A Squeeze of LIME, a Pinch of SHAP: Adding the Flavor of Explainability to Sentence-Level Commonsense Validation

Mirta Moslavac, Roko Grbelja, Matej Ciglenečki

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{mirta.moslavac,roko.grbelja,matej.ciglenecki}@fer.hr

Abstract

Addressing the lack of transparency in AI systems' decision-making regarding commonsense knowledge, we propose leveraging explainable artificial intelligence (XAI) techniques. Inspired by SemEval-2020 Task 4, we refine the subtask of commonsense validation to quantify the probability of a single sentence exhibiting common sense. Our approach involves fine-tuning a pretrained transformer-based model and using Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) for model explainability. Our study investigates the impact of individual words on commonsense validation decisions and contributes to the broader objective of improving the reliability and trustworthiness of commonsense models. We observe that the two different explainability techniques yield different explanations as to how individual words attribute to the decisions on the commonsense nature of a sentence.

1. Introduction

Commonsense knowledge is crucial for human intelligence, allowing us to understand the world, infer intentions, and predict outcomes. However, incorporating commonsense knowledge into artificial intelligence (AI) remains challenging (Davis and Marcus, 2015). Previous research has made progress in tasks like question-answering (Talmor et al., 2018), natural language inference (Zellers et al., 2018), and visual commonsense reasoning (Zellers et al., 2019). Although transformer-based models have been successful in addressing these challenges, they are opaque in their decision-making process, unlike humans who possess individual and inherent common sense (Smith, 1995). Therefore, it is important for humans to understand the reasoning path of these models.

Despite the advancements, there is a research gap in explicitly validating and quantifying the common sense of individual sentences. Inspired by SemEval-2020 Task 4: Commonsense Validation and Explanation (ComVE) (Wang et al., 2020), specifically subtask A (Validation), we build upon the task organizers' work in commonsense validation. However, we modify the original task formulation by focusing on quantifying the probability of an individual sentence being commonsense or not. We also adapt the provided dataset for a better fit with our task requirements.

Using eXplainable artificial intelligence (XAI) techniques, our goal is to examine the impact of individual words on the predicted degree of common sense of a sentence. To achieve this, we fine-tuned a pretrained transformer-based neural language model called Decoding-enhanced BERT with Disentangled Attention (DeBERTa) to create a commonsense validation model. To gain insights into the model's explainability, we utilized two techniques: Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017). By pursuing this approach, our aim is to contribute to the broader objective of enhancing model explainability through the understanding and application of commonsense knowledge.

2. Background

2.1. Commonsense Knowledge

Incorporating commonsense knowledge into AI tasks is essential for bridging the gap between human understanding and machine intelligence. It serves as a foundation for reasoning, decision-making, and effective communication, enabling AI systems to gain a deeper comprehension of the world, improve predictive accuracy, and engage in intuitive interactions. However, challenges arise due to the implicit nature, contextual reliance, and variability of commonsense knowledge across time and cultures (Smith, 1995). Traditional machine learning models struggle to grasp intuitive concepts like cause-and-effect relationships and social norms, as formalizing and representing these concepts is complex. While previous approaches employ external knowledge graphs (Speer et al., 2017), they face limitations in coverage and scalability. Recent advancements have explored contextualized embeddings to enhance the integration of common sense into ML models (Bouraoui, 2022).

2.2. Explainability

Explainability is crucial in XAI for human understanding and trust in AI system decision-making. A similar term, interpretability, involves presenting model outputs in a human-understandable manner, emphasizing cause-and-effect connections (Doshi-Velez and Kim, 2017). In contrast, explainability focuses on understanding the internal mechanisms of an AI system and the processes influencing its decision-making (Linardatos et al., 2020). While an interpretable model enhances understanding, it does not provide insight into the model's inner logic. This paper specifically investigates explainability to analyze the impact of input features on commonsense validation decisions.

XAI techniques can be classified based on their emphasis and model compatibility (Linardatos et al., 2020). Two categories are model-specific methods designed for specific model families and model-agnostic methods applicable to any model. Techniques can also be categorized as local or global, with local methods providing explanations for individual model decisions using simpler interpretable models based on neighboring context (Ras et al., 2022). In our study, we focus specifically on model-agnostic and local techniques, namely LIME and SHAP, to explore explainability in a black-box model for sentence-level commonsense validation. These posthoc methods analyze pre-trained models, as opposed to intrinsic methods integrated into the explained model. Note that we do not investigate the combination of LIME and SHAP known as Kernel SHAP in this paper (Lundberg and Lee, 2017).

2.2.1. LIME

Local Interpretable Model-agnostic Explanations (LIME) is an explanation technique proposed by Ribeiro et al. (2016). It provides local interpretability for ML models by sampling neighboring instances of input and generating perturbations to observe the impact of input features on the output. A simpler model, such as a linear one, is trained using these perturbations to approximate the predictions of the original model. By analyzing the weights of this simpler model, LIME showcases the contribution of each feature, even for textual input where it can be a single word in a sentence, to the output of the original model by comparing distances from the perturbations to the input instance.

2.2.2. SHAP

The SHapley Additive exPlanations (SHAP) framework, introduced by Lundberg and Lee (2017), provides an explanation mechanism similar to LIME, attributing contributions to each feature of input towards a specific prediction. By leveraging SHAP values based on Shapley values from cooperative game theory, SHAP constructs a simplified local approximation model. SHAP values quantify the change in the expected model prediction when a specific feature is considered, offering a measure of the feature's impact on the prediction. Unlike LIME, SHAP values ensure desirable properties such as local accuracy and consistency in the explanations provided.

3. Related Work

The previously mentioned SemEval-2020 Task 4 (ComVE) (Wang et al., 2020) consists of three subtasks, two discriminative (A, B) and one generative (C). For our approach, we are inspired by subtask A, but other tasks also tackle similar commonsense issues. Transformer-based pre-trained language models (PLMs), such as BERT, have become the standard for solving ComVE subtasks. They have demonstrated state-of-the-art performance in SemEval 2020 Task 4 (Fadel et al., 2020). Zhang et al. (2020) achieved 97% accuracy on subtask A using K-BERT and knowledge graph triples. Huy et al. (2022) recently surpassed the state-of-the-art results using ensemble learning with RoBERTa, DeBERTa, and ELECTRA, highlighting the effectiveness of output aggregation from individual models.

While Liu et al. (2023) addressed a related problem, there is a lack of work specifically focused on predicting the degree of sentence commonsense. Most recent research has concentrated on SemEval 2020 Task 4, approaching the problem as binary classification. In this paper, inspired by subtask A, we propose a novel approach that brings explainability to the task of commonsense validation.

Table 1: Example of the original dataset format. Label 0 indicates that the sentence with index 0 is non-commonsense.

Index	Instance	Label
0	*He walked his fish.,He walked his dog.*	0
1	*She eat a lot of food.,Food eat her a lot.*	1

Table 2: Example of the modified dataset format. Label 1 indicates that the sentence is commonsense, 0 not.

Index	Instance	Label
0	*He walked his fish.*	0
1	*He walked his dog.*	1
2	*She eat a lot of food.*	1
3	*Food eat her a lot.*	0

4. Method

4.1. Model

For the model, we chose a pre-trained DeBERTaV3 trained on 160GB data (He et al., 2021). We opted for the smallest version of the pre-trained model (DeBERTaV3$_{\text{small}}$) which has 44M parameters. To perform the downstream task of binary classification, we use DeBERTaV3's features in combination with a linear head classifier.

4.2. Dataset

The original dataset contains 11,997 sentence pairs with binary labels indicating which sentence violates common sense. Each pair consists of two sentences with the same structure but differ in a few words (Table 1). The dataset is divided into training (10,000 pairs), test (1,000 pairs), and validation (997 pairs) subsets.

To align with our modified problem formulation, we made changes to the original dataset while maintaining the predefined train/test/validation split. Each sentence pair was split into two separate instances: one representing the non-commonsense sentence labeled as 0, and the other representing the remaining sentence as 1, indicating common sense alignment. This modification effectively doubled the dataset size, resulting in 20,000 instances for training, 2,000 for testing, and 1,994 for validation (Table 2).

5. Experiments

5.1. Packages

To train the model and perform the analysis, we used the following Python packages: PyTorch [1] and transformers [2] for training the model, NLPAug [3] for sentence augmentations, shap [4] and lime [5] for commonsense analysis, and tensorboard [6] for experiment tracking.

[1] https://github.com/pytorch/pytorch

[2] https://github.com/huggingface/transformers

[3] https://github.com/makcedward/nlpaug

[4] https://github.com/slundberg/shap

[5] https://github.com/marcotcr/lime

[6] https://github.com/tensorflow/tensorboard

5.2. Experiment Details

We used the dataset and split defined in 4.2.. We fine-tuned the model with binary cross entropy as the loss function, AdamW optimizer (Loshchilov and Hutter, 2017) with default PyTorch parameters, and a cosine learning rate scheduler accompanied by a linear warm-up scheduler with a 0.1 ratio. Models were trained between 3 and 5 epochs (60,000-100,000 steps). During the training, we evaluate the model on the validation set every 500 steps and save the model with the highest Macro F1. Every model was trained on a single RTX 3060 mobile. It took us a total of 9 hours to train 28 different models.

5.3. Augmentation

We used NLPAug to randomly swap words during training. To avoid turning common sense into non-common sense, this augmentation was applied only to sentences that were already non-commonsense, as it is likely that they would remain so after the augmentation. Each example in the batch had a 50% chance of being augmented. If applied, it swapped 30% of the words in the sentence. For instance, if a sentence had 14 words, 4 of them would be swapped.

5.4. Hyperparameter Optimization

Multiple experiments were conducted to find the best hyperparameter combination. The optimal set was determined based on the maximum Macro F1 metric on the validation set. Augmentation, learning rate, and model freezing strategy had the most impact on Macro F1, while other hyperparameters remained constant during training.

Among the 28 trained models, the 12 best-performing models had BERT freezing disabled, while the rest had it enabled. Out of the 5 best models, only one (2nd best) utilized data augmentation described in 5.3.. Attempting to fine-tune only the linear classification head by freezing the parameters of DeBERTaV3 resulted in inferior results. Therefore, the final model was fine-tuned without freezing any parameters. The hyperparameters for our best model are as follows: BERT freezing set to FALSE, no data augmentation applied (Augmenter = None), the learning rate of 5×10^{-6}, and a training duration of 5 epochs while the metrics on the validation set are the following: Accuracy = 0.862, Macro F1 = 0.863, ROC-AUC = 0.862.

The word swapping augmentation did not improve the model's performance. The augmented sentences could further compromise the sentence's grammar, making them even less meaningful. This may cause the model to learn both correct and incorrect grammar instead of learning the overall semantics of the sentence. Augmented sentences that lack sense and potentially have grammatical errors may inadvertently push the model to focus more on distinguishing between correct and incorrect grammar.

5.5. Baselines

In evaluating our model, we compared it to two baselines: random labeling and a logistic regression model using TF-IDF vectorization. Results for the baselines and our model are shown in Table 3.

Table 3: A baseline model comparison of metrics obtained on the validation set.

Baseline	Accuracy	Macro F1	ROC-AUC
Random	0.500	0.500	0.500
Logistic Regression	0.574	0.583	0.574
Our Model	0.862	0.863	0.862

6. Explainability Analysis and Discussion

Due to the short form of the paper and the local nature of the XAI techniques in question, meaning that an individual sentence can be analyzed at a time, we will showcase an explanation analysis on a pair of sentences. This pair was randomly selected from the original dataset and mapped to our modified dataset, where each sentence is treated as a separate instance. The utilization of paired sentences ensures that one sentence aligns with common sense expectations and that the other deviates from them.

To determine the commonsense probability of each example, we utilize our commonsense validation model. The rounded-off probabilities are also provided in the LIME visualizations, as depicted in Figures 1 and 2.

During LIME explanation generation, up to top 10 most influential features, corresponding to words in the sentence, are considered. The neighborhood size for learning the linear model associated with a specific sentence is limited to 150 due to hardware constraints. It is crucial to acknowledge that these explanations may not fully capture the intricate complexity and variability of the underlying model due to the limited neighborhood size. As a result, the reliability of the explanations might be somewhat compromised.

In the process of SHAP explanation generation, the computation of SHAP values is performed for each sentence individually. Subsequently, the resulting positive and negative feature contributions are visually presented for both possible classifications: LABEL_0 (or NCS, representing non-commonsense) or LABEL_1 (or CS, representing commonsense). The base value in the visualizations represents the prediction which would occur if if we did not know any features to the current output, while the output value $f_{LABEL0/1}(inputs)$ refers to the actual prediction (Lund-

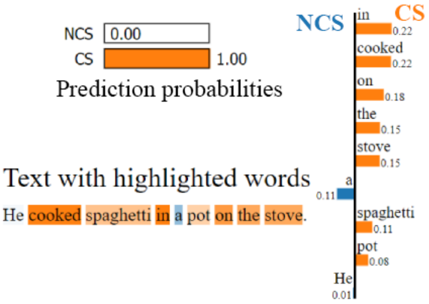

Figure 1: LIME commonsense input explanation. NCS indicates the non-commonsense class, CS commonsense.

berg and Lee, 2017). The individual word SHAP values add up into the final output value, the SHAP value of the input. The positive (red) SHAP values attributed to the words indicate that the presence of the specific word increases the likelihood of the observed label, whereas the negative (blue) suggests the opposite.

6.1. Commonsense Example

The provided sentence, *He cooked spaghetti in a pot on the stove.* is categorized as CS in the modified validation dataset. The commonsense validation model assigns a prediction probability of 0.9986 for the CS class.

The LIME explanation in Figure 1 highlights the relevance of words like *in*, *cooked*, *on*, *the*, *stove*, *spaghetti*, and *pot* for the commonsense interpretation. Among these words, *cooked* and *in* have the highest coefficient value of 0.22, indicating their crucial role in conveying the emerging commonsense nature of the sentence. These words emphasize the act of preparing spaghetti and its specific positioning in relation to the pot. While *spaghetti* and *pot* have slightly less significance, they still contribute to the overall context understood by the model. Conversely, *a* and *He* have weaker associations with the non-commonsense interpretation, as reflected by their coefficients of 0.01 and 0.11, respectively.

The SHAP analysis, shown in Figure 3, reveals a different explainability perspective on the sentence's commonsense nature. The base and the output value for both classes are the same, which suggests that the sentence features captured by the SHAP values have minimal impact on the model's final prediction. Nonetheless, we can observe that the words that positively contribute the most to the NCS label are *spaghetti* (0.374) and *cooked* (0.373), serving as strong indicators of the sentence being classified as NCS. Somewhat expected, the same words carry the most contribution in the possibility of CS classification, but now in a negative manner. The opposing logic applies to the words *in*, *He*, etc., due to yielding the most negative contributions for NCS classification and positive for CS.

6.2. Non-Commonsense Example

The provided sentence *He cooked a pot inside of spaghetti.* is categorized as NCS in the modified validation dataset. The commonsense validation model assigns a prediction probability of 0.9997 for the NCS class.

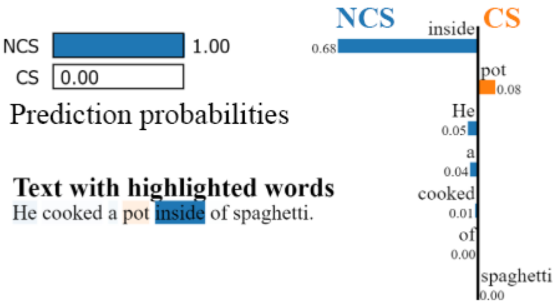

Figure 2: LIME non-commonsense input explanation. NCS means the non-commonsense class, CS commonsense.

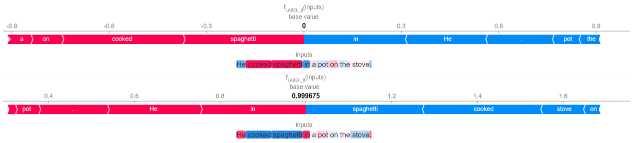

Figure 3: SHAP explanation of a commonsense sentence. The top representation indicates contributions to non-commonsense classification, bottom commonsense.

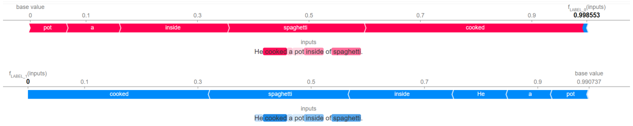

Figure 4: SHAP explanation of a non-commonsense sentence. The top representation indicates contributions to non-commonsense classification, bottom commonsense.

The LIME explanation is given in Figure 2. The word *inside* has a coefficient of 0.68, indicating its strong contribution to the non-commonsense interpretation. This suggests that the idea of pot placement inside spaghetti is a key factor in determining the non-commonsense nature of the sentence. Other words in the sentence do not show strong contributions to the non-commonsense classification. The word *pot* demonstrates a slight connection to the commonsense interpretation.

The SHAP analysis, depicted in Figure 4, demonstrates that all words in the sentence strongly contribute to the NCS classification while having a negative impact on the CS classification. This aligns with the expected classification of the sentence, indicating that the presence of these words serves as reliable indicators of the NCS label.

7. Conclusion

In conclusion, this paper addresses the challenge of commonsense validation at the sentence level, aiming to quantify the probability of an individual sentence being commonsense or not. Our approach builds upon existing research on commonsense knowledge and XAI, addressing a research gap in explicitly quantifying the common sense of individual sentences.

To achieve this, we fine-tune a transformer-based PLM and employed XAI techniques, namely LIME and SHAP, to analyze the model's explainability. By focusing on the explainability of our model and analyzing the impact of individual words on commonsense validation, we contribute to the broader objective of enhancing commonsense model transparency and trustworthiness. Our findings reveal that employing different explainability techniques to our proposed model results in distinct explanations regarding the contribution of individual words to the determination of a sentence's commonsense nature.

References

Zied Bouraoui. 2022. *Inducing Commonsense Knowledge Using Vector Space Embeddings*. Ph.D. thesis, Université d'Artois.

Ernest Davis and Gary Marcus. 2015. Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Communications of the ACM*, 58(9):92–103.

Finale Doshi-Velez and Been Kim. 2017. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*.

Ali Fadel, Mahmoud Al-Ayyoub, and Erik Cambria. 2020. JUSTers at SemEval-2020 Task 4: Evaluating Transformer Models against Commonsense Validation and Explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 535–542, Barcelona (online), December. International Committee for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving Deberta Using Electra-Style Pretraining with Gradient-Disentangled Embedding Sharing. *arXiv preprint arXiv:2111.09543*.

Ngo Quang Huy, Tu Minh Phuong, and Ngo Xuan Bach. 2022. Autoencoding Language Model Based Ensemble Learning for Commonsense Validation and Explanation. *arXiv preprint arXiv:2204.03324*.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2020. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23(1):18.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2023. Vera: A General-Purpose Plausibility Estimation Model for Commonsense Statements. *arXiv preprint arXiv:2305.03695*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*.

Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. *Advances in neural information processing systems*, 30.

Gabrielle Ras, Ning Xie, Marcel Van Gerven, and Derek Doran. 2022. Explainable Deep Learning: A Field Guide for the Uninitiated. *Journal of Artificial Intelligence Research*, 73:329–397.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Barry Smith. 1995. Common Sense. In Barry Smith and David Woodruff Smith, editors, *The Cambridge Companion to Husserl*, pages 394–437. New York: Cambridge University Press.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. *arXiv preprint arXiv:1811.00937*.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 Task 4: Commonsense Validation and Explanation. *arXiv preprint arXiv:2007.00236*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. *arXiv preprint arXiv:1808.05326*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From Recognition to Cognition: Visual Commonsense Reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Yice Zhang, Jiaxuan Lin, Yang Fan, Peng Jin, Yuanchao Liu, and Bingquan Liu. 2020. CN-HIT-IT.NLP at SemEval-2020 Task 4: Enhanced Language Representation with Multiple Knowledge Triples. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 494–500, Barcelona (online), December. International Committee for Computational Linguistics.