Tuning Graph2vec with Node Labels for Abuse Detection in Online Conversations

Noé Cecillon, Richard Dufour, Vincent Labatut, Georges Linares

Tuning Graph2vec with Node Labels for Abuse Detection in Online Conversations

Noé Cécillon[1 [0000−0002−9889−0931]], Richard Dufour[1 [0000−0003−1203−9108]], Vincent Labatut[1 [0000−0002−2619−2835]], and Georges Linarès[1 [0000−0001−8049−9056]]

Laboratoire Informatique d'Avignon – LIA EA 4128, Avignon Université, France
{firstname.lastname}@univ-avignon.fr

1 Introduction

In recent years, online social media have allowed people to meet and discuss world-wide. These popular platforms are confronted with increasing abusive content. In order to automate the detection of abusive content in such social media, researchers have proposed various methods based on Natural Language Processing (NLP) [10, 1], and have leveraged behavioral information about users and the structure of conversations [3, 8].

In our previous work [11, 4], we proposed to combine NLP and conversational graph-based features to detect abusive messages in chat logs extracted from an online game. These conversational graphs model interactions between users (*i.e.* who is arguing with whom?), while completely ignoring the language content of the messages. We characterized the structure of these graphs by computing a large set of manually selected topological measures, and used them as features to train a classifier into detecting abusive messages.

Graph embedding methods allow representing graphs as low-dimensional vectors while preserving at least a part of their topological properties. In addition to the plain structure, certain methods are able to capture additional information such as node labels or the weight and direction of edges. These representations are automatically learned, so they have the advantage of not requiring to perform any feature selection or feature engineering. One can distinguish four main categories of graph embedding methods, depending on the nature of the considered objects: *node* [5, 12], *edge* [2], *subgraph* [13] and *whole-graph* [6, 9] embeddings. Each category better fits the needs of different applications and problems.

In this paper, we focus on the information that is used in addition to the plain structure by some embedding approaches. Especially, we study the impact of the node labels that are used by Graph2vec [9], a whole-graph embedding method. We study the effectiveness of such additional information in the context of online abuse detection.

2 Data and Method

We focus on a task consisting in detecting abusive messages in chat logs. This can be formulated as a classification problem consisting in deciding if a message

is abusive or not. Our dataset is the same as the one described in [11]. It is composed of 1,320 messages, equally distributed between the *Abuse* and *Non-abuse* classes.

In order to turn a chat log into a graph, we rely on an extraction method that we previously introduced in [11]. For each message, it produces a conversational graph whose nodes represent users and links model their interactions. It is built by leveraging the targeted message itself, but also the neighboring messages constituting the conversation to which it belongs. Classifying a message amounts to classifying the corresponding conversational graph. In our previous work [11], we experimented with a large set of 459 topological measures in order to get the most exhaustive representation of the graphs that we could, and fetched them to the classifier. We had to leverage feature selection to identify the most discriminant ones.

Here, instead, we use graph embedding approaches to automatically learn a representation of conversational graphs. Specifically, we use Graph2vec [9], a method that is able to represent a whole graph as a low-dimensional vector while preserving some of its topological properties. The algorithm takes the set of graphs to represent, and outputs their representations by applying a two-step process. It first identifies the subgraphs surrounding each node and constituting the graph. More precisely, it looks for so-called *rooted* subgraphs, *i.e.* node neighborhoods of a certain order. Second, these subgraphs are considered as the vocabulary and fetched to a *doc2vec SkipGram* [7] model. This embedding method captures *structural equivalence*, *i.e.* graphs whose nodes tend to possess this form of similarity will be close in the representation space.

In addition, this method requires the user to provide a label associated to each node. It does not have to be unique, for instance by default the method uses the node degree. We propose a few alternative labeling strategies, and assess how much they fit our specific situation. The first three can be considered as baselines. First, *Degree* is the default approach, that uses the degree of each node as its label. Second, *Random multiple* assigns a random label to every node by considering each graph separately. The same label can thus be assigned to distinct nodes over the whole corpus. Third, *Random unique* assigns a random label to every node, each label being unique in the whole corpus (and not only in the considered graph). The last three strategies are designed specifically for our situation. Fourth, as nodes correspond to users in our graphs, and users have unique IDs in our dataset, *Author ID* consists in using the ID of each node as its label. Fifth, *Distance to target* uses the distance to the targeted node (author of the targeted message) as the label. Sixth and finally, *Targeted* is a binary label depicting whether the node is the targeted node (here, the message to classify) or not.

3 Results

We use graph embedding approaches to generate vector representations of the conversational graphs, then fetch these representations to an SVM to perform

the classification. We set-up our experiments using a 10-fold cross-validation, using 70% of the data for training and the remaining 30% for testing.

Table 1. F-measures obtained by Graph2vec with different labeling strategies. The last two rows correspond to our previous method from [11], which relies on topological measures and its subset of *Top Features* (TF). The total runtime is expressed as *h*:*min*:*s*.

Node labeling strategy	Micro F-measure	Runtime
Degree	80.47	8:05
Random multiple	78.26	6:59
Random unique	79.00	7:04
Author ID	81.79	7:41
Distance to target	84.03	8:28
Targeted	81.90	7:10
Topological measures	88.08	8:19:10
Topological measures TF	86.01	14:10

Table 1 presents the micro F-measure values obtained by Graph2vec on our dataset with the different labeling strategies. It shows that node labels have an important impact on the classification performances. Unsurprisingly, the two random strategies yield the lowest performance, as their labels do not bring any information. In the case of *Random multiple* strategy, they can even introduce incorrect relations between nodes representing different users but sharing the same label. *Degree* is better than both other baselines, which suggests that nodes with the same degree might have a similar role in the graph. Maybe one does not need to distinguish between *individual* nodes, but can instead adopt a more generic approach and deal with *classes* of nodes. Strategies *Author ID* and *Targeted* are both above *Degree*, and perform approximately at the same level. This indicates that introducing a problem-related but non-structural information seems to improve the representation. Furthermore, *Targeted* performs slightly better than *Author ID* which corroborates our assumption regarding the possibility to handle nodes in a more generic way. The best performance is reached with the *Distance to target* strategy, which is centered around the targeted node, a key element of our problem. It allows combining both a generic description of the nodes and some problem-related aspects in the same representation.

The method we presented previously [11, 4], which relies on topological measures, outperforms all the strategies proposed in this article, with a 88.08 F-measure. However, our best strategy, *Distance to target*, achieves a promising 84.03 F-measure while being much more time efficient. *Topological measures TF* yields a better performance than *Distance to target* while having a runtime close to that of this strategy. However, it requires an important effort to determine the *top features*. Our experiments show that using appropriate labels with Graph2vec allows improving the quality of the generated embeddings. Based on this observation, we can suppose that other graph embedding methods that only focus on the graph structure to construct representations could benefit from such information. Hence, adapting such methods in order to incorporate additional information in the embedding process could reduce the gap between embedding approaches and our *Topological Measures* method.

References

1. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: 26th International Conference on World Wide Web Companion. pp. 759–760 (2017). https://doi.org/10.1145/3041021.3054223
2. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: 25th AAAI Conference on Artificial Intelligence. pp. 301–306 (2011), https://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659/3898
3. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A.: Mean birds: Detecting aggression and bullying on twitter. In: 2017 ACM on Web Science Conference. pp. 13–22 (2017). https://doi.org/10.1145/3091478.3091487
4. Cécillon, N., Labatut, V., Dufour, R., Linarès, G.: Abusive language detection in online conversations by combining content- and graph-based features. Frontiers in Big Data **2**, 8 (2019). https://doi.org/10.3389/fdata.2019.00008, https://www.frontiersin.org/article/10.3389/fdata.2019.00008
5. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864 (2016). https://doi.org/10.1145/2939672.2939754
6. de Lara, N., Pineau, E.: A simple baseline algorithm for graph classification. arXiv (2018), https://arxiv.org/pdf/1810.09155.pdf
7. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: 31st International Conference on International Conference on Machine Learning. vol. 32, p. II–1188–II–1196 (2014), http://proceedings.mlr.press/v32/le14.html
8. Mishra, P., Del Tredici, M., Yannakoudakis, H., Shutova, E.: Author profiling for abuse detection. In: 27th International Conference on Computational Linguistics. pp. 1088–1098 (Aug 2018), https://www.aclweb.org/anthology/C18-1093
9. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. In: 13th International Workshop on Mining and Learning with Graphs (MLG) (2017), http://www.mlgworkshop.org/2017/paper/MLG2017_paper_21.pdf
10. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: 25th International Conference on World Wide Web. pp. 145–153 (2016). https://doi.org/10.1145/2872427.2883062
11. Papegnies, E., Labatut, V., Dufour, R., Linarès, G.: Conversational networks for automatic online moderation. IEEE Trans. Comput. Social Systems **6**(1), 38–55 (Feb 2019). https://doi.org/10.1109/TCSS.2018.2887240
12. Perozzi, B., Kulkarni, V., Skiena, S.: Don't walk, skip! online learning of multi-scale network embeddings. In: 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 258–265 (2017). https://doi.org/10.1145/3110025.3110086
13. Yanardag, P., Vishwanathan, S.: Deep graph kernels. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1365–1374 (2015). https://doi.org/10.1145/2783258.2783417