Key-point-analysis and explanations for quantitative text analysis

Daniel Schroter and **Hannes Schroter** and **Yuyin Lang**
daniel.schroter@tum.de
hannes.schroter@tum.de
yuyin.lang@tum.de

Abstract

An increasing fraction of public debates takes place online. The ability to automatically capture key ideas and opinions in online communication can be a game changing for politics and businesses. Argumentative texts can often be reduced to some core ideas, called key points. Our work addresses the research question presented in the KPA shared task 2021 of matching arguments to key points. We propose two different models. The first model combines a transformer based denoising autoencoder (TSDAE) with a siamese neural network. The second one enriches a transformer based siamese neural network with additional features like part-of-speech tags. Our evaluation shows and performance improvement of 2% to 5% compared to the best performing model in prior research. On the other hand those models are characterised by a highly complex architecture and low interpretability. We implement different methods to answer why certain argument key point pairs receive a high similarity score. Furthermore, we propose a simple approach on applying LIME to siamese neural networks (SNNs) which is novel in literature. Hence, our twofold contribution to research comprises the presentation of new models for solving the KPA shared task and a new approach of applying LIME to SNNs.

1 Introduction

With the rise of the internet and social media platforms like Twitter and Facebook, an increasing fraction of public discussion, debates and the expression of opinions takes place online. The ability to automatically capture central ideas, identify key opinions and summarize core topics present in the constant stream of online communication would be a valuable resource for decision makers in politics and businesses. Deciding about the development of a certain product, analysing trends in society like an increasing consciousness about sustainability or capturing the impact of political campaigns can be game changing use cases for organisations (Bar-Haim et al., 2020). Hence, strong research streams have developed in the last couple of years dealing with computational argumentation (Lawrence and Reed, 2020), opinion analysis (Levy et al., 2018) and semantic textual similarity (Agirre et al., 2013). One part of the challenge has been presented at the EMNLP 2021, a conference on empirical methods in natural language processing, described as Key Point Analysis (KPA shared task). The problem builds upon the work of Bar-Haim et al. (2020) and aims at summarizing arguments with key points that capture the essential information of the argument. As a first step, solutions are required to appropriately match arguments to key points that are already given. The ArgKP dataset (Bar-Haim et al., 2020) contains approximately 24k argument key points pairs in 28 controversial topics like “Home Schooling should be banned”. The test dataset contains 3 additional topics that are not present in the original training dataset. The arguments are a subset of the IBM-Arg-30kArgs dataset, and the key points have been created by a professional debater and annotated to the matching arguments. Hence the dataset contains argument and key point pairs together with a label if they match. We present two siamese neural network models that achieve a precision that is 2% to 5.7% higher compared to the best performing model in prior research with respect to our evaluation. Our first model is characterised by the combination of an unsupervised pretraining method called transformer based denoising autoencoder (TSDAE) and a siamese neural network. The second model is characterised by a more complex siamese neural network architecture that is enriched by manually engineered features like part-of-speech tags. Both models rely on the sentence transformer architecture Roberta. However, such models are characterised by a complex deep learning architecture leading to results that are difficult to explain. With the power of such deep learning models the demand for interpretability increases. Apart from the plain predictions, in many use cases some rational behind a prediction is required to answer why a model made a certain decision. Research on the explainability of siamese neural networks seems to be emerging but is still rather thin. Consequently, we apply different explanation techniques and propose a simple but novel idea on how local interpretable model-agnostic explanations (LIME) can be implemented in the case of siamese neural networks. Our contribution to literature is (1) the combination of TSDAE with a siamese neural network, (2) a siamese neural network architecture incorporating manually engineered features and (3) a novel approach on applying LIME to siamese neural network for the stake of explainability.

2 Prior Research

The KPA shared task basically builds upon the work and the dataset of Bar-Haim et al. (2020) who originally presented the problem of matching argument to corresponding key points. Within a broader context the task is associated with the more general research stream of semantic textual similarity. Since the KPA shared task was published as a competition there have been 18 different teams proposing solutions for the problem. Friedman et al. (2021) provide an overview about the different solution concepts. The evaluation metric is mean average precision (Map). The details of the evaluation procedure are described in Friedman et al. (2021). The difference between the strict (Map strict) and the relaxed (Map relaxed) score is that in the first one undecided pairs are counted as not matching whereas in the latter they are counted as matching. The best ranked approach in the competition was presented by Alshomary et al. (2021). They propose a siamese neural network architecture and use a contrastive loss function to learn embedding representation of arguments and key points where a matching pair is close to each other. They finetuned a Roberta model for 10 epochs and a batch size of 32. They report a Map strict of 0.84 and a Map relaxed of 0.96. Kapadnis et al. (2021) propose a different approach and ranked 5th on the competition. They train a binary- classifier where the arguments and key points are concatenated and jointly fed into a transformer. In contrast to Alshomary et al. (2021) there is only one sentence embedding received by the transformer model representing the argument and key point jointly. They further enriched the sentence embedding by additional feature like part-of-speech tags, dependency tags and TF-IDF. Furthermore, additional datasets like the STS dataset and the IBM Arg30k dataset were used to finetune different transformer models. They report a Map strict of 0.872 and a Map relaxed of 0.966. Transformer models by nature have a relatively complex architecture. Consequently, results are not easy to explain and interpret. There seems to be a research gap in explainability methods for siamese neural networks. The research stream is emerging and different approaches have been published recently. Utkin et al. (2019) suggest a new method for explaining siamese neural networks that is based on an autoencoder. The key idea is to compare the explained example with a prototype at the embedding layer and then reconstruct the features of the embedding layer with an autoencoder. Robinson (2020) presents an interpretable visualization algorithm for siamese neural networks. The algorithm identifies differentiating and similar features which are used to project the dataset into a lower dimensional space. Interpretability is implemented by applying parametric interpretability methods like SHAP. We haven't found prior work that specifically focuses on the explanation of siamese neural networks for textual data. Therefore our work can be seen as a first contribution to this research gap.

3 Model

Our primary goal in model development was to achieve a superior performance on the KPA shared task of matching arguments to key points. The core element in prior research and the models we propose are transformer models. Transformer models can be used to represent the semantic meaning of words or sentences in a high dimensional space such that sentences with a similar meaning are located close to each other. We focused on three main pillars of development: (1) data used to pretrain and finetune the model, (2) unsupervised pretraining to tailor the general pretrained sentence transformer models to the specific domain of the dataset at hand and (3) supervised finetuning to finally predict a similarity score between matching arguments and key point.

3.1 Data

Common transformer models like BERT have been pretrained on a large and general corpus of text. To fit them to the specific language domains present in the topics of the KPA shared task some finetuning processes are required. A first logical option to finetune the model is to use the argument and key points from the original dataset. However, literature also provides alternative datasets that have been created for similar tasks like semantic textual similarity and semantic representation (Cer et al., 2017). The STS dataset provides 8020 labeled pairs of english sentences and their corresponding similarity score (Kapadnis et al., 2021). The IBM 30k dataset provides argument topic pairs together with a similarity score (Hovy et al., 2013). In prior work the use of the STS and IMB30k dataset led to a significant performance increasement on the KPA shared task (Kapadnis et al., 2021). Since both datasets are labeled, they can be used for unsupervised pretraining and supervised finetuning. However, labeled training data is not a prerequisite for finetuning sentence transformer models. We have furthermore created our own text corpus containing 10k sentences from the specific topics of the KPA task. They have been automatically collected by crawling newspaper articles on the internet.

3.2 Unspuervised Pretraining

There are several ways of training a transformer model in the absence of labeled training data. The original pretraining of the BERT family was preformed by masked language modelling. The idea is to mask out different words in a sentence and train the model by predicting the probability of the missing word. Other state-of-the-art methods specialising on the unsupervised learning of sentence embeddings are simple contrastive learning of sentence embeddings (SimCSE) (Gao et al., 2021) and semantic retuning with contrastive tension (CT) (Carlsson et al., 2021). Wang et al. (2021) recently proposed another promising unsupervised domain adaption approach called transformer based denoising autoencoders (TSDAE). The idea is to create a damaged version of a sentence by modifying characters, words or adding noise. The damaged sentence is fed into an encoder which learns the encoding of this sentence. A decoder is trained to reconstruct the original sentence afterwards.

3.3 Supervised Finetuning

In order to solve the final argument key point matching a supervised model is trained to learn a function predicting the similarity between a specific argument key point pair. Siamese neural network models have been specifically designed for similarity learning tasks. The architecture is capable of learning similarity scores for pairs of inputs but also to differentiate between different pairs of inputs. Our experiments with other architectures like classification models could not reach the same performance levels. Subsequently we focused our work on Siamese Neural Networks.

3.4 Siamese Neural Network with contrastive learning for semantic textual similarity

The basic idea of siamese networks is to have two identical parts of the model where each part is dedicated to one of the inputs. In our case these parts are two transformer models (Roberta) that learn sentence embeddings for each of the sentences. In the case of BERT, the input sentences are projected into a 768-dimensional embedding space. To make sure that both transformer models process the input sentences in the same way, weights are shared between the transformer layers of the model. Consequently, one receives two datapoints in the learned embedding space where distance measures can be applied to calculate the similarity between both sentences. Similarity is calculated by using the cosine distance. A contrastive loss function guides the training process in a way such that matching pairs are close to each other whereas non-matching pairs a far from each other (Hadsell et al., 2006).

$$L = yd^2 + (1 - y)max(margin - d, 0)^2$$

The formula shows the contrastive loss function implemented in our models: y is the true label of an argument key point pair (1=matching, 0=not matching), margin is a hyperparameter for the minimum distance of a non-matching pair of inputs, and d is the distance measure between both sentence embeddings (cosine distance). In case of a matching pair the model minimizes the squared distance between them. In case of a non-matching pair the model minimizes the squared margin subtracted by the distance. Intuitively the transformer models receive the required feedback for learning the embedding representation of the arguments and key points in a way that matching pairs have a small cosine distance whereas non-matching pairs have a large one.

3.5 Siamese Neural Network with Part-of-Speech feature

The idea for this model is mainly a combination of different approaches presented by Kapadnis et al.

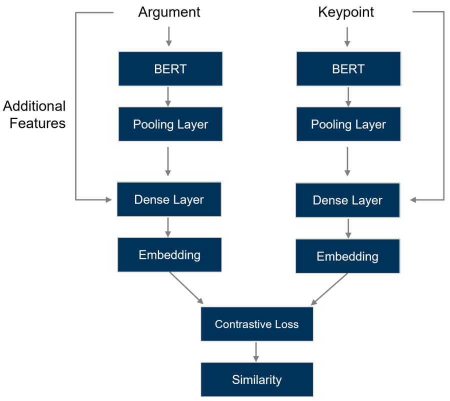


Figure 1: Architecture of SNN with additional Part-of-Speech feature

(2021) and Alshomary et al. (2021). The architecture of the model is again a siamese neural network. Additionally, the input to the model is enriched by manual engineered features like part-of-speech tags that are expected to capture some structural information of the arguments beyond the raw sentence embeddings provided by the transformer layers. Figure 1 visualizes the core idea of this approach. For each of the inputs the additional features are calculated separately. The sentences are then processed through a transformer layer and a pooling layer to receive sentence embeddings. On the next stage the additional features are added to the sentence embeddings. A fully connected layer allows the model to learn the importance of each feature by finding appropriate representations in the embedding space. The training procedure follows a similar logic as described above by following a contrastive learning approach. It is important to note that the transformer layers are finetuned in an independent step because it was not possible in the backpropagation step to differentiate between the additional feature added after the pooling layer and the features represented by the sentence embeddings without rebuilding the original BERT architecture. The final model uses part-of-speech tags as additional features.

3.6 Final Models and Results

Considering the number of different configurations with respect to datasets, unsupervised pretraining methods and supervised training methods we will focus on the presentation and discussion of the two models with highest performance. Both models are trained only on the original KPM dataset and both models built upon the Roberta architecture. The first configuration is called TSDAE SNN where the transformer model was initially pretrained with a transformer based denoising autoencoder and the similarity scores are learned with the simpler Siamese Neural Network guided by contrastive loss. No additional features are added. The second model is called SNN POS. No unsupervised pretraining methods has been implemented but instead the more complex siamese neural network is used together with part-of-speech tags. Table 1 shows the results.

Model	Map strict	Map relaxed
TSDAE SNN	0.921	0.967
SNN POS	0.912	0.970
SNN Benchmark	0.864	0.950
ENIGMA Benchmark	0.844	0.931

Table 1: Evaluation of model performance

With respect to the comparability of the result it is important to note that results of SNN Benchmark and ENIGMA Benchmark differ from the ones reported by Kapadnis et al. (2021) and Alshomary et al. (2021). Both groups published their code and the results shown in the table are the ones we received by reproducing their models. It is important to note that in the KPA shared task the final evaluation was performed by the publishing team and results were considerably lower in Friedman et al. (2021). Since our models were only evaluated by ourselves it is possible that reported results could change if they are subject to the official evaluation procedure. The results of Alshomary et al. (2021) with their siamese neural network model can be seen as a benchmark, as they performed best in the KPA shared task competition. Compared to the SNN Benchmark the SNN POS achieves a 4.8% performance improvement in Map strict and a 2% improvement in Map relaxed. The TSDAE SNN achieves a 5.7% improvement on Map strict and a 1.7% improvement on Map relaxed. Whereas TSDAE SNN scores higher on Map strict, SNN POS scores higher on Map relaxed. Both models show a significant performance improvement compared to all models that have been presented in the KPA shared task.

4 Discussion

We found strong evidence that unsupervised pretraining lead to a significant performance increment on the task at hand. The transformer based denoising autoencoder lead to better results compared to other pretraining methods like masked language modelling supporting the findings of Wang et al. (2021). Furthermore, it seems that Siamese Neural Networks are a superior architecture in semantic textual similarity settings compared to our experiments with classification models. With respect to the SNN POS model we found evidence that a significant part of the performance increment originates from hyperparameter tuning and is not only based on the architecture. Literature claims that transformer models like BERT are prone to overfitting issues (Sun et al., 2019). We have seen this problem throughout the hyperparameter tuning process and tackled it by finetuning the transformer models only for three epochs. It should be pointed out that, the topics in the test dataset are different from those in the training data. Consequently, extensive finetuning on topics in the training data can be a reason for worse generalisation on unseen topics. The SNN POS architecture seems to be promising, however it depends highly on the quality of the manually engineered features. It would be interesting to experiment with more complex features like TF-IDF as additional input to the model. In terms of performance the TSDAE SNN shows slightly better results and is the simpler model. Following Occam's Razor, we will evaluate this model with different explainability methods.

5 Explainability

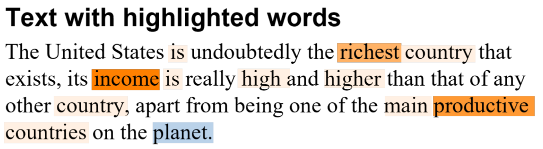
To understand how the developed model can be explained it is important to shortly recapture how the model works. The model predicts the similarity between arguments and key points. Hence the predicted value is a similarity score depending on the two input parameters: argument and key point. When employing state-of-the-art explainability techniques it becomes evident that these similarity scores can neither be clearly labelled as regression task nor as a classification task. The predicted score has no global meaning, as it would have in a typical regression task. The prediction of the market values of houses based on a set of input features can be interpreted more easily. By applying explainability techniques we can calculate the importance and contribution of certain features towards the predicted market value. In our case the predicted similarity score also depends on the two input parameters. But the score itself only receives its meaning in combination with the specific input-pair. It describes how the inputs are related to each other similarity-wise. So, it does not become clear in first place how a certain feature within the argument contributes to a high similarity score, because it depends on the specific features on the key point. Although we try to assign arguments to certain key points, the model cannot be labelled as a typical classification task, because we do not predict a certain class label. Instead, the class (key point) is part of the input parameters, which is usually not the case for classification tasks. However, those characteristics are typical for Siamese Neural Network and the application of explainability techniques for Siamese Neural Networks is a small and young research field (see section 2). Hence the question arises how to apply explainability techniques for Siamese Neural Networks. In the following we examine some explainability techniques and how they need to be adjusted to explain our SNN. Within the underlying case this leads to the question: Why specific argument key point pairs are similar or dissimilar? Some of the modifications can be transferred towards explainability of SNN in general and should be considered as one of the major contributions of this paper.

5.1 Leave one Out – Permutation Technique

Lei et al. (2017) introduced a methodology called Leave-one-covariate-out or LOCO to assess the importance of variables for predictive models. As proposed by (Lei et al., 2017) one variable is left out and then the model is refitted with the reduced feature space. The differences in the original model error and the new model error then leads to a score representing the importance of the left-out variable. In our case we have a model that can handle various sizes within the input variables. Hence, we can apply this idea without refitting the model again. Therefore, we iteratively leave out the words within the argument and analyse the difference in the resulting similarity score with respect to a specific key point. Thereby we gain an idea about the contribution of each word towards the similarity score.

5.2 LIME - Local interpretable model-agnostic explanations

LIME is a methodology to explain predictions locally, by training a simple, interpretable model. The basic idea is that complex models like deep neural networks can deliver great advantage and model complex dependencies. Those dependencies might not be representable by simpler interpretable models. However, for a specific prediction it is possible to train a simpler, interpretable model to explain a specific prediction locally (Ribeiro et al., 2016). Thereby the input feature vector is permutated, which creates a new dataset that is similar to the original input vector. Then an interpretable model is trained on the new dataset trying to predict the corresponding labels (Molnar, 2019, p. 168f). As pointed out previously Siamese Neural Networks cannot be clearly assigned to a regression or a classification task. Consequently, the question arises of which local interpretable model to use to approximate local predictions. In the following we show how to apply LIME to SNNs. Our approach builds upon the fact that LIME is a local explainability methods. In other words it aims at explaining why the model predicts a certain similarity score for a specific argument key point pair. The similarity score predicted by the SNN has a value range from 0 to 1. Since LIME is a local method, we keep one part of the input (key point) fixed. We can now interpret the local problem as a classification task with 0 representing dissimilarity and 1 representing similarity to the fixed key point. In a second step we create permutations of the argument only and receive a local classifier that explains the feature importance of each token in the argument for the similarity score to the respective key point. Note that the approach is symmetric which means that we can also fix the argument and explain the feature importance of the tokens in the key point. Figure 2 shows the result for an argument key point pair. The corresponding key point is: “The US has a good economy/high standard of living”. The orange terms contribute positively to the similarity score, whereas the blue terms contribute negatively to the similarity score. The intensity of colour increases with an increasing contribution of the term. The outcome of the classifier shows that the model considers words like income, richest and production important for a high similarity score to the respective key point, even though those words are not explicitly present in the key point. The proposed approach of reformulating SNN predictions as a local classification model by keeping one part of the input fixed is novel in literature and is expected to generalize on SNNs in different domains like face recognition. Hence, we strongly encourage future research on the proposed idea as it provides a simple approach of applying LIME to Siamese Neural Networks.



Text with highlighted words

The United States is undoubtedly the richest country that exists, its income is really high and higher than that of any other country, apart from being one of the main productive countries on the planet.

Figure 2: LIME applied to a specific argument. Highlighted words contribute the similarity score. Corresponding key point: "The US has a good economy/high standard of living"

5.3 Shapley Values

The basic idea of shapley values is to calculate the marginal contributions of each feature value towards the resulting score. Thereby game-theoretic techniques are employed, where the prediction is the “payout” and the feature values act as “players” contributing to it. For a more depth introduction to shapely values, the reader is referred to Molnar (2019, p. 177f). The Shapely values provided us with the ability to take a different perspective on the explanation part of our model. So far, we have analysed how important certain tokens of a specific argument are for the similarity to a specific key point. But what are the most important terms contributing to a high similarity score for a certain key point? In other words, we would like to know the importance of the tokens across all argument (within a certain topic) for the similarity to a specific key point. Therefore, we average the shapely values across all arguments for a specific key point. Figure 3 shows the terms that contribute most towards a high similarity score with the key point: “Routine child vaccinations, or their side effects, are dangerous”. Terms like diseases, virus, deadly are highly important. Thereby we get an understanding of what terms an argument would need to have in order receive a high similarity score with the investigated key point.

6 Conclusion

We have built several models by introducing additional features or shifting the language models more towards our specific domain by applying unsupervised pretraining methods like TSDAE. The best performance across Map strict and Map relaxed was achieved by the TSDAE SNN. With respect to the evaluation possibilities available our model achieved a significant performance increasement compared to the best ranked approach in the KPA shared task. We observed that BERT Transformers are likely to overfit, which must be as-

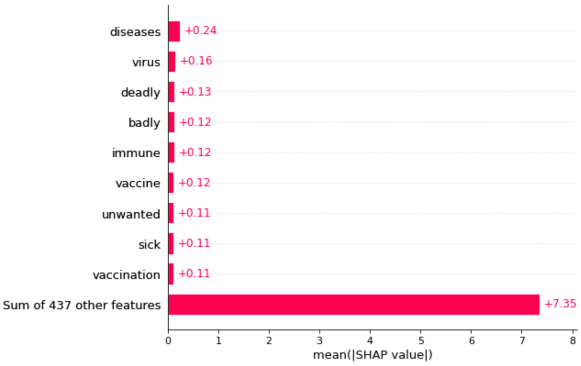


Figure 3: Most contributing terms to a high similarity score with the key point: "Routine child vaccination, or their side effects, are dangerous"

sessed carefully. With respect to explainability it was surprising that only little research exists dealing with Siamses Neural Networks. The state-of-the-art explainability techniques require some minor changes to work with SNN. Especially LIME is not directly applicable. Hence, we introduced a method that enables the training of local, interpretable models for Siamese Neural Networks by reformulating the predictions as a local classification model. We strongly encourage future research on the proposed idea since it is much simpler compared to prior research and is expected to generalize on other domains of SNNs.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Milad Alshomary, Timon Gurcke, Shahbaz Syed, Philipp Heinrich, Maximilian Spliethöver, Philipp Cimiano, Martin Potthast, and Henning Wachsmuth. 2021. Key point analysis via contrastive learning and extractive argument summarization.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. From arguments to key points: Towards automatic argument summarization.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Roni Friedman, Lena Dankin, Yufang Hou, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. Overview of the 2021 key point analysis shared task.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Manav Nitin Kapadnis, Sohan Patnaik, Siba Smarak Panigrahi, Varun Madhavan, and Abhilash Nandy. 2021. Team enigma at argmining-emnlp 2021: Leveraging pre-trained language models for key point matching.

John Lawrence and Chris Reed. 2020. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. 2017. Distribution-free predictive inference for regression.

Ran Levy, Ben Bogin, Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. Towards an argumentative content search engine using weak supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Christoph Molnar. 2019. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. Leanpub, Victoria, CA.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

Isaac Robinson. 2020. Interpretable visualizations with differentiating embedding networks.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Lev V. Utkin, Maxim S. Kovalev, and Ernest M. Kasimov. 2019. An explanation method for siamese neural networks.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. Tsdae: Using transformer-based sequential denoising auto-encoder for unsupervised sentence embedding learning.