DeepBlock: A Novel Blocking Approach for Entity Resolution using Deep Learning

Delaram Javdani
Iran University of Science and Technology
Tehran, Iran
d_javdani@cmps2.iust.ac.ir

Hossein Rahmani
Iran University of Science and Technology
Tehran, Iran
h_rahmani@iust.ac.ir

Milad Allahgholi
Iran University of Science and Technology
Tehran, Iran
milad_allahgholi@comp.iust.ac.ir

Fatemeh Karimkhani
Iran University of Science and Technology
Tehran, Iran
f_karimkhani@comp.iust.ac.ir

***Abstract*—Entity resolution refers to the process of identifying and integrating records belonging to unique entities. The standard methods are using a rule-based or machine learning models to compare and assign a point, to indicate the status of matching or non-matching the pair of records. However, a comprehensive comparison across all the records pairs leads to a second-order matching complexity. Therefore blocking methods are using before the matching, to group the same entities into small blocks. Then the matching operation is done comprehensively. Several blocking methods provided to efficiently block the input data into manageable groups, including the token blocking, that holds records with a similar token in the same block. Most of the previous methods did not take any semantic criteria into account. In this paper, we propose a new method, called DeepBlock that uses deep learning for the task of blocking in entity resolution. DeepBlock combines syntactic and semantic similarities to calculate the similarity between records. We have evaluated the DeepBlock over the real-world dataset and compared it with the existing blocking technique (token blocking). Our experimental result shows that the combination of semantic and syntactic similarity can considerably improve the quality of blocking. The results show that DeepBlock outperforms the token blocking method significantly with respect to pair quality (PQ) measure.**

Keywords— Entity Resolution, Blocking, Deep Learning

I. INTRODUCTION

Integrating two or more data sources in the absence of a unique identifier is a perennial and challenging problem that not only causes redundancy of data but also inaccurate processing of queries and extracting knowledge [1]. Entity resolution used to identify, match and integrate the records of an entity in different data sources [2]. There are two challenges in entity resolution. First, the various titles of entity resolution in the literature [3] such as Record Linkage [4-9], Duplicate Detection [10-12], Reference Matching [13]. Second, data uncertainty such as differences in writing, abbreviations, different schema, and null values. Entity resolution is the task of information integration, which has many practical uses in the implementation, integration, and elimination of duplication in large and varied data sources [14]. One of the fundamental steps in entity resolution is the calculation of the similarity between the pair of records. Since the total number of pairwise similarities with the size of the input data is second-order tasks, performing the above tasks in a large dataset is problematic. Additionally, even for a small dataset, estimating the total similarity matrix can be very difficult using costly similarity functions. At the same time, the majority of computational similarities are unnecessary [15].

Blocking methods divide the dataset into blocks or clusters of records that are common in a block property or similar to a certain metric. Blocking in entity resolution groups the records into a set of small blocks, and only records that represent a similar entity place in the same block [16]. The purpose of these techniques is deleting records that do not match [14]. One of these methods is token blocking that places records with the same token in the same block [17]. Most of the blocking methods did not take any semantic criteria into account. In this paper, we propose a new method called DeepBlock that uses deep learning for the task of blocking in entity resolution. DeepBlock uses syntactic and semantic similarities to calculate the similarity between records. DeepBlock evaluated over the real-world data set. The experimental results show that integrating syntactic and semantic features into the blocking process can significantly improve the quality of blocks.

The structure of this article is as follows: Section 2 explains the previous blocking methods. The DeepBlock model described in detail in Section 3. The results of the DeepBlock discussed in Section 4. Section 5 provides conclusions and future work.

II. RELATED WORK

Several studies have been conducted on blocking that categorized in 5 main categories such as schema-based and schema-agnostic blocking techniques, meta-blocking, block processing techniques and parallel blocking techniques (see Fig. 1).

