Preliminary Results of Group Detection Technique Based on User to Vector Encoding

Giulio Biondi[1], Valentina Franzoni[1,2(✉)], and Alfredo Milani[1]

[1] Department of Mathematics and Computer Science, University of Perugia, Perugia, Italy
{giulio.biondi,valentina.franzoni,alfredo.milani}@unipg.it
[2] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

Abstract. This paper presents a novel approach for detecting groups of users based on observations of co-occurrences of user behavior. A Deep Neural Network is trained to encode users into a vector representation using an innovative adaptation of the word embedding architecture used in Natural Language Processing, which has been recently applied and modified for various domains, including graph data, recommender systems, and DNA gene sequence embedding. Preliminary experiments show promising results for the proposed adaptation to group detection based on the user-to-vector encoding derived from behavior observations in a variety of scenarios.

Keywords: user behaviour detection · user to vector encoding · machine learning

1 Introduction

Word2Vec [20] denotes a group of algorithms designed to learn high-order word embeddings from large textual datasets. Originally tested on word similarity tasks, the development of efficient word embedding techniques have enabled their application to a range of Natural Language Processing tasks, [13,14] including recommender systems, [1] sentiment analysis, [3,11,17] and question classification. [16,27] The underlying concept of Word2Vec has been successfully applied to various domains. For instance, in graph data analysis, the skip-gram technique has been utilized to generate higher-order graph node representations. [4,5,12,18] In this context, node ids are treated as words, and sentences are constructed from sequences of ids collected through random walks on the graph. Node embeddings have been employed to address several network tasks such as graph classification [26] and link prediction. [7] Other notable applications include recommender systems, [24] where items are embedded by leveraging their attributes and user interactions, and gene sequence embedding for understanding DNA replication mechanisms [25]. The versatility of Word2Vec and its

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14108, pp. 179–190, 2023.
https://doi.org/10.1007/978-3-031-37117-2_14

ability to learn meaningful representations of data has made it a popular tool in various fields. Its success in generating high-order word embeddings has inspired researchers to adapt the technique to other types of data, leading to the development of new methods and applications. As research in this area continues, we will likely see more innovative uses of Word2Vec and related techniques in the future.

In this work, we present preliminary results for a novel approach for detecting groups of users, based on observations of co-occurrences of user behavior. Observations are used to train a Deep Neural Network (DNN) to learn and encode users into a vector representation by an innovative adaptation of the popular word embedding technique used in Natural Language Processing. The paper is organized into four sections. Section 2 introduces a new method for encoding users into vectors for group detection, introducing a reference scenario for user observations, scheme and algorithmic details of the proposed group detection method, and the metrics used to assess performances. Section 3 presents the experimental phase including a description of the dataset, experimental settings, and a discussion of the results. Finally, Sect. 4 concludes the paper and summarizes the main findings.

2 User to Vector Encoding for Groups Detection

2.1 User Observation Scenario

We introduce for clarification purposes a scenario for user observation where group detection will be held. Let us assume, for example, that users belonging to known groups (e.g., classes of students in a given school) are observed during random social behavior. For instance, student detectors are available at school gates and record the entry/exit flow of students at the beginning/end of the school day.

An observation is a set of students observed while passing at the gates in a given time interval. It is easy to note, as shown in Fig. 1, that each observation is a set of individuals characterized by subgroups of students belonging to the same group/class, this is because people in the same class tend to exit school at the same moment, and tend to stay grouped at least in pairs or larger sets. At the same time, people in a subgroup tend to vary randomly, and subgroups in a single observation likely belong to different classes/groups [2]. On the other hand, observations with no subgroups (i.e. made only of single individuals all belonging to different classes/groups) are very unusual in a real social scenario, where people tend to interact and cluster. The goal is to collect multiple observations along days and devise a method to infer the groups to which the individual belongs, exploiting pattern regularities in subgroups observed over time. Despite Fig. 1 showing group labels by different graphical textures, in real observations data individuals would not be labeled with their group, but we assume every single person can be recognized through multiple passages.

