Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2839–2845
Marseille, 11–16 May 2020

Irony Detection in Persian Language: A Transfer Learning Approach Using Emoji Prediction

Preni Golazizian, Behnam Sabeti, Seyed Arad Ashrafi Asli, Zahra Majdabadi, Omid Momenzadeh and Reza Fahmi
Miras Technologies International
Number 92, Movahed Danesh St., Tehran, Iran
{preni, behnam, arad, zahra, omid, reza}@miras-tech.com

Abstract

Irony is a linguistic device used to intend an idea while articulating an opposing expression. Many text analytic algorithms used for emotion extraction or sentiment analysis, produce invalid results due to the use of irony. Persian speakers use this device more often due to the language's nature and some cultural reasons. This phenomenon also appears in social media platforms such as Twitter where users express their opinions using ironic or sarcastic posts. In the current research, which is the first attempt at irony detection in Persian language, emoji prediction is used to build a pretrained model. The model is finetuned utilizing a set of hand-labeled tweets with irony tags. A bidirectional LSTM (BiLSTM) network is employed as the basis of our model which is improved by attention mechanism. Additionally, a Persian corpus for irony detection containing 4339 manually-labeled tweets is introduced. Experiments show the proposed approach outperforms the adapted state-of-the-art method tested on Persian dataset with an accuracy of 83.1%, and offers a strong baseline for further research in Persian language.

Keywords: Irony Detection, Emoji Prediction, Transfer Learning, Multitask Learning, Deep Learning

1. Introduction

Irony is a creative phenomenon that has been widely studied in linguistics, philosophy, psychology and cognitive science (Colston and Gibbs, 2007), however, it's difficult to reach a consensus on a single general definition for it. Most theorists would agree that in irony the literal meaning of the words does not hold and the speaker says something that seems to be the opposite of what they mean (Colston and Gibbs, 2007). In the current research sarcasm is considered as a special case of irony (Farías et al., 2016; Gibbs, 2000).
In recent years, Twitter has turned into a source of information about users' expressions, ideas and opinions almost in any domain; therefore it has gained a lot of interest among researchers and companies who aim at user sentiment analysis, opinion mining, emotion recognition and other similar tasks.
Twitter includes a high percentage of tweets with irony usage among users. Persian speakers use this figurative term even more often due to the Persian language's nature and some cultural reasons. Presence of irony in a text can flip its sentiment's polarity, thus irony-aware predictions are crucial for more accurate performance.
The majority of the researches in automatic irony detection task has been addressed in English. The lack of such models for the Persian language has motivated us for the current research. This study aimed at irony detection in Persian language which was the first attempt to the best of our knowledge.
A common form of sarcasm consists of a positive sentiment contrasted with a negative situation (Riloff et al., 2013), therefore it was likely that learning the emotional information of a text would facilitate the task of irony/sarcasm prediction. On the other hand, it was seen that the majority of Persian Twitter users include either humor, irony or sarcasm in their posts. Taking this observation into account, a neural network model was presented which was first pretrained on a large unlabeled dataset containing Persian tweets with emoji occurrences to predict emojis. Then it was finetuned on a manually labeled dataset to detect irony. This approach, which was inspired by *DeepMoji*'s model (Felbo et al., 2017), utilizes tweets with emojis to pretrain the model and extract relevant representations. The pretrained model simultaneously provided a better initialization for the irony detection model and addressed the limitations in the labeled data samples.
Examples of our model predictions for both emoji and irony are presented in Table 1.
Contributions: The contributions of this study are the following:

- Proposing a strong baseline model for irony detection in Persian language.
- Introducing two datasets for Persian language; the first manually labeled irony detection dataset in Persian, and the largest dataset of Persian tweets with emoji labels.

2. Related Work

Due to the rapid growth of users in online social media platforms the attention of researchers and companies in the area of sentiment analysis, emotion recognition and irony detection has increased. Since then, several approaches to irony and sarcasm detection have been developed.
Some studies have used feature sets of the text to classify the text as ironic or not. *Reyes et. al.* described four sets of textual features (signatures, degrees of unexpectedness, stylistic features, and emotional scenarios) for recognizing verbal irony at a linguistic level (Reyes et al., 2013). *Riloff et. al.* presented a bootstrapping algorithm to learn the positive sentiment phrases and negative situation phrases in

Tweet	Predicted Emojis	Predicted Irony Class	Translation
مرسی برای این همه بدبختی ، خدایا شکرت	😂 🤦‍♀️ 😒 😑 😬	ironic	*Thanks for all the misery, thank God.*
من امروز خیلی خوشحالم دمت گرم روزمو ساختی.	😉 😋 👌 😁 😅	non-ironic	*I'm so happy today, bravo you made my day.*

Table 1: Example sentences and their corresponding top 5 most likely predicted emojis and irony labels.

sarcastic tweets, claiming that a common form of sarcasm happens in the result of contrast between positive sentiment and negative situation (Riloff et al., 2013). *Barbieri et. al.* aimed at detecting sarcasm by its inner structure (e.g. unexpectedness, the intensity of the terms or imbalance between registers) using seven sets of lexical features (Barbieri et al., 2014). *Ptáček et. al.* focused on supervised machine learning approach with various sets of n-grams and language-independent features and it was the first attempt at sarcasm detection in the Czech language (Ptáček et al., 2014). *Bouazizi and Ohtsuki* proposed a pattern-based approach that uses machine learning algorithms with four sets of features that cover different types of sarcasm (Bouazizi and Ohtsuki, 2016). *Fersini et. al.* introduced the Bayesian Model Averaging approach that takes into account several classifiers according to their reliabilities and their marginal probability predictions and also evaluated the impact of the most used expressive signals in the proposed approach and other baseline models (Fersini et al., 2015).

Some studies have considered context other than text features (e.g. authors' historical tweets, profile information, audience features, environmental features) to detect irony in the text. *Rajadesingan et. al.* described different forms of sarcasm and constructed relevant features for each of these forms to train a classification algorithm. In addition they took advantage of users' historical tweets and psychological and behavioral aspects of sarcasm to detect sarcasm in texts (Rajadesingan et al., 2015). *Bamman and Smith* included extra-linguistic features such as properties of the author, the audience and the immediate communicative environment in their model (Bamman and Smith, 2015). *Wallace et. al.* also considered contextual features like the forum's type (which the comment was posted to) and comments' overall sentiment (Wallace et al., 2015). The main limitation of these approaches is the need for additional information (authors' historical data, profile information, etc.) which is not always available.

By growing the popularity of deep learning techniques in NLP applications, the majority of the recent researches in current field have been based on these techniques. *Felbo et. al.* pre-trained a neural network model to predict emojis in the text and then transferred the model for different related tasks including sarcasm detection (Felbo et al., 2017). In contrast with approaches that use feature engineering to extract features , in (Amir et al., 2016) features are automatically extracted by learning user embeddings which requires users' preceding messages. A composite neural model consisting of a *CNN* followed by a *LSTM* and a *DNN* was proposed in (Ghosh and Veale, 2016) which was observed to outperform the text-based models. *Zhang et. al.* automatically captured both semantic and contextual features with the use of bi-directional gated recurrent neural network and pooling neural network and compared these features with discrete manual features (Zhang et al., 2016). *Hazarika et. al.* proposed a hybrid approach of both content and context-driven modeling for sarcasm detection which utilizes user embeddings that encode personality features, combined with content-based feature extractors such as *CNNs* (Hazarika et al., 2018). *Kumar et. al.* also proposed a hybrid neural network model containing soft attention-based *BiLSTM* and *CNN*. *GLoVe* word vectors were applied for representing word embeddings in this model. Punctuation-based features (e.g. number of question marks, exclamation marks and etc.) were also merged into the model (Kumar et al., 2019). Even though Deep learning models achieve major improvements in various tasks including irony detection, their main challenge is the large amount of data they require for a good performance, which is not available in many cases.

In the current study we aimed at developing an irony detection model for Persian language using transfer learning approach, while considering the challenges and limitations of the described approaches.

3. Datasets

3.1. Irony Definition

In Cambridge Dictionary irony is defined as *"The use of words that are the opposite of what you mean, as a way of being funny."* and *"A situation in which something which was intended to have a particular result has the opposite or a very different result."* . In Persian it has been translated as sarcasm, satire, mockery and ridicule. In fact there is a thin line between the meaning of these figurative terms especially in Persian language. In this research, we did not intend to distinguish the meaning of these terms, but to detect any occurrence in text where the literal meaning of the words did not hold.

3.2. Emoji Dataset Construction

To pretrain the model on emoji prediction task, a series of Persian tweets posted from May 2017 to September 2019 were collected using Twitter Firehose API. This resulted in a set of 94 million tweets in total from 848,275 different authors.

First, non-Persian tokens were eliminated and words were normalized (In Persian informal language, especially in Twitter, it is common to repeat a vowel of a word several times to intensify or emphasize the importance of the

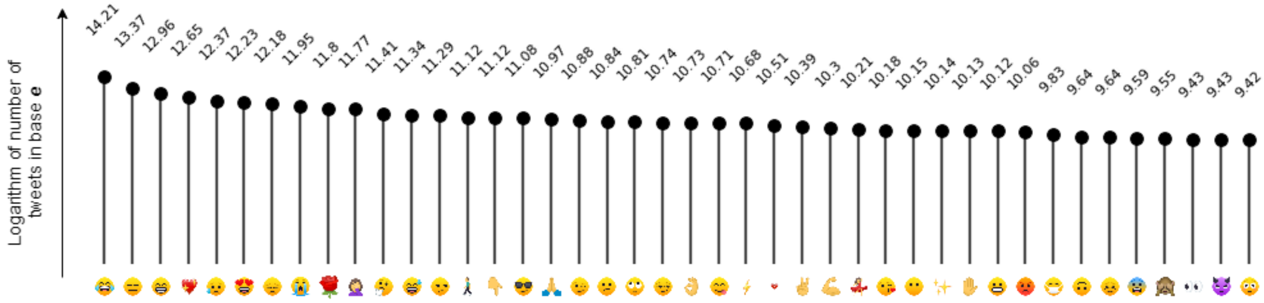


Figure 1: Distribution of tweets across final emoji labels.

word). Then all the tweets with emojis were selected. The ones containing URLs or mentions (tweet replies) were removed since their meaning may depend on a context other than the tweet itself (Felbo et al., 2017). Tweets with less than five non-emoji tokens were also removed because they were less likely to imply any remarkable meaning. Each tweet was then padded to fifty tokens for uniformity.

196 unique emojis were extracted from the remaining dataset. A categorization of emojis was done where emojis with similar emotional content were put together and the dataset was updated by converting each emoji to its category's main emoji (e.g., 😳, 😮 and 🤯 were replaced with 😮 in the dataset) .

To categorize the emojis, three annotators were asked to group them separately. The emojis which annotators had mutually agreed on their categories, were fixed and the ones that were put in different groups by each annotator, were voted by five other annotators. Each emoji was put in the group with the most votes by those annotators. This process led to 80 emoji categories. These categories were then used to update the dataset, where all emojis were replaced with their corresponding emoji label. Finally, all repetitive emojis were removed.

The model was pretrained with different setups for emoji labels and the best result was attained when emojis without any emotional content or with less than *lower-limit*(which was experimentally set) times occurrence in the dataset were eliminated. The distribution of tweets across final emoji labels used for the model are illustrated in Figure 1.

3.3. Irony Dataset Construction

Unlike English tweets where there are many posts containing #sarcasm or #irony hashtags, not enough such tweets for Persian were found, thus we had to annotate Persian tweets manually. A telegram bot was constructed for this purpose in which tweets from our provided dataset were randomly displayed to the annotators to tag as either *ironic*, *non-ironic* or *unknown*. A total of 12 annotators were involved in this process. The inter-annotator agreement was to tag the tweets based on the definition provided at 3.1. The *unknown* tag was also provided for the annotators, in case they were not confident about a tweet's label.

To collect data for annotation, first, we made use of our emoji corpus, selecting the tweets which had contrasting emojis (e.g., از شبکه خبرزنگ زدن به بابام میگن کانالو عوض کن ما یه چایی بخوریم. 😑 😂 Translation: *Someone called from the news channel to tell my dad 'change the channel so we can drink some tea'.*), because due to the observations in (Riloff et al., 2013) this kind of tweets would be more likely to have irony. After annotating 1000 tweets only 16% of them happened to be ironic. Therefore another set of tweets were collected from a telegram channel called *OfficialPersianTwitter* [1] that posts selected tweets from Twitter on a daily basis which are usually ironic, humorous or sarcastic. Once more, after annotating 1000 tweets, 37% of them were tagged as ironic so we continued with the second source. The same steps of tokenization and pre-processing were done for this dataset as well.

The dataset is published and available at the *MirasIrony* repository[2] in Github so that future researchers would make use of it.

3.4. Dataset Statistics

The size of the emoji dataset after all the preprocessing was reduced to 4,463,430 tweets. For the irony dataset, a total number of 4339 tweets were annotated, 1398 of which were tagged as *unknown* since the annotators were either not confident about or couldn't agree on their label. The details of these two datasets are illustrated in Tables 2 and 3 respectively.

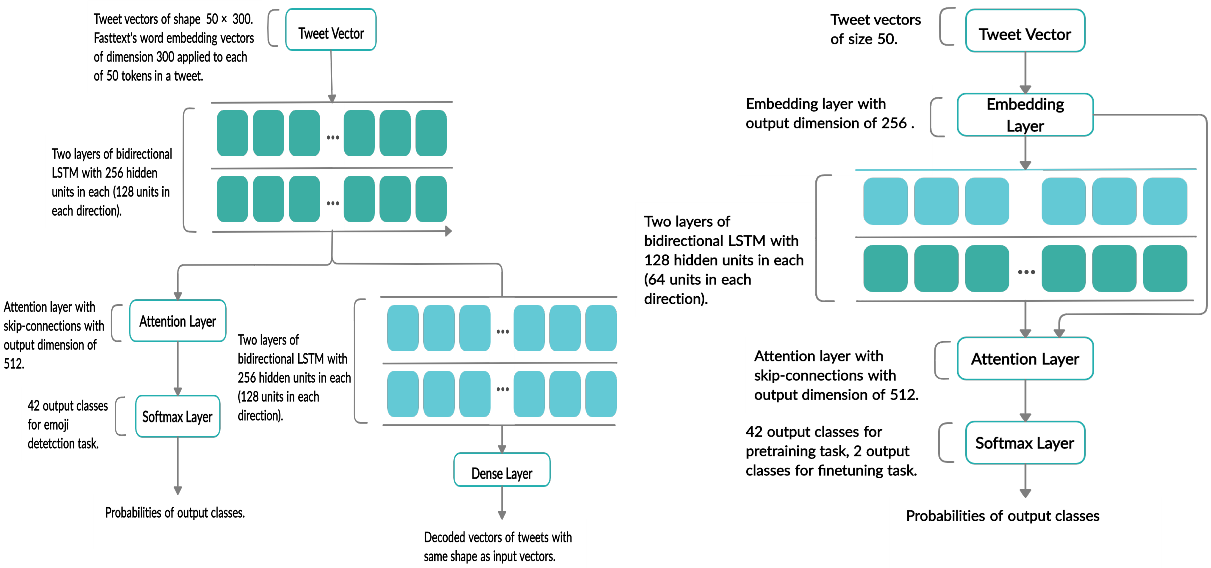
Property	Value
No. of tweets	4,463,430
Avg. no. of tokens per tweet	10.16
Max. no. of tokens per tweet	75
Avg. emoji occurrance per tweet	1.15

Table 2: Details of the emoji dataset.

Property	Ironic	Non-ironic
No. of tweets	1278	1663
Avg. no. of tokens per tweet	37.36	28.34
Max. no. of tokens per tweet	50	50

Table 3: Details of the irony dataset.

[1] https://t.me/OfficialPersianTwitter
[2] https://github.com/miras-tech/Mirasirony



(a) Multitask learning architecture

(b) Single-task learning architecture

Figure 2: Architecture of the proposed models.

4. Proposed Approach

4.1. Emoji Prediction

Two different models were proposed for the emoji prediction task. Figure 2 illustrates an overview of the models' architectures. The first model is an instance of multitask-learning which was jointly pretrained on two tasks of emoji prediction and text reconstruction using weight-sharing. The model includes an input layer that takes tweet vectors of fixed size (See *tweet-vector size* in Table 4) which were represented using *Fasttext*'s (Bojanowski et al., 2016) pre-trained word embedding vectors for Persian language. This is followed by two hidden layers of bidirectional LSTMs . To predict emojis, both of these layers were passed to a final softmax layer for classification. For auto-encoding task, output of the last BiLSTM layer was passed to another two hidden layers of bidirectional LSTMs sequentially to decode the vector representations and produce their initial embeddings.

The second model also included an input layer that takes tweet-vectors of fixed size and passes it to an embedding layer. A hyperbolic tangent activation function was applied to each word embedding vector. L2 regularization was used for embedding layer. The embedding layer was followed by two hidden layers of bidirectional LSTMs. Three previous layers were concatenated and passed to an Attention layer (Bahdanau et al., 2014). Attention mechanism allows the model to capture the words that are more important for predicting emojis in tweets. The output of this layer was passed to a softmax layer for emoji classification.

4.2. Irony Detection

The *single-task learning model* was chosen for transfer learning and irony detection, since as shown in section 5.2 it had a better performance on emoji prediction task during our experiments. Transfer learning is a common policy in many deep learning tasks that enables the model to transfer its knowledge from related tasks. The model was fine-tuned with two different approaches on irony detection task which are illustrated in Figure 3. The first approach freezes all layers at first and trains the softmax layer and then finetunes all the other layers together. The other approach trains the softmax layer at first and then starting from the first layer, finetunes each layer at a time while freezing all the other layers. At the end all layers are finetuned together once more (Felbo et al., 2017) (hereafter referred to as *'soft-tune'* and *'full-tune'* respectively).

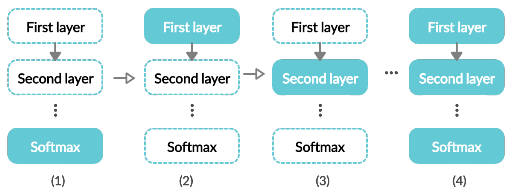


Figure 3: In each step layers filled with color are fine-tuned and other layers are frozen. Steps (1) to (4) illustrate the *full-tune* approach while step (1), directly followed by step (4), illustrates the *soft-tune* approach.

5. Experiments

5.1. Implementation Notes

The models were implemented [3] using *Keras* Framework, a high level interface for *Tensorflow* library (Abadi et al., 2015).

The emoji and irony datasets were balanced and 20% of each dataset was put aside for testing purposes and another 20% of the training set was held out for validation. The examples for train and test set were chosen randomly, because

[3] Some implemented classes of Deepmoji's model were used in our implementations. https://github.com/bfelbo/DeepMoji

it was likely that having the same keywords and being close in time, tweets would the same topics. This way the possible correlation of tweets was avoided.
Table 4 lists the hyper-parameter settings for the proposed models. *Adam* optimizer (Kingma and Ba, 2014) was employed with an initial learning rate of 0.001 for training. *Keras* default weight-initializers for each layer were preserved.

Hyperparameter	Value
Tweet-vector size	50
Embedding Dim.	32
LSTM size in the *Single-task learning model*	64
LSTM size in the *Multitask learning model*	128
No. of emoji classes for output	42
No. of epochs	20
Batch size	512

Table 4: Hyper-parameters' values

5.2. Results

The proposed models for emoji prediction were evaluated using accuracy, recall, precision and f-measure on the held-out test set.
As the results demonstrate in Table 5 the performance of the *Single-task learning model* was better than the *Multitask learning model*. The performance difference is likely to be due to the initial word embedding vectors in the *Multitask learning model*. *Fasttext*'s word embeddings were trained on a Wikipedia corpus which has a different context than Twitter, so it may have represented the words in a less relevant way.

Metric	Multitask learning model	Single-task learning model
Accuracy	40.84%	43.39%
Top 5 accuracy	51.47%	60.74%
Weighted precision	40.94%	43.94%
Weighted recall	40.84%	43.39%
Weighted f1-score	40.70%	43.30%

Table 5: Performance of the emoji prediction models with 42 output classes on test dataset.
'Multitask learning model' refers to the model pretrained on two tasks of auto-encoding and emoji prediction. *'Single-task learning model'* refers to the model pretrained only on emoji prediction task.

No previous researches have addressed the irony detection in the Persian language specifically, therefore we compared the proposed model with a state-of-the-art model proposed by *Ghosh and Veale* by training it on our Persian irony dataset (Ghosh and Veale, 2016). *Ghosh and Veale* described a neural network consisting of CNN, LSTM and DNN layers which gained an f-score of .921 in their experiments. We made use of their own implementation of the model [4]. The number of their network's parameters were too large for our dataset, so the embedding's dimension and the number of hidden units in LSTM were decreased to 32 and 64 respectively, also the dropout-rate was set to 0 (since they gained the best result without dropout); besides that, the other settings were preserved. Initially we trained the model with the settings mentioned above. Later we applied the embedding layer's weights of our pretrained model and trained again. Both versions were evaluated over our test dataset. It can be observed from Table 6 that our model comfortably outperforms both trained versions of *Ghosh and Veale*'s state-of-the-art model with an accuracy of 83.1%, which reflects the validity and advantage of our model for Persian language.
Our proposed model was also compared with multiple variants of itself to analyze the importance of the pretraining and the essence of each component present in its architecture.

The variants of our model employed for comparison are as follows:

- *Without pretraining*: The proposed *single-task learning model* on irony detection task without pretraining on emoji prediction (to analyze the importance of pretraining step).

- *Fasttext embedding*: The *Without pretraining* model with *Fasttext*'s word vectors as embeddings and having removed the embedding layer (to make sure that learning new embeddings related to context is essential for a good performance).

- *Without attention*: The *Without pretraining* model without the attention layer (to investigate the essence of attention layer in capturing the important words).

- *Single BiLSTM* : The *Without pretraining* model having removed one of the BiLSTM layers (to investigate the essence of BiLSTM layer in capturing the context).

As the results demonstrate in Table 6, removing each component from the model has led to a remarkable accuracy reduction.
Finetuning policy is also quite important since different policies can produce very different results.
Our model achieved its best performance using *full-tune* finetuning approach which indicates the validity of this policy for finetuning.
Eliminating the pretraining step led to 9.3% decrease in accuracy, so capturing the emotional context of the tweets were useful and transfer learning seemed to be a good policy.
Removing the BiLSTM layer from the *Without pretraining* model's architecture led to 14.9% decrease in accuracy, which indicates the important role of this layer in capturing the context of the tweets.
Removing each of the attention and embedding layers from the *Without pretraining* model also led to 9.1% and 2% decrease in accuracy respectively, which confirms the effect of capturing the important words and representing the words in a context-relevant way in model's performance.

[4] https://github.com/AniSkywalker/SarcasmDetection

Description	Model	Accuracy
Ghosh and Veale's model (Ghosh and Veale, 2016)	CNN-LSTM-DNN (no initial embeddings)	63.1%
	CNN-LSTM-DNN (with pretrained embeddings)	64.4%
Variants of the proposed model	Single BiLSTM	58.9%
	Without attention	64.7%
	Fasttext embedding	71.8%
	Without pretraining	73.8%
Proposed model with two different approaches for transfer learning	*Soft-tune* approach	77.0%
	Full-tune approach	**83.1%**

Table 6: Comparison of the proposed model with its variants and a state-of-the-art model.

Overall all of the components in our model seem to be essential for attaining the best performance.
The Receiver operating characteristic (ROC) curves of all the described models above are plotted in Figure 4 for a better visualization of results.

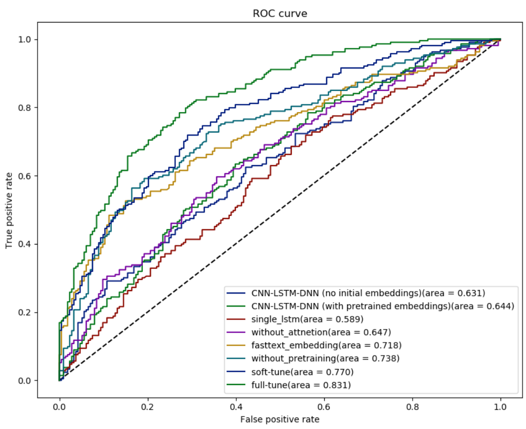


Figure 4: ROC curves of the proposed model and other models compared with it.

6. Conclusion and Future Work

This research presented a transfer learning approach for irony detection in Persian language. Initially two deep learning models were proposed and pretrained on emoji prediction task. The model with the best performance (43.39% accuracy when trained on 42 output classes) was finetuned using two different approaches for irony detection. For emoji prediction task, a large dataset containing 4,463,430 tweets with emoji occurrences was constructed and preprocessed. As a first attempt for irony detection in Persian language, a large manually annotated dataset containing 4339 Persian tweets with one of three *'ironic'*,*'non-ironic* or *'unknown'* labels was also constructed. Our model achieved an accuracy of 83.1% when evaluated on the test set.
Our plan for future work is to take different forms of irony into account and extend the model so it can predict the type of the irony as well. We plan to merge this model with a sentiment analysis model and investigate the result of irony-aware sentiment analysis. We also plan to test our model on other languages to investigate its language dependency properties.

7. Acknowledgements

The authors would like to thank everyone at *Miras Technologies International* for their technical support especially those who put a lot of effort in building the presented corpus.

8. Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Amir, S., Wallace, B. C., Lyu, H., and Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Barbieri, F., Saggion, H., and Ronzano, F. (2014). Modelling sarcasm in twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bouazizi, M. and Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488.

Colston, H. and Gibbs, R. (2007). A brief history of irony. *Irony in language and thought: A cognitive science reader*, pages 3–21.

Farías, D. I. H., Patti, V., and Rosso, P. (2016). Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)*, 16(3):19.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Fersini, E., Pozzi, F. A., and Messina, E. (2015). Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–8. IEEE.

Ghosh, A. and Veale, T. (2016). Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169.

Gibbs, R. W. (2000). Irony in talk among friends. *Metaphor and symbol*, 15(1-2):5–27.

Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., and Mihalcea, R. (2018). Cascade: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., Abdel-Basset, M., et al. (2019). Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access*, 7:23319–23328.

Ptáček, T., Habernal, I., and Hong, J. (2014). Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223.

Rajadesingan, A., Zafarani, R., and Liu, H. (2015). Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106. ACM.

Reyes, A., Rosso, P., and Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language resources and evaluation*, 47(1):239–268.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714.

Wallace, B. C., Charniak, E., et al. (2015). Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP 2015*, pages 1035–1044. Association for Computational Linguistics (ACL).

Zhang, M., Zhang, Y., and Fu, G. (2016). Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460.