First, schema-based blocking techniques such as standard blocking [5, 18], sorted neighborhood [19], the extended sorted neighborhood [20], q-grams blocking [21], extended q-grams blocking [20], MFIBlocks [22], canopy clustering [23, 24], extended canopy clustering [20], suffix arrays [25] and extended suffix arrays [20] were proposed, that generates blocks based on the blocking key. There were two problems with these methods. First, the choice of features to combine was a hard and error-prone process that requires domain experts. For this purpose, the classification algorithm can be used, but it needs tagged data. Second, if two datasets have different schemas, schema coordination between dataset must be implemented, before the implementation of entity

978-1-7281-1431-6/19/$31.00 ©2019 IEEE

resolution. But unfortunately for the heterogeneous and full of noise data, schema coordination techniques are no longer applicable [26].

Second, the schema-agnostic blocking approaches such as token blocking, attribute-clustering blocking [27], TYPiMatch [28] and prefix-infix-suffix blocking [29] were suggested, which do not use any schema information. Token blocking is the most common schema-agnostic method. Each token in the dataset is considered as a blocking key. Therefore, each block contains all the records having the same token. The schema-agnostic techniques reduced the likelihood of loss of matching and increased the likelihood of insertion of non-matching records in the same block by placing each record in multiple blocks, which resulted in a high recall but cost in precision [26].

Third, the meta-blocking approaches [30] such as BLAST [26], BLOSS [31], supervise meta-blocking [32] and multi-core meta-blocking [33] were proposed to rebuild a set of blocks, to keep the most promising comparisons. To do so, the set of blocks shows by the weighted graph, called a blocking graph. In this graph, each record shows by node and edges exist between two nodes if the corresponding records appear at least in one block together. The weights of the edges were calculated for the match probability. Then pruning algorithms such as considering for each node all its adjacent edges and retain only those having a weight higher than the local average, were applied on weighted edges. In the end, each pair of nodes connected by edge created a new block [30].

Forth, the block processing techniques such as block purging [34], block filtering [35], block clustering [36], comparison propagation [37] and iterative blocking [38] with the aim of eliminating comparisons repeated across different blocks and comparisons involved non-matching entities, without affecting matching comparisons were proposed.

Finally, The parallel methods such as parallel standard blocking [39], parallel block filtering [40], parallel meta-blocking [41], parallel token blocking [29], block attribute-clustering blocking [29] and prefix-infix-suffix blocking [29] were suggested, which combines previous blocking methods with a map-reduce algorithm.

III. The Proposed Method (DeepBlock)

In this section, we propose a new method, called DeepBlock, for blocking in entity resolution using deep learning, which includes six steps. Fig.2 shows the DeepBlock steps. In the first step, data blocked by applying token blocking method and the records with the same token fall into the same block. In the second step, the blocking keys obtained so that their similarity calculated in the third step. In the third step, the similarity of the blocking keys is calculated using a syntactic similarity criterion which is a combination of Soundex [42], edit-distance [43], hamming distance [44] and Jaro-Winkler distance [45] and a semantic similarity criterion which applying deep learning methods such as Word2Vec [46, 47] and word movers' distance (WMD) [48].

Word2Vec is the method proposed by Google in 2013, which is a very efficient way of displaying words and texts and processing them. In this method, with the help of the neural network, a small size vector computed for displaying all the words and texts, in the model training phase. In this vector, each column only displays one number. If we assume that the vector size is N, then we will have N-dimensional