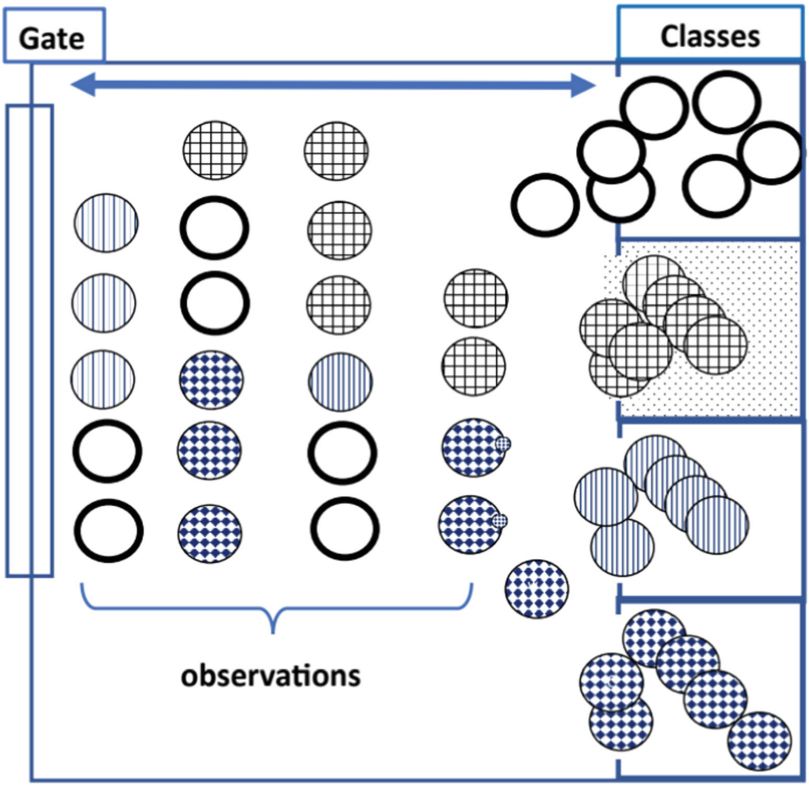

Fig. 1. Group Detection Scenario: classes of students

2.2 Proposed Method

The word encoding architecture, originally proposed in [20], was introduced to learn word embeddings (e.g., meaningful projections of words in a higher-dimensional vector space) from large corpora of texts. [6,8] In the vector space, words that frequently appear in the same context in a reference corpus are considered semantically close, then projected to close vector points according to a distance metric (e.g., euclidean distance) in the multidimensional space of embedding.

The core idea of our adaptation of word2Vect architecture is to assume observed users as single words and to consider observations of users as sentences composed of words. Therefore, conducting user-to-vector encoding on the user sequences represented by the observations would project in the vectorial multi-dimensional space users which co-occur more often in observations, representing them by closer points. Therefore, after computing user embedding, the set of a group of students can be easily detected by applying a geometric clustering technique to the encoded points, i.e. the generated vectorial representation.

The general workflow of the proposed method is shown in Fig. 2.

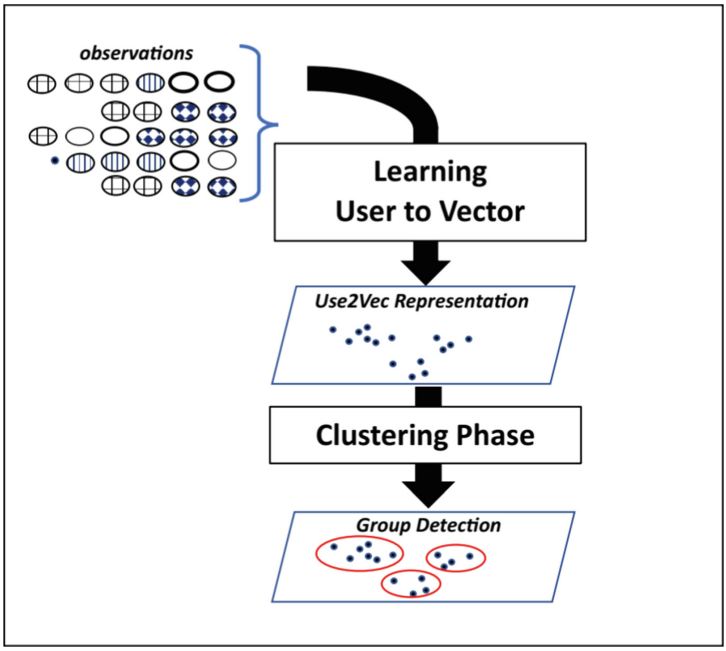

Fig. 2. Group Detection Workflow

Learning User Embedding. Embeddings are learned either through a Continuous-Bag-of-Words (CBOW) approach, predicting a word based on the surrounding context, or employing the skip-gram technique and predicting the context, given a word; in this work, we use the skip-gram technique. Word2Vec skip-gram-based methods require setting the values for the following hyper-parameters:

- *minimum count* parameter defines the minimum number of occurrences for a word to be considered significant and thus included in the vocabulary for subsequent embedding. [22] Unfrequent words are removed from the tokenized sentences in the preprocessing phase, before the embedding computation; a value of 1 means that all the words in the corpus are retained for embedding.
- *context size* c, or window size, defines the left and right context for a given word w: words in a sentence at a distance $<= c$ from w are considered semantically close to w, with weight proportional to their distance. Common window sizes for applications in large corpora are 2, 5, or 10.
- *embedding size*, specifies the number of components of the embedding vectors, in [21]; an intuitive criterion is that projecting the elements to be embedded in a higher-dimension space allows better discrimination, but too large values can be detrimental for performance.

In our implementation of user embedding, different parameters setting has been explored and tested on an experimental basis.

Clustering Vector Representations of User. A suitable candidate algorithm for clustering users (i.e., points in the multidimensional space obtained by the embedding phase) is K-Means [15,19]. K-means is an iterative clustering algorithm that partitions elements in a domain by randomly computing a list of points, called led *centroids*, in the multidimensional space of the domain, and assigning each element to the closest centroid. The centroids are re-computed by averaging all the elements in the group, which are then re-assigned to the new closest centroid; the aim is to minimize the intra-cluster sum of square differences between each point and the centroid. The desired number of clusters is a parameter of the algorithm, along with the number of iterations and the specification of the k-means procedure, e.g., *lloyd*, and *elkan* procedures.

2.3 Metrics

The following metrics are typically used in the literature for assessing clustering quality compared to the ground truth (i.e., the real groups to which the users belong); all the metrics have a lower bound of 0, indicating total independency between the clustering results and the ground truth labels, and an upper bound of 1 for perfect labeling.

- *Homogeneity score* [9,23] describes to which extent the points in the computed clusters belong to the same ground-truth group, ranging from 0 (i.e., each cluster contains two elements from the same group) to 1 (i.e., completely homogeneous clusters).
- *Completeness score* [23] measures the rate of aggregation of elements from the same group in the computed clusters: a value of 0 means that all the elements from the same group fall into different clusters, while 1 means that the fall into the same cluster.
- *V_measure score* [23] is the harmonic mean of homogeneity and completeness.
- *Fowlkes-Mallows score* (FMI) [10] is the geometric mean of the pairwise precision and recall measures:

$$FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}}$$

  where TP counts the pairs in the same group clusterized together, FP the pairs in the same group but not in the same cluster, and FN in the same cluster but not in the same group.

3 Experiments

3.1 Observations Dataset

Two different scenarios derived from the reference user observation scenario (see Sect. 2.1) were analyzed, with 10 groups/classes of 10 elements/students each, and 10 groups/classes of 25 elements/students each. (*Scenario 1*)(*Scenario 2*). For each grouping, 4 datasets of observations have been randomly produced, containing respectively 1000, 2500, 5000, and 10000 observations with random elements; each observation contains subgroups of size from 2 to 5 elements from 3 different groups, thus having a minimum length of 6 and a maximum length of 15. We applied randomization using a Gaussian normal distribution to every decision step of the generation of a single observation. Decisions about which groups to include, the size of each subgroup, and the extraction of each individual from its group are also randomized in the same way.

3.2 Experiment Settings

The following settings are applied to the *user to vector learning* phase: the window size is set to 5, and the minimum count to 1 since no individual user had to be discarded when building the embedding. For each scenario, various embedding sizes are considered (i.e., 2, 5, 10, 25, 50, 100, and 200). The k-means algorithm is used to cluster the individuals according to the embeddings created from the user observations; the number of desired clusters is set to 10 for all the experiments (i.e., equal to the number of groups/classes in the ground truth), k-means++ is the chosen initialization technique.

3.3 Experimental Results

In Table 1 the results for experiments with 10 groups, of 10 individuals each, are reported. In Table 2 the results for experiments with 10 groups, of 25 individuals each, are reported. In Table 2 the results for experiments with 10 groups, of 25 individuals each, are reported. It is apparent that, regardless of the scenario and the vector size, 1000 observations are not sufficient to learn any meaningful embedding. Scaling with the number of observations, Scenario 1 Word2Vec produces embeddings that allow perfect clustering with 2500 observations, albeit for high embedding sizes only (i.e., 100, and 200). The same number of observations proves not sufficient for Scenario 2, under any setting (Table 3).

Table 1. Test 1 results: scenario 1 (10 groups, 10 individuals per group)

# Observations	Embedding size	Homogeneity	Completeness	V_Measure	Fowlkes-Mallows
1000	2	0.2363	0.2439	0.2401	0.1146
1000	3	0.2445	0.2476	0.2461	0.1119
1000	5	0.2194	0.2251	0.2222	0.0998
1000	10	0.2570	0.2631	0.2600	0.1238
1000	25	0.2366	0.2429	0.2397	0.1171
1000	50	0.2383	0.2541	0.2459	0.1185
1000	100	0.1702	0.1877	0.1785	0.0974
1000	200	0.1561	0.1682	0.1620	0.0866
2500	2	0.2756	0.2911	0.2832	0.1473
2500	3	0.3643	0.3829	0.3734	0.1917
2500	5	0.5530	0.5645	0.5587	0.3870
2500	10	0.5810	0.6086	0.5944	0.4106
2500	25	0.9062	0.9284	0.9172	0.8332
2500	50	0.9177	0.9427	0.9300	0.8496
2500	100	1.0000	1.0000	1.0000	1.0000
2500	200	1.0000	1.0000	1.0000	1.0000
5000	2	0.5916	0.6079	0.5997	0.3777
5000	3	0.8378	0.8801	0.8585	0.7231
5000	5	1.0000	1.0000	1.0000	1.0000
5000	10	1.0000	1.0000	1.0000	1.0000
5000	25	0.9398	0.9698	0.9546	0.8756
5000	50	1.0000	1.0000	1.0000	1.0000
5000	100	1.0000	1.0000	1.0000	1.0000
5000	200	1.0000	1.0000	1.0000	1.0000
10000	2	0.7784	0.7934	0.7859	0.6639
10000	3	0.9854	0.9859	0.9857	0.9789
10000	5	1.0000	1.0000	1.0000	1.0000
10000	10	1.0000	1.0000	1.0000	1.0000
10000	25	1.0000	1.0000	1.0000	1.0000
10000	50	1.0000	1.0000	1.0000	1.0000
10000	100	1.0000	1.0000	1.0000	1.0000
10000	200	1.0000	1.0000	1.0000	1.0000

Table 2. Test 2 results: Scenario 2(10 groups, 25 individuals per group)

# Observations	Embedding size	Homogeneity	Completeness	V_Measure	Fowlkes-Mallows
1000	2	0.0721	0.0728	0.0724	0.0948
1000	3	0.0715	0.0720	0.0718	0.0944
1000	5	0.1000	0.1017	0.1009	0.1087
1000	10	0.0652	0.0657	0.0654	0.0946
1000	25	0.0864	0.0907	0.0885	0.1102
1000	50	0.0798	0.0836	0.0817	0.1104
1000	100	0.0900	0.0926	0.0913	0.1060
1000	200	0.0855	0.0938	0.0895	0.1124
2500	2	0.1587	0.1663	0.1624	0.1345
2500	3	0.1795	0.1822	0.1808	0.1468
2500	5	0.1919	0.1930	0.1924	0.1535
2500	10	0.2972	0.2987	0.2979	0.2243
2500	25	0.4061	0.4123	0.4091	0.3130
2500	50	0.5160	0.5205	0.5182	0.3906
2500	100	0.5847	0.5987	0.5916	0.4561
2500	200	0.4759	0.4827	0.4793	0.3325
5000	2	0.4631	0.4767	0.4698	0.3068
5000	3	0.5782	0.5874	0.5827	0.4529
5000	5	0.9231	0.9236	0.9233	0.8961
5000	10	0.9398	0.9690	0.9542	0.8816
5000	25	1.0000	1.0000	1.0000	1.0000
5000	50	1.0000	1.0000	1.0000	1.0000
5000	100	1.0000	1.0000	1.0000	1.0000
5000	200	1.0000	1.0000	1.0000	1.0000
10000	2	0.6764	0.7023	0.6891	0.5324
10000	3	0.9780	0.9781	0.9780	0.9757
10000	5	1.0000	1.0000	1.0000	1.0000
10000	10	1.0000	1.0000	1.0000	1.0000
10000	25	1.0000	1.0000	1.0000	1.0000
10000	50	1.0000	1.0000	1.0000	1.0000
10000	100	1.0000	1.0000	1.0000	1.0000
10000	200	1.0000	1.0000	1.0000	1.0000

Table 3. Test 3 results: Scenario 2 (10 groups, 25 individuals per group), 20 clusters

# Observations	Embedding size	Homogeneity	Completeness	V_Measure	Fowlkes-Mallows
1000	2	0.1870	0.1496	0.1662	0.0814
1000	3	0.1809	0.1402	0.1580	0.0675
1000	5	0.1494	0.1168	0.1311	0.0622
1000	10	0.1751	0.1391	0.1551	0.0752
1000	25	0.1803	0.1421	0.1589	0.0716
1000	50	0.1652	0.1345	0.1483	0.0772
1000	100	0.1761	0.1405	0.1563	0.0728
1000	200	0.1824	0.1442	0.1611	0.0716
2500	2	0.2452	0.1950	0.2172	0.1006
2500	3	0.2794	0.2197	0.2460	0.1208
2500	5	0.2863	0.2239	0.2513	0.1206
2500	10	0.3703	0.2889	0.3246	0.1623
2500	25	0.4926	0.3844	0.4318	0.2488
2500	50	0.5932	0.4654	0.5216	0.3256
2500	100	0.8078	0.6312	0.7087	0.5529
2500	200	0.7153	0.5637	0.6305	0.4395
5000	2	0.5005	0.3961	0.4422	0.2144
5000	3	0.6501	0.5104	0.5719	0.3514
5000	5	0.8976	0.7124	0.7944	0.6572
5000	10	0.9945	0.8038	0.8891	0.7876
5000	25	1.0000	0.8146	0.8978	0.7971
5000	50	1.0000	0.8231	0.9030	0.8260
5000	100	1.0000	0.8087	0.8942	0.7887
5000	200	1.0000	0.8133	0.8970	0.7979
10000	2	0.7321	0.5814	0.6481	0.4005
10000	3	0.9662	0.7641	0.8533	0.7185
10000	5	1.0000	0.8159	0.8986	0.8149
10000	10	1.0000	0.7928	0.8844	0.7559
10000	25	1.0000	0.7940	0.8852	0.7625
10000	50	1.0000	0.7945	0.8855	0.7646
10000	100	1.0000	0.8030	0.8907	0.7755
10000	200	1.0000	0.8023	0.8903	0.7802

4 Conclusions

In this work, we have introduced a novel technique of user group detection based on user-to-vector encoding learned from user behavior observations of a random group of unknown users. The approach leverages the co-occurrence of users in observations, which is used to induce group information. The preliminary experimental results are quite promising, yet a deeper and more systematic investigation is needed to evaluate the robustness of the approach in the presence of noisy observations and to use data from real domains. Moreover, results suggest that it is worth exploring the potential applications of techniques of learning vectorial representation embedding user behavior in a wide variety of domains, which will be the target of future works.

Acknowledgments. This work is partially supported by the Italian Ministry of Research under PRIN Project "PHRAME" Grant n.20178XXKFY. Supported also by KIT-Lab, Knowledge, and Information Technology Laboratory, University of Perugia, Italy.

References

1. Baek, J.W., Chung, K.Y.: Multimedia recommendation using word2vec-based social relationship mining. Multim. Tools Appl. **80**, 34499–34515 (2021). https://doi.org/10.1007/s11042-019-08607-9
2. Carpi, A., D'Alessandro, F.: On the commutative equivalence of bounded semilinear codes. In: Mercaş, R., Reidenbach, D. (eds.) WORDS 2019. LNCS, vol. 11682, pp. 119–132. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28796-2_9
3. Chen, Q., Sokolova, M.: Specialists, scientists, and sentiments: Word2Vec and Doc2Vec in analysis of scientific and medical texts. SN Comput. Sci. **2**(5), 1–11 (2021). https://doi.org/10.1007/s42979-021-00807-1
4. Chiancone, A., Franzoni, V., Li, Y., Markov, K., Milani, A.: Leveraging zero tail in neighbourhood for link prediction. In: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 3, pp. 135–139. IEEE (2015)
5. Chiancone, A., Milani, A., Poggioni, V., Pallottelli, S., Madotto, A., Franzoni, V.: A multistrain bacterial model for link prediction andrea chiancone. In: 2015 11th International Conference on Natural Computation (ICNC), pp. 1075–1079. IEEE (2015)
6. D'Alessandro, F., Carpi, A.: On incomplete and synchronizing finite sets. Theoret. Comput. Sci. **664**, 67–77 (2017). https://doi.org/10.1016/j.tcs.2015.08.042
7. Du, X., Yan, J., Zhang, R., Zha, H.: Cross-network skip-gram embedding for joint network alignment and link prediction. IEEE Trans. Knowl. Data Eng. **34**(3), 1080–1095 (2022). https://doi.org/10.1109/TKDE.2020.2997861
8. Carpi, A., D'Alessandro, F.: On the commutative equivalence of context-free languages. In: Hoshi, M., Seki, S. (eds.) DLT 2018. LNCS, vol. 11088, pp. 169–181. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98654-8_14

9. Carpi, A., D'Alessandro, F.: On the hybrid černý-road coloring problem and Hamiltonian paths. In: Gao, Y., Lu, H., Seki, S., Yu, S. (eds.) DLT 2010. LNCS, vol. 6224, pp. 124–135. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14455-4_13
10. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Statist. Assoc. **78**(383), 553–569 (1983). https://doi.org/10.1080/01621459.1983.10478008
11. Franzoni, V., Biondi, G., Milani, A.: A Web-Based System for Emotion Vector Extraction. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10406, pp. 653–668. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62398-6_46
12. Franzoni, V., Chiancone, A., Milani, A.: A multistrain bacterial diffusion model for link prediction. Int. J. Pattern Recognit. Artif. Intell. **31**(11), 1759024 (2017)
13. Franzoni, V., Li, Y., Mengoni, P., Milani, A.: Clustering Facebook for biased context extraction. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10404, pp. 717–729. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62392-4_52
14. Franzoni, V., Milani, A.: A pheromone-like model for semantic context extraction from collaborative networks. In: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 540–547. IEEE (2015)
15. Franzoni, V., Milani, A.: A semantic comparison of clustering algorithms for the evaluation of web-based similarity measures. In: Gervasi, O., et al. (eds.) ICCSA 2016. LNCS, vol. 9790, pp. 438–452. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42092-9_34
16. Franzoni, V., Milani, A., Mengoni, P., Piccinato, F.: Artificial intelligence visual metaphors in e-learning interfaces for learning analytics. Appl. Sci. **10**(20), 7195 (2020)
17. Franzoni, V., Poggioni, V., Zollo, F.: Automated classification of book blurbs according to the emotional tags of the social network Zazie. ESSEM@ AI* IA **1096**, 83–94 (2013)
18. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks (2016)
19. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc. (2013)
21. Patel, K., Bhattacharyya, P.: Towards lower bounds on number of dimensions for word embeddings. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 31–36. Asian Federation of Natural Language Processing, Taipei (2017)
22. Poggioni, V., Bartoccini, U., Carpi, A., Santucci, V.: Memes evolution in a memetic variant of particle swarm optimization. Mathematics **7**(5) (2019). https://doi.org/10.3390/math7050423
23. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 410–420. Association for Computational Linguistics, Prague (2007)
24. Vasile, F., Smirnova, E., Conneau, A.: Meta-prod2vec: Product embeddings using side-information for recommendation. In: RecSys '16, pp. 225–232. Association

for Computing Machinery, New York (2016). https://doi.org/10.1145/2959100.2959160
25. Wu, F., Yang, R., Zhang, C., Zhang, L.: A deep learning framework combined with word embedding to identify DNA replication origins. Sci. Rep. **11**, 844 (2021). https://doi.org/10.1038/s41598-020-80670-x
26. Xie, Y., Liang, Y., Gong, M., Qin, A.K., Ong, Y.S., He, T.: Semisupervised graph neural networks for graph classification. In: IEEE Transactions on Cybernetics, pp. 1–14 (2022). https://doi.org/10.1109/TCYB.2022.3164696
27. Yilmaz, S., Toklu, S.: A deep learning analysis on question classification task using Word2vec representations. Neural Comput. Appl. **32**(7), 2909–2928 (2020). https://doi.org/10.1007/s00521-020-04725-w