Explainability in Irony Detection

Ege Berk Buyukbas(✉), Adnan Harun Dogan, Asli Umay Ozturk, and Pinar Karagoz

Computer Engineering Department, Middle East Technical University, Ankara, Turkey
{ege.buyukbas,adnan.dogan,ozturk.asli}@metu.edu.tr, karagoz@ceng.metu.edu.tr

Abstract. Irony detection is a text analysis problem aiming to detect ironic content. The methods in the literature are mostly for English text. In this paper, we focus on irony detection in Turkish and we analyze the explainability of neural models using Shapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME). The analysis is conducted on a set of annotated sample sentences.

Keywords: Irony detection · Explainability · Sentiment analysis · Neural models · SHAP · LIME

1 Introduction

Irony detection on textual data is one of the most difficult subproblems of sentiment analysis. Irony is defined as *the expression of one's meaning by using language that normally signifies the opposite, typically for humorous or emphatic effect*[1] by the Oxford Dictionary. It is particularly a difficult problem since opposition of the meaning is mostly implicit, and automated emotion detection methods lack the understanding of *common sense* that we humans share. Irony detection on textual data can be considered as a classification task to determine whether the given text is either *ironic* or *non-ironic*. In this work, we focus on irony detection problem from *explainability* point of view, and we particularly explore the performance of neural models on Turkish text. In addition to BERT and LSTM, we adapt Text-to-Text Transfer Transformer (T5 [6]) for binary classification of irony.

Explainability is a popular topic on machine learning, aiming to provide insight for the predictions generated by models. Models generated by some of the algorithms, such as Decision Tree (DT), tend to explicitly provide such insight due to the nature of the algorithm. However, neural network based models and transformers generate *black-box models*. In this work, for explainability analysis, we use Shapley Additive Explanations (SHAP [4]) and Local Interpretable Model-Agnostic Explanations (LIME [7]) methods, both of which are model-agnostic, hence can be applied on any predictive model.

[1] https://www.lexico.com/en/definition/irony.

M. Golfarelli et al. (Eds.): DaWaK 2021, LNCS 12925, pp. 152–157, 2021.
https://doi.org/10.1007/978-3-030-86534-4_14

Explainability of text classification is an emerging topic with a limited number of studies using LIME and SHAP. In [1], layer-wise relevance propagation (LRP) has been used to understand relevant words in a text document. In [5], theoretical analysis of LIME for text classification problem is examined. According to their analysis, LIME can find meaningful explanations on simple models such as linear models and DTs, but the complex models are not fully analyzed.

As an important difference from such previous studies, we focus on irony detection problem, which carries further difficulty as it involves a non-standard use of natural language. As another difficulty, we conduct the analysis on a morphologically complex language, Turkish. For the analysis, we use a new irony data set in Turkish, which includes 600 sentences with balanced number of labels.

2 Methods

In this study, similar to the approaches used for English and Turkish in the literature [2,8,9], Bidirectional LSTM (Bi-LSTM) neural model is used. Additionally, the masked language model BERT [3][2] is used with the BERT Base Multilingual Cased pre-trained model and fine-tuned for the classification task [2]. Different from the previous studies, Text-to-Text Transformer (T5) [6][3] is also employed. T5 model is trained with an open-source pre-training TensorFlow dataset of about 7 TB, Colossal Clean Crawled Corpus[4] (C4).

The first explainability method we use is LIME [7], which is a local surrogate model. The basic idea behind the algorithm is perturbing a data instance, then establishing new predictions with those perturbed data from the black box model. By this way, LIME can explain decisions for instances locally. In this work, we use LIME Text Explainer package[5] for constructing and analyzing BERT and T5 models. For LIME explainer, the number of words in each sentence is used as the number of features. The number of perturbed samples is set as 5000, which is the default parameter. The second explainability analysis method is SHAP [4], which assigns importance values to features of the data. In our study, we use SHAP Kernel Explainer[6] with its default parameters for the analysis of the Bi-LSTM model. With SHAP, we can examine not only contribution of words, but also features of a data instance, such as *existance of exclamation mark (!)* or a *booster*.

3 Experiments and Results

For the analysis, we use IronyTR Extended Turkish Social Media Dataset[7], which consists of 600 sentences. Data set is balanced with 300 ironic and 300

[2] https://github.com/google-research/bert.
[3] https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html.
[4] https://www.tensorflow.org/datasets/catalog/c4.
[5] https://github.com/marcotcr/lime.
[6] https://github.com/slundberg/shap.
[7] https://github.com/teghub/IronyTR.

Table 1. Comparison of methods on Turkish data set

Method	Accuracy	Precision	Recall	F1-score
LSTM	51.33%	55.09%	52.73%	53.88%
Bi-LSTM	50.16%	51.44%	62.07%	56.26%
BERT	**69.00%**	**71.34%**	**65.75%**	**68.43%**
T5	59.50%	59.21%	57.68%	56.85%

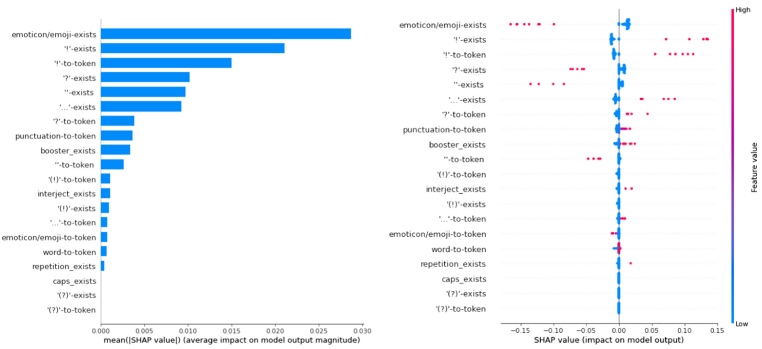

Fig. 1. Average and instance based feature impacts on Bi-LSTM Model (by SHAP method).

non-ironic instances. Sentences in the data set are collected from social media platforms either manually or by using the APIs of the platforms, and annotated by 7 Turkish native speakers to set the ground truth through majority voting.

For each model used for irony detection, we present accuracy, precision, recall, and F1-scores. All experiments are performed under 10-fold cross-validation. Table 1 gives the performance of LSTM, Bi-LSTM, BERT and T5, comparatively.

For explainability analysis, we applied SHAP method on Bi-LSTM model, LIME method on BERT and T5 models. The employed Bi-LSTM model uses feature-based inputs as well as text as sequence input. In order to analyze the effect of feature-based inputs, we preferred to apply SHAP for the analysis of this model.

We present a comparative explainability analysis of the models over two sample sentences, one with *irony* label, and the other with *non-irony* label. However, before this instance based comparison, we present the overall explainability analysis results by SHAP method on Bi-LSTM model.

General Analysis on Bi-LSTM by SHAP Method. In Fig. 1, the first graphics (on the left) show the average impacts of the features on the outcome as magnitudes, and the second graphics (on the right) shows the direct impacts. In the second graphics, the x-axis shows SHAP values of the features per instance. The orientations of the features' effects are displayed as well. In the figures, the most important feature on the model output is given as *emoticon/emoji-*

Table 2. Sentences, sentence numbers and translations.

Sent. #	Sentence
(a)	Hafta sonu meteor yağmurları gözlemlenebilecek.
transl.:	*Meteor showers can be observed at the weekend.*
(b)	Evde bayram temizliği yapacağız diye beni dışarı yolladılar, yolda fark ettim ki temizliğe benden başlamışlar çok üzgünüm şu an...
transl.:	*They sent me out to do holiday cleaning at home, I realized on the way that they started the cleaning with me, I'm very sad now...*
(c)	Yağmurlu havada su birikintilerinden hızlı geçmeye devam edin lütfen, ıslanmaya bayılıyoruz
transl.:	*Keep passing by through puddles fast in rainy weather please, we love to get wet.*

exists, which denotes if the sentence contains at least one emoji. According to the second graphics, if the sentence contains at least one emoji, the instance's prediction tends to be non-ironic, as the blue nodes denote that the sentence contains at least one emoji, whereas reddish nodes mean just the opposite. The second most important feature is *!- exists*, which denotes the sentence includes exclamation mark. According to the second graphics, if the sentence contains at least one exclamation mark, the instance's prediction tends to be ironic.

Analysis on Non-irony Sample. The non-irony sample we use is Sentence (a) in Table 2. According to Bi-LSTM model, all words in the sentence are effective for predicting the label as *non-irony*. The most effective words on the prediction are *yağmur (shower/rain)* and *meteor (meteor)* whose absolute SHAP values are maximum. Moreover, not only words but also the features *'!'- exists*, *'!'- to-token*, *'?'- to-token*, *'. . .'- to-token*, which are some of the most effective features given in Fig. 1, have similar effects on the model as their SHAP values are negative. The contribution of the words is higher than the features, as observed in the magnitude of the SHAP values. According to the results of the LIME method for BERT model, all the words in the sentence lead to *non-irony* label prediction as in Bi-LSTM. Moreover, *yağmur (shower/rain)* is the most effective word again, but the second most effective word is *gözlemlenebilecek (can be observed)* which is the least effective word in Bi-LSTM. The word *meteor (meteor)* is the third effective word. According to the result of the LIME method for T5 model analysis, *meteor (meteor)*, *yağmur (shower/rain)* and *gözlemlenebilecek (can be observed)* effect the label prediction as *non-irony*, as in BERT, but *hafta (at the week)* and *son (end)* affected the prediction oppositely. All three models predicted the sentence correctly as non-ironic.

Analysis on Irony Sample. For this analysis, we use Sentence (c) in Table 2. According to the results of SHAP method on Bi-LSTM, the following 7 words lead to prediction as *non-irony*: *yağmur (rain/shower)*, *hava (weather)*, *birikinti (puddle)*, *hizli (fast)*, *geçmek (pass by)*, *devam (keep on)* and *islanmak (get wet)*.

Table 3. SHAP scores for Bi-LSTM, LIME scores for BERT and T5 models on the sample sentences.

Bi-LSTM	Snt #	# of pos.	Avg of pos.	Max pos.	# of neg.	Avg of neg.	Max neg.	Pred.
	(a)	0	–	–	5	−0,050	−0,081	TN
	(b)	9	0,046	0,244	0	–	–	TP
	(c)	3	0,029	0,033	7	−0,066	−0,098	FN
BERT	Snt #	# of pos.	Avg of pos.	Max pos.	# of neg.	Avg of neg.	Max neg.	Pred.
	(a)	0	–	–	5	−0,017	−0,028	TN
	(b)	6	0,094	0,225	10	−0,025	−0,048	TP
	(c)	3	0,032	0,043	8	−0,022	−0,054	FN
T5	Snt #	# of pos.	Avg of pos.	Max pos.	# of neg.	Avg of neg.	Max neg.	Pred.
	(a)	2	0,325	0,374	3	−0,308	−0,624	TN
	(b)	9	0,033	0,075	7	−0,008	−0,018	TP
	(c)	3	0,155	0,324	8	−0,158	−0,515	FN

On the other hand, the words *su (water)*, *etmek (to do/to make)*, and *lütfen (please)* have positive SHAP values, and drive the model to irony. Additionally, *'!'- exists*, *'!'- to-token*, *'?'- to-token*, *'. . . '- to-token* and *booster-exists* features in the sentence lead the prediction as non-irony, as their SHAP values are negative, whereas *emoticon/emoji-exists* has a positive SHAP value. The LIME method for BERT model indicates that every word in the sentence except for *su (water)* and *bayilmak (love to)* have the same effect on the model as in Bi-LSTM model. The word *su (water)* has an opposite effect on the BERT model, and the word *bayilmak (love to)* has an impact on the model to lead the prediction as irony, but overall prediction of BERT model is non-irony. According to result of LIME method for T5 model, given in Table 3, the only words that lead to irony label prediction are *birikinti (puddle)*, *lütfen (please)* and *bayilmak (love to)*. However, the sentence is misclassified again as the other words contribute to non-irony for label prediction. Therefore, although the three models captured some words that lead to irony, they all mispredicted the sentence as non-irony.

Analysis on Sample Sentences. In order to give a more general view, we sampled 3 sentences as given in Table 2, where the first sentence is non-ironic, and the rest are ironic. The summary of SHAP results statistics and class label predictions as given in Table 3 for Bi-LSTM model. Similarly, summary of LIME weights and irony label predictions for BERT and T5 are presented in Table 3 and Table 3, respectively. For each prediction model, we observe slightly different effects of words. Considering the sentence (a), which is correctly labeled as *non-irony* by all three models, it is seen that in Bi-LSTM and BERT, all the words contribute to the correct label prediction, whereas in T5, although 2 of the words contribute to *irony* class label, 3 of the words determine the class label, especially one word with the maximum magnitude of −0,624. As another example, for the sentence (b), which is correctly labeled as *irony* by all models, in Bi-LSTM, again, all the words contribute to the correct label prediction. On the other hand, in BERT, the majority of the words lead to incorrect labels with small weights, whereas there is a word with the maximum weight of 0,225 to determine

the class label as irony. As for T5, the behavior is similar but not as strong as in BERT. It is seen that 7 out of 16 words have an effect towards non-irony label, and the effect of the other 9 words lead to the correct prediction with a maximum weight of 0,075. Overall, we can conclude that the effect towards the label prediction is distributed over a set of words in Bi-LSTM, yet we see few words providing a stronger effect for prediction in the transformer models.

4 Conclusion

In this paper, we investigate the explainability of LSTM, Bi-LSTM, BERT, and T5 based irony detection models on Turkish informal texts using SHAP and LIME methods. In terms of explainability, our analysis shows that, as expected, usage of punctuations such as *"!"*, *"(!)"* and *"..."* is a sign of irony in the detection models. The contribution of the words to the label prediction slightly differs for the models. In Bi-LSTM, many of the words in a sentence contribute to the prediction with comparatively smaller weights. On the other hand, for BERT and T5, fewer number of strong words determine the class label. As future work, using a multi-lingual model for T5 may be considered for irony detection performance and explainability analysis.

References

1. Arras, L., Horn, F., Montavon, G., Müller, K.R., Samek, W.: "What is relevant in a text document?": an interpretable machine learning approach. PLoS ONE **12**(8) (2017)
2. Cemek, Y., Cidecio, C., Ozturk, A.U., Cekinel, R.F., Karagoz, P.: Turkce resmi olmayan metinlerde ironi tespiti icin sinirsel yontemlerin incelenmesi (investigating the neural models for irony detection on turkish informal texts). IEEE Sinyal Isleme ve Iletisim Uygulamalari Kurultayi (SIU2020) (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 4765–4774. Curran Associates, Inc. (2017)
5. Mardaoui, D., Garreau, D.: An analysis of lime for text data. In: International Conference on Artificial Intelligence and Statistics, pp. 3493–3501. PMLR (2021)
6. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
7. Ribeiro, M.T., Singh, S., Guestrin, C.: "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, pp. 1135–1144 (2016)
8. Wu, C., Wu, F., Wu, S., Liu, J., Yuan, Z., Huang, Y.: THU-NGN at SemEval-2018 task 3: tweet irony detection with densely connected LSTM and multi-task learning. Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pp. 51–56 (2018)
9. Zhang, S., Zhang, X., Chan, J., Rosso, P.: Irony detection via sentiment-based transfer learning. Inf. Process. Manage. **56**(5), 1633–1644 (2019)