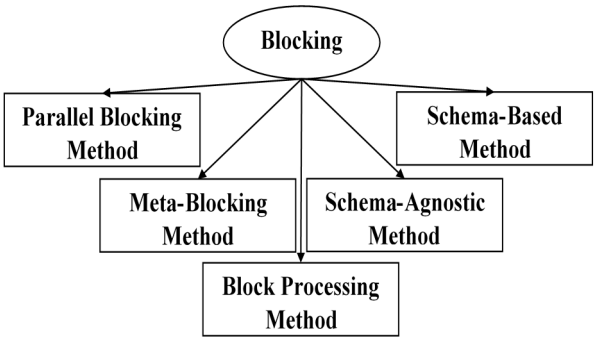

Fig 1. *5 main categories of blocking techniques*

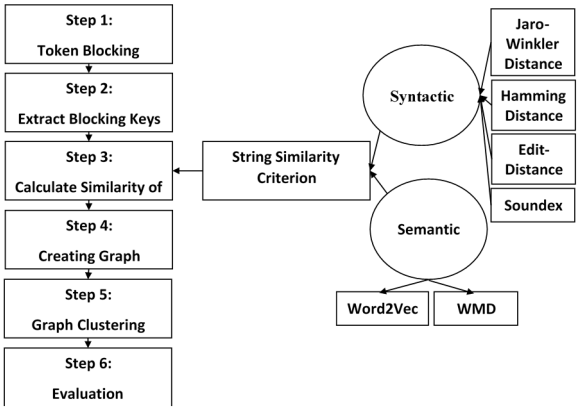

Fig 2. *6 main steps of DeepBlock*

space and each word in this space will have a unique representation. To increase the accuracy of this method, the required data set to train the model should include about a billion words that are used within millions of documents or text [46, 47]. DeepBlock calculates the similarity between the blocking keys using Word2Vec. Also, DeepBlock uses WMD to calculate semantic similarity. WMD is a method that enables us to evaluate the distance between two sentences in a meaningful manner, even when they have no common words. WMD uses word vectors generated by the Word2Vec method [48].

Representation of the relationships between the blocks can simplify analyzing them, and we can find patterns that cannot be reflected in the flat analysis (non-graphic) of the blocks. For this reason, DeepBlock displayed similar records with a graph in the fourth step. Graph's nodes are records. Clustering among all the algorithms that apply to the graph, provides a high-level view for the viewers. So, in the fifth step, we cluster the graph. In the last step, we evaluate our proposed model using block quality criteria such as pair completeness (PC) [16] and pairs quality (PQ) [16] criteria. The PC (recall) calculates the degree to which blocks retain true matches using (1):

$$PC = detected\ matches\ /\ existing\ matches \quad (1)$$

The PQ (precision) calculates the useful comparisons, in other words, calculates the percentage of true matches in the pairs of the blocks using (2):

$$PQ = detected\ matches\ /\ executed\ comparisons \quad (2)$$

IV. EVALUATION

In this section, we evaluate the DeepBlock using the PC and PQ criteria and compare DeepBlock with token blocking method.

A. Dataset

To evaluate the DeepBlock, we use CORA dataset [49], which includes 1,879 machine learning publications. The number of a profile in CORA is 1,300, the number of features is 12 and the number of available matches is 17,000. We are using the article's titles for blocking.

B. Results

In this section, we evaluate DeepBlock. To evaluate the DeepBlock, first, we tokenized the article's titles. We put the articles with the same token in the same block (token blocking). Second, we extracted blocking keys. Third, we calculated the similarity between the tokens by combining the syntactic and semantic similarity criteria (explained in Section 3).

For semantic similarity, we used Word2Vec which applied Skip-Gram learning algorithm [46, 47] and vector size of 200. Also, we obtained the similarity of the article's titles using the WMD and combined the results with the previous similarities to improve the results. Fourth, we created the graph of similar records. Fifth, we clustered the graph (see Fig 3). And finally, records in the same cluster formed a block.

We applied two common measures to evaluate the quality of blocking: Pair Completeness (PC) and Pair quality (PQ). According to the experimental results, the number of blocks has been reduced compared to token blocking. Also in the DeepBlock, there were no overlaps between the blocks. Moreover, in the DeepBlock, PQ has increased compared to token blocking (see Fig. 4)).

