

# Detection of cyberbullying in Arabic social media using dynamic graph neural network<sup>\*</sup>

Ahmed Bouliche<sup>1,\*</sup> and Abdellah Rezoug<sup>2,†</sup>

<sup>1</sup>Department of computer science, Faculty of sciences, University of Boumerdes, Boumerdes, Algeria

## Abstract

Despite all the advantages social networks have brought to the world, they are also a very favourable environment for the growth of so-called electronic crimes. Textual exchanges between users may include clues to crimes committed or being prepared. Usually, methods of Natural Language Processing (NLP) and neural networks are effective ways to detect cybercrimes particularly cyberbullying. In this paper, we proposed techniques that allow to use structures of dynamic temporal graphs as direct inputs to a model without turning them into static graphs as well as a message passing algorithm that fits well with the approach. The effectiveness of these techniques was tested on a prototype model. Fortunately, the proposed techniques have been proven to work, but with poor model performance. The applicability of a crime detector can be established with a session classifier if the data is more general, i.e., represents all the language used by bullies.

## Keywords

natural language processing, Arabic language, dynamic graph neural network, graph neural network, cybercrime detection, cyberbullying


## 1. Introduction


The widespread use of social media in recent years has helped to spread many types of dangerous cybercrimes. Fraud, swearing, harassment, terrorism, bullying, and drug dealing are examples of many cybercrimes that people have been suffering from in recent years. Cyberbullying is common among adolescents and has negative effects on the psychological and even physical state of the victim. Therefore, focus has recently turned to developing solutions capable of automatically detecting and predicting cybercrimes based mainly on artificial intelligence and natural language processing (NLP).

It is obvious that the domain of NLP is developing very well for some languages, such as English. Yet, for the Arabic language, there are only some experimental attempts that are issued here and there, which remain insufficient. In addition, the Arabic language poses another kind of difficulty related to the way it is written. Also, there is no unified language among the Arab

---

TACC 2022: Tunisian-Algerian Joint Conference on Applied Computing, 13 - 14 Dec Constantine, Algeria

 [bouliche.ahmed.2@gmail.com](mailto:bouliche.ahmed.2@gmail.com) (A. Bouliche); [a.rezoug@univ-boumerdes.dz](mailto:a.rezoug@univ-boumerdes.dz) (A. Rezoug)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

countries, as each country uses its own dialect.

In recent years, researchers have proposed very effective machine learning-based solutions to detect cyberbullying by processing published texts, images, and videos. In this study, we are only interested in methods based on NLP. Arabic cyberbullying detection using sentiment analysis was proposed by Almutiry and Abdel Fattah [1]. Abusive language detection on Arabic tweets based on matching texts with a list of obscene words was presented by Mubarak et al. [2]. Aldjanabi et al. [3] developed a classification system for determining offensive and hate speech using a multi-task learning (MTL) model built on top of a pre-trained Arabic language model. Fatemah Husain [4] applied a single learner machine learning approach and an ensemble machine learning approach for offensive language detection in the Arabic language.

This article presents an approach to solving the problem of detecting cyberbullying through text analysis using a dynamic graph neural network (DGNN). The proposed method is a graph-level classification to classify an entire comment session, whether it contains cyberbullying or not. These sessions are a set of comments where each relationship between two comments is a response relationship between the two. Each node in the graph is a comment, and an edge between them is a reply link. The temporal aspect of the data must be taken into consideration because texts are time-series data. The size of comments varies and is not fixed, so the type of time graph that will be used is dynamic time graphs. According to our modest research, the majority of methods on time graphs assume that graphs are static and often treat them as static [28]. We propose a method that takes into account the dynamism of time graphs with a shift method for processing temporal and spatial aspects.

The remainder of this article is organised as follows. Section 2 summarises the state of the art of the proposed solutions to detect cyberbullying using artificial intelligence methods. Section 3 explains in detail the approach proposed in this article. Section 4 summarises the conducted experiments and the obtained results. This article is finished with conclusions on the most relevant observations and results.

## 2. Related works

Several researchers have tried to handle cyberbullying detection using machine learning algorithms. To detect aggressive behaviours, the work by [5] employs three deep learning models: Closed Bi-directional Recurrent Unit (BiGRU), Transformers, and Convolutional Neural Networks (CNN). Experimental studies were conducted to examine how effective the models are at classifying well-known hate speech Twitter data-sets into aggressive and non-aggressive. An accuracy of about 88% was reached.

The authors proposed by [6] another method based on supervised machine learning approaches. Several classifiers have been used to train and detect acts of bullying. The data set shows that neural networks perform better and achieve an accuracy of 92.8%, whereas simple vector machines (SVM) achieve 90.3%. Reynolds et al. [7] used textual features and constructed several metrics to assess swear words through a swear word dictionary.

What makes cyberbullying detection particularly challenging compared to offline bullying is that language can pose difficulties like ambiguity and sarcasm. Role prediction can also be a difficult task if we look at a conversation thread between the individuals involved in the conversation (victim, bully, bully type, advocate, etc.). The predictions made by the previous automatic methods for the detection of cyberbullying do not reflect the complexity of the task described above. Recent efforts include combined word normalisation, named entity recognition to detect person-specific references [8]. In their work, [9] collected a large data set from *ask.fm* and used BOW (Bag Of Words) features as a first test, then extended it with term lists, subjectivity lexicons, and subject model features.

Recently, neural word embedding and neural network techniques have been applied. Convolutional neural networks (CNN) on phonetic features have been applied by [10] and [11] to study, among others, the same architecture on textual features in combination with long-short-term memory (LSTM) networks.

People tend to describe their experience of bullying in messages called bullying traces. The researchers by [12] presented a method to examine bullying traces. They identify several key issues in using social media to study bullying: text categorisation, role labelling, and topic modelling.

Using graphs to visualise potential cyberbullies and their connections is a recent method. The researchers by [13] concretely implemented techniques from graph theory: *Network functionalities* (degree of centrality: number of incident links from a node; close centrality: average distance the shorter from each vertex to the other vertex.), *Content-based features* (length, sentiment, offensive words, pronouns).

The focus of most existing work done on cyberbullying is the independent content analysis of comments within social media sessions. Suyu et al. [14] cited three different key limitations for detection of cyberbullying, 1) only consider the content within a single comment rather than the topic coherence across comments 2) remain generic and exploit limited interactions between social media users, 3) overlook the temporal correlations among different comments. They showed that interaction within the same session may evolve. Modelling topic coherence between comments and temporal interactions is critical to capturing repetitive bullying characteristics, leading to better predictive performance. To achieve this goal they constructed a unified temporal graph for each social media session and then proposed a principled graph-based approach for modelling the temporal dynamics and topic coherence throughout user interactions.

Research in psychology and social science has shown that cyberbullying is carried out repeatedly against victims [15]. Studies of user interactions allow us to characterise their repetition by both content and temporal analysis. However, modelling interaction in social media sessions presents some challenges: sparsity, repetition, and user characteristics.

### 3. The proposed approach

Our approach is based on dynamic graph neural networks. This part explains every step of the procedure. It is composed of: data collection, graph construction and how constraints are handled, coordination matrix creation, tokenisation and training the model.

#### 3.1. Graphs generation

Data generation was the hardest task and the process that took us the most time. Understanding the data structure is also an important factor for data generation. In our case, data generation is a process based mainly on dynamic temporal graph creation.

##### 3.1.1. Data collect

With some research and exploitation, we were able to collect 11268 comments, of which 8417 are negative instances that do not contain cyberbullying behaviour and 2851 positive instances that do contain cyberbullying behaviour. The links to the databases used were shared by [16].

##### 3.1.2. Construction of graphs

The construction of temporal graphs consists of understanding our data structure as well as the two types of temporal graphs.

1. **Structure of a session** The sessions will be represented by temporal graphs where each node represents a comment and the edges between them a response relationship. At the beginning (step  $t = 0$ ), the graph has nodes representing tokens corresponding to the  $i^{th}$  comment. Each session is a dynamic temporal graph, dynamic temporal graphs are records over time. The records are graphs at each time step, with the  $i^{th}$  node representing the token of the  $i^{th}$  comment at time step  $t$ . Except the size of comments is not fixed, so some nodes and edges will be removed over time.

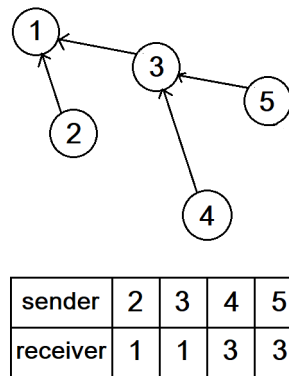
There are two constraints that describe the structure we want to use for the sessions we are going to process:

- **edge loop.** Each node in our graph is a comment and not a user. If a comment  $j$  is a reply to a comment  $i$ , and we add a cycle edge (i.e) comment  $i$  is also a reply to  $j$ , this just doesn't make sense.
- **session owner.** The session owner or the owner of a post is often the person bombarded with comments and insults.

2. **Constraint of dynamic temporal graphs creation**

- **Node Deletion.** nodes must be deleted only if the comment that corresponds to the node is less than the current time step, if the comment contains 10 tokens then the node will be deleted from the graph in the  $11^{th}$  time step.

- **Edge removal.** Edges should be removed only if one of the nodes linked to the edge is removed.  
Removing and adding nodes is important for processing the temporal aspect of the graph. The deletion of the edges is done if a node linked to it is deleted, because no spatial data is propagated in the network.
- **Number of nodes equal to the number of nodes in the COO matrix.** The COO matrix (CO-Ordination matrix) is a an optimised representation of the edges in the graph that consumes less space and reduces the number of units of calculation. The matrix contains two rows. In each column the first row contains the sender nodes and the second row the receiver nodes. The union of the two lines gives us all the nodes of the matrix.



**Figure 1:** Example of coordination matrix

If the set of nodes in the COO matrix is greater than the set of nodes in the node embedding matrix, it means that the embedding matrix is missing information about some nodes. This will make message passing impossible.

- **Isolated node.** Isolated nodes cause problems because they do not belong to the set of nodes in the COO matrix. The absence of a node in the COO matrix puts us in a case where we do not know if the node has been deleted or is isolated. Deleting a node causes an edge to be deleted and deleting an edge causes a node to be isolated.
- **Shifted node embedding.** A node is added to the graph at time  $t$  if and only if the length of the comment which corresponds to the node is greater than  $t$ . If between  $[0 - t - 1]$ , some nodes have been deleted. Deleting these nodes will cause a shift in the order of the nodes in the node embedding matrix. This means that the embedding of node 14 can actually be the embedding of node 18.

### 3. Graph creation

- **tokenisation:** The Transformers library [17] offers an entire hub where multiple developers can put their algorithm to use. For the segmentation of Arabic words, we use the segmenter [18] and for the tokenisation algorithm we use a WordPiece

algorithm developed by [19] for the Arabic language. WordPiece was originally designed by Google when designing the famous BERT model.

The tokenisation is performed on the entire data-set. When creating temporal graphs, for each node ( $i$  for example) that we add to a graph in a time  $t$ , we add the identifier of the  $w_i$  token of the comment  $i$  in the embedding matrix. These identifiers will then be passed to an embedding layer which as output gives us an embedding vector (the dimension of the vector is a hyper-parameter).

- **Coordination matrix (COO-matrix).** We start by picking a random number to determine the length of the session (number of comments or node in the session). In the first time step  $t = 0$  we build the first graph where each node  $i$  will pick a random number  $e \in [1, size_{session}]$ ,  $e$  is the node that will be connected to node  $i$ . This will build the COO matrix which represents the edges and structure of the graph. We are still in the first step in time. To solve the isolated node problem, we add a self loop connection to the isolated node in the CO-Ordination matrix. Both the sender and the receiver are the isolated node and the weight of the edge is null. To solve all the rest of the constraints for the creation of the dynamic temporal graph mentioned above, we proposed shifting the indices of nodes. For each step in time, we save a COO matrix with the new structure (the nodes not deleted) and another COO matrix where the indexes of the nodes are shifted backwards. In the case of spatial data processing, we can use the nodes shifted edge indices, because the indices will be reset and this will prevent us from falling in the case where a node index is greater than the number of nodes in the embedding matrix. In the case of time series data processing, we use the original indices to check if the node is not deleted and the shifted edge indices matrix to select the embedding vector of nodes for processing their temporal information. Finally, we were able to generate 753 dynamic temporal graphs from which 191 contain cyberbullying behaviour and 562 do not.

## 3.2. Model training

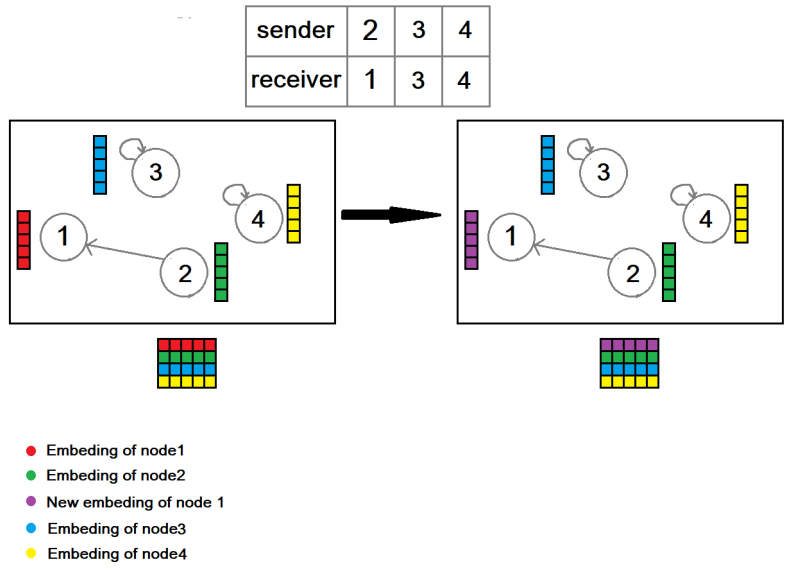
### 3.2.1. Spatial data processing

Generally, message passing is done through a dot product between the adjacency matrix and the node embedding. We have a CO-Ordination matrix which consists of two rows and in each column the two connected nodes. How can the COO matrix be used for message passing?

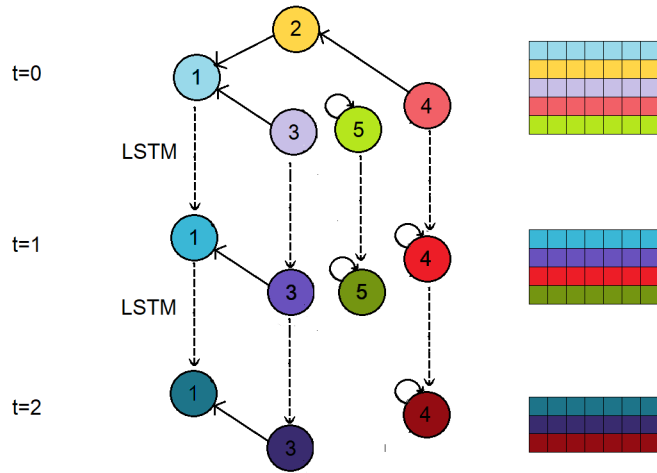
The embedding matrix is created using an embedding layer that takes the token identifiers of the nodes at time step  $t$  as input. The embedding dimension we chose is  $dim = 128$ . We create a copy of the embedding matrix and we update the vectors of the nodes which receive information from their neighbours in the embedding matrix. This can be achieved as follows:

For each column in the shifted edge indices CO-Ordination matrix, we update the embedding vector of the receiver node in the new embedding matrix. Example the first column is:  $shifted_{edge\_indices}_{.,1} = [1, 3]$ , in the new node embedding we will use the vectors in the position

1 and 3 to update the vector in the position 3. And if the vector 3 has already been updated, we use the vectors in position 1 and the new vector in position 3.



**Figure 2:** The message passing algorithm



**Figure 3:** Analysis of temporal aspect in graphs

This procedure must be done for all the columns in the shifted edge indices COO matrix. The result are weighted by parameterised readable weight matrix to learn the interactions (the number of layers we have chosen is three.). After applying these transformations to the graph

in the first time step, we will do the same for all the graphs in the future time steps individually. This way, all spatial information about neighbouring nodes having been propagated through the graph in all time steps, the resulting node embedding can be used as input data for processing the temporal aspect.

### 3.2.2. Handling temporal aspect of graphs

After applying message passing for each node in each graph over time, our node embedding all contain information about their neighbouring nodes. It was previously mentioned that deletion leads to an embedding lag which can cause problems for handling the dynamic graph temporal aspect. If the nodes will be shifted then the time series data processing will learn corrupted and incorrect sequences. To solve this problem, we shifted the indices of the CO-Ordination matrix so that Deep Learning architectures used for processing time-series data learn the correct sequences. This is achieved as follows:

We start by concatenating all the records of a node through time. This will prepare the input data for the chosen Deep Learning architecture and will make processing temporal data very easy. To concatenate the vectors of the node through time, we have to loop over the time steps and check if the node has not been deleted from the COO matrix (this will indicate that the number of comment tokens is greater than  $t$ ) and we have the shifted index of the node so we can extract the node embedding vector from the embedding matrix at time  $t$ , concatenate it with the node embedding vector at time  $t - 1$ . We will have a representation of all the records over time in a matrix that will be used as input data for an LSTM layer. This concatenation operation will be done for all the nodes of the session.

An LSTM architecture will be used to process the temporal aspect of the data, the result of each node that passes through the LSTM will also be concatenated. Then we calculate the transpose the resulting matrix and get the average on the last dimension for shape reduction. A finale output layer is added to predict if the session contains cyberbullying or not.

## 4. Experimental results

In order to verify the effectiveness of the proposed approach, we conducted experimentation using the dataset. The input data contains 11268 Arabic comments (tweets), where 8417 do not contain cyberbullying and 2851 do contain cyberbullying. From these comments, 753 dynamic temporal graphs were generated, of which 191 contained cyberbullying and 562 did not.

The training results of the first epoch were 49% but we saw improvements in the second epoch with an accuracy of 74% which means that the learning took place and the proposed techniques applied to the CO-Ordination matrix worked. We were not able to continue the training for more than 2 epochs due to OOM (Out Of Memory) reasons. The temporal graphs are very memory-consuming data structures. We believed that the imbalance and the size of the database reduced the performance. The model that we have chosen is a prototype model to test the efficiency and profitability of the shifted edge indices COO matrix that we proposed.



We could have used regularisation technique for generalisation or scheduling the learning rate for better optimisation, but performance of the model was not our goal.

We tested on the same set that we used for training and we noticed that there was an extreme bias in the model predictions. This is not only because of the imbalance in the dataset, where almost 75% of the examples are negative examples, but also because of the small size of the dataset.

**Table 1**  
confusion matrix

	Negative	positive
Negative	562	191
Positive	0	0

From this confusion matrix, we calculated the following micro metrics (positive predictive rate, recall, specificity, negative predictive rate, precision) and the f1 score as a macro metric. The results are summarised in Table 2.

**Table 2**  
performance indicator metrics

PPR	Recall	NPR	Specificity	acc	f1 score
0	0	100%	74%	74%	85%

Knowing that comments are text in nature and responses between one comment and another can be structured with a temporal graph gives us the ability to harness the power of dynamic temporal graphs. With different message-passing algorithms available, GNNs capture the relationships between nodes and learn to properly represent a graph in node embedding that represents each node and the information acquired from their neighbours. And with different deep learning architectures that process time-series data, we can learn how these node interactions change over time. We leverage the dynamism of the dynamic temporal graphs with a bunch of techniques that prove the learning has taken place despite the complex structure of the dynamic temporal graph being kept.

## 5. Conclusion

Cyberbullying is a dangerous cybercrime that is lightly taken by governments despite the psychological damage it causes to its victims. It has spread very rapidly due to the spread of social media, which has caused an increase in the number of bullies that has led to a permanent cycle of violence.

Session classification seems more applicable for cyberbullying detection. Graph neural networks (GNN) make it possible to capture the relationships between entities, and by adding an architecture that deals with the temporal aspect of graphs, we have the ability to model relationships between entities over time. We decided to leverage dynamic temporal graphs

because there is little research on this data structure, and hence the majority of research considers these graphs to be static temporal graphs. When turning a dynamic temporal graph into a static temporal graph, we think that the algorithm will waste a lot of training steps learning insignificant alternatives that we added to make the structure static. Therefore, we have proposed a technique that uses the dynamism of this structure. The method proved that learning took place. Unfortunately, we did not have the necessary means to properly test the performance of the method.

## References

- [1] S. Almutiry, M. Abdel Fattah, Arabic cyberbullying detection using arabic sentiment analysis, *The Egyptian Journal of Language Engineering* 8 (2021) 39–50.
- [2] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on arabic social media, in: *Proceedings of the first workshop on abusive language online*, 2017, pp. 52–56.
- [3] W. Aldjanabi, A. Dahou, M. A. Al-qaness, M. A. Elaziz, A. M. Helmi, R. Damaševičius, Arabic offensive and hate speech detection using a cross-corpora multi-task learning model, in: *Informatics*, volume 8, MDPI, 2021, p. 69.
- [4] F. Husain, Arabic offensive language detection using machine learning and ensemble machine learning approaches, *arXiv preprint arXiv:2005.08946* (2020).
- [5] M. Alotaibi, B. Alotaibi, A. Razaque, A multichannel deep learning framework for cyberbullying detection on social media, *Electronics* 10 (2021). URL: <https://www.mdpi.com/2079-9292/10/21/2664>. doi:10.3390/electronics10212664.
- [6] J. Mounir, M. Nashaat, M. Ahmed, Z. Emad, E. Amer, A. Mohammed, Social media cyberbullying detection using machine learning, *International Journal of Advanced Computer Science and Applications* 10 (2019) 703–707. doi:10.14569/IJACSA.2019.0100587.
- [7] K. Reynolds, A. Kontostathis, L. Edwards, Using machine learning to detect cyberbullying, in: *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 2, 2011, pp. 241–244. doi:10.1109/ICMLA.2011.152.
- [8] U. Bretschneider, T. Wöhner, R. Peters, Detecting online harassment in social networks, in: *Thirty Fifth International Conference on Information Systems, Auckland 2014*, 2014, pp. 1–14.
- [9] C. Emmery, B. Verhoeven, G. D. Pauw, G. Jacobs, C. V. Hee, E. Lefever, B. Desmet, V. Hoste, W. Daelemans, Current limitations in cyberbullying detection: on evaluation criteria, reproducibility, and data scarcity, *CoRR abs/1910.11922* (2019). URL: <http://arxiv.org/abs/1910.11922>. arXiv:1910.11922.
- [10] X. Zhang, J. Tong, N. Vishwamitra, E. Whittaker, J. P. Mazer, R. Kowalski, H. Hu, F. Luo, J. Macbeth, E. Dillon, Cyberbullying detection with a pronunciation based convolutional neural network, in: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 740–745. doi:10.1109/ICMLA.2016.0132.
- [11] H. Rosa, D. Matos, R. Ribeiro, L. Coheur, J. P. Carvalho, A “deeper” look at detecting cyberbullying in social networks, in: *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489211.
- [12] J.-M. Xu, K.-S. Jun, X. Zhu, A. Bellmore, Learning from bullying traces in social

- media, in: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Montréal, Canada, 2012, pp. 656–666. URL: <https://aclanthology.org/N12-1084>.
- [13] A. Squicciarini, S. Rajtmajer, Y. Liu, C. Griffin, Identification and characterization of cyberbullying dynamics in an online social network, in: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2015, pp. 280–285. doi:10.1145/2808797.2809398.
- [14] S. Ge, L. Cheng, H. Liu, Improving cyberbully detection with user interaction, CoRR abs/2011.00449 (2020). URL: <https://arxiv.org/abs/2011.00449>. arXiv:2011.00449.
- [15] J. Dooley, J. Pyżalski, D. Cross, Cyberbullying versus face-to-face bullying a theoretical and conceptual review, *Journal of Psychology* 217 (2009) 182–188. doi:10.1027/0044-3409.217.4.182.
- [16] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on Arabic social media, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 52–56. URL: <https://aclanthology.org/W17-3008>. doi:10.18653/v1/W17-3008.
- [17] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [18] A. Abdelali, K. Darwish, N. Durrani, H. Mubarak, Farasa: A fast and furious segmenter for Arabic, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, San Diego, California, 2016, pp. 11–16. URL: <https://aclanthology.org/N16-3003>. doi:10.18653/v1/N16-3003.
- [19] W. Antoun, F. Baly, H. Hajj, Arabert: Transformer-based model for arabic language understanding, in: LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020, 2020, p. 9.