Llama2 is affected by abstraction in semantic plausibility classification, the frequencies distribution of unigram, bigram, and trigram abstractness tags for these misclassified samples was compared with the original PAP test data.

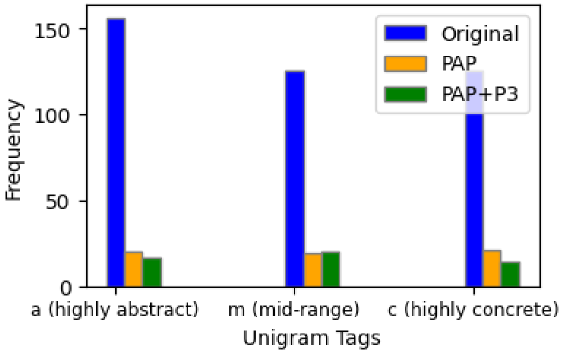

Figure 5: Comparison of unigram frequencies. Shows the abstractness distribution of three data, test data (Original), misclassified data of the Llama2-7B model fine-tuned with PAP (PAP) and misclassified data of the model fine-tuned with PAP+PEP-3K (PAP+P3). In the test data, the number of highly abstract words is slightly higher. In the misclassified results of the PAP fine-tuning model, there is not much difference in the number of three abstract categories. In the data misclassified by the model fine-tuned by PAP+PEP-3K, highly abstract and highly concrete words are less misclassified than those by the PAP fine-tuned model.

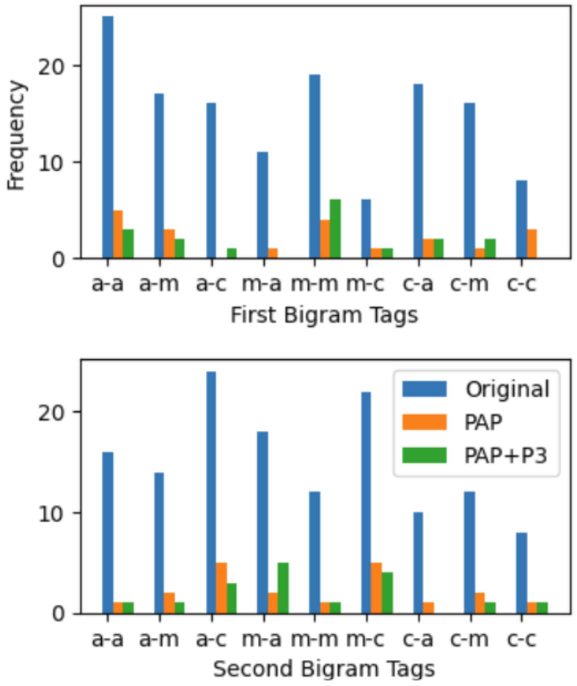

Figure 6: Comparison of bigram frequencies. The legend labels have the same meaning as in Figure 5.

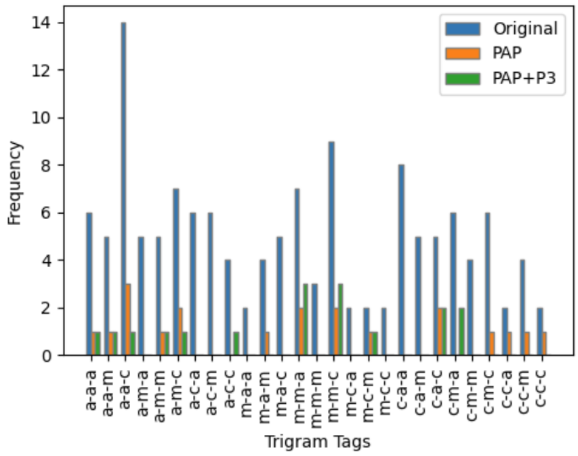

Figure 7: Comparison of trigram frequencies. The legend labels have the same meaning as in Figure 5.

Overall, as shown in the three result figures[8], no obvious patterns were found from the misclassified results of the PAP fine-tuned model, indicating that the model is not sensitive to abstraction. However, this analysis is only based on more than one hundred test data of the PAP dataset, so the abstractness analysis may be limited by the samples size that are not enough to show the impact.

Another finding is that after adding the fine-tuning dataset of PEP-3K, among the samples misclassified by the model, the reduction of highly concrete is the most obvious. For example, it can be seen from Figure 7 that c-m-c, c-c-a, c-c-m, c-c-c all have a small number of samples misclassified by the PAP fine-tuned model, but not in the PAP+PEP-3K fine-tuned model. Whether this change is caused by the additional PEP-3K training dataset can be studied in future work.

[8] The drawing code comes from ChatGPT https://chat.openai.com/ and manual adjustment.