Additionally, by applying semantic similarity using deep learning methods, DeepBlock succeeded to discover the same articles with different titles and put them in the same block. Table I shows some of these titles.

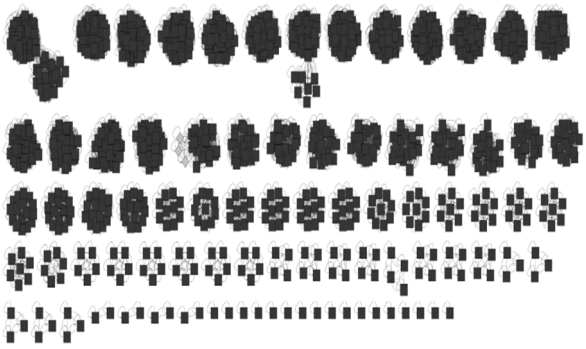

Fig 3. *The fifth step of DeepBlock which cluster the graph of records.*

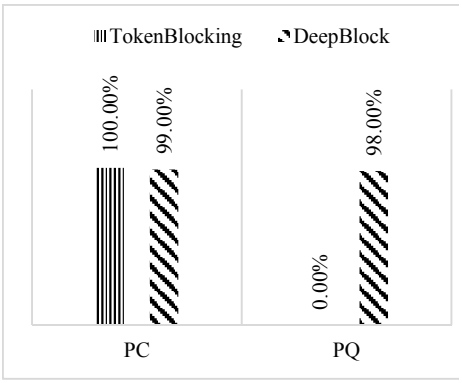

Fig 4. *PC and PQ of token blocking and DeepBlock*

TABLE I . SAME ARTICLES WITH DIFFERENT TITLE

	TITLE OF ARTICLE	TITLE OF ARTICLE
1	A BOUND ON THE ERROR OF CROSS VALIDATION USING THE APPROXIMATION AND ESTIMATION RATES, WITH CONSEQUENCES FOR THE TRAINING-TEST SPLIT.	A BOUND ON THE ERROR OF CROSS VALIDATION, WITH CONSEQUENCES FOR THE TRAINING-TEST SPLIT.
2	EXACT IDENTIFICATION OF READ-ONCE FORMULAS USING FIXED POINTS OF AMPLIFICATION FUNCTIONS.	EXACT IDENTIFICATION OF CIRCUITS USING FIXED POINTS OF AMPLIFICATION FUNCTIONS.
3	BOOSTING PERFORMANCE IN NEURAL NETWORKS.	IMPROVING PERFORMANCE IN NEURAL NETWORKS USING A BOOSTING ALGORITHM.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new method for blocking in entity resolution using deep learning, called DeepBlock. DeepBlock uses syntactic and semantic similarities to calculate the similarity between records, grouping similar records in the same blocks. We experimentally evaluated it on the real-world dataset. The experimental results showed how DeepBlock outperforms the existing blocking approaches. According to the experimental results in a DeepBlock, the pair quality (PQ) have increased significantly compared to token blocking. In the future, we intend to investigate the effect of our method in the larger data set or other languages articles such as Persian's articles. Moreover, we want to combine Word2Vec with another algorithm such as TF-IDF and N-Gram. Also, we can use another learning algorithm such as CBOW instead of Skip-Gram for Word2Vec.

REFERENCES

[1] Bhattacharya, Indrajit, and Lise Getoor. "Collective entity resolution in relational data." ACM Transactions on Knowledge Discovery from Data (TKDD) 1, no. 1 (2007): 5.

[2] Christen, Peter. Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer Science & Business Media, 2012.

[3] Getoor, Lise, and Ashwin Machanavajjhala. "Entity resolution for big data." In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1527-1527. ACM, 2013.

[4] Newcombe, Howard B., James M. Kennedy, S. J. Axford, and Allison P. James. "Automatic linkage of vital records." Science 130, no. 3381 (1959): 954-959.

[5] Fellegi, Ivan P., and Alan B. Sunter. "A theory for record linkage." Journal of the American Statistical Association 64, no. 328 (1969): 1183-1210.

[6] Winkler, William E. "The state of record linkage and current research problems." In the Statistical Research Division, US Census Bureau. 1999.

[7] Bhattacharya, Indrajit, and Lise Getoor. "Iterative record linkage for cleaning and integration." In Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pp. 11-18. ACM, 2004.

[8] Ravikumar, Pradeep, and William W. Cohen. "A hierarchical graphical model for record linkage." In Proceedings of the 20th conference on Uncertainty in artificial intelligence, pp. 454-461. AUAI Press, 2004.

[9] Winkler, William E. Methods for record linkage and Bayesian networks. Technical report, Statistical Research Division, US Census Bureau, Washington, DC, 2002.
[10] Monge, Alvaro, and Charles Elkan. "An efficient domain-independent algorithm for detecting approximately duplicate database records." (1997).
[11] Sarawagi, Sunita, and Anuradha Bhamidipaty. "Interactive deduplication using active learning." In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 269-278. ACM, 2002.
[12] Ananthakrishna, Rohit, Surajit Chaudhuri, and Venkatesh Ganti. "Eliminating fuzzy duplicates in data warehouses." In VLDB'02: Proceedings of the 28th International Conference on Very Large Databases, pp. 586-597. Morgan Kaufmann, 2002.
[13] McCallum, Andrew, Kamal Nigam, and Lyle H. Ungar. "Efficient clustering of high-dimensional data sets with application to reference matching." In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 169-178. ACM, 2000.
[14] De Vries, Timothy, Hui Ke, Sanjay Chawla, and Peter Christen. "Robust record linkage blocking using suffix arrays and Bloom filters." ACM Transactions on Knowledge Discovery from Data (TKDD) 5, no. 2 (2011): 9.
[15] Bilenko, Mikhail, Beena Kamath, and Raymond J. Mooney. "Adaptive blocking: Learning to scale up record linkage." In Sixth International Conference on Data Mining (ICDM'06), pp. 87-96. IEEE, 2006.
[16] Wang, Qing, Mingyuan Cui, and Huizhi Liang. "Semantic-aware blocking for entity resolution." IEEE Transactions on Knowledge and Data Engineering 28, no. 1 (2016): 166-180.
[17] Papadakis, George, Ekaterini Ioannou, Claudia Niederée, and Peter Fankhauser. "Efficient entity resolution for large heterogeneous information spaces." In Proceedings of the fourth ACM international conference on Web search and data mining, pp. 535-544. ACM, 2011.
[18] Papadakis, George, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. "Comparative analysis of approximate blocking techniques for entity resolution." Proceedings of the VLDB Endowment 9, no. 9 (2016): 684-695.
[19] Hernández, Mauricio A., and Salvatore J. Stolfo. "The merge/purge problem for large databases." In ACM Sigmod Record, vol. 24, no. 2, pp. 127-138. ACM, 1995.
[20] Christen, Peter. "A survey of indexing techniques for scalable record linkage and deduplication." IEEE transactions on knowledge and data engineering 24, no. 9 (2012): 1537-1555.
[21] Gravano, Luis, Panagiotis G. Ipeirotis, Hosagrahar Visvesvaraya Jagadish, Nick Koudas, Shanmugauelayut Muthukrishnan, and Divesh Srivastava. "Approximate string joins in a database (almost) for free." In VLDB, vol. 1, pp. 491-500. 2001.
[22] Kenig, Batya, and Avigdor Gal. "MFIBlocks: An effective blocking algorithm for entity resolution." Information Systems 38, no. 6 (2013): 908-926.
[23] McCallum, Andrew, Kamal Nigam, and Lyle H. Ungar. "Efficient clustering of high-dimensional data sets with application to reference matching." In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 169-178. ACM, 2000.
[24] Baxter, R., P. Christen, and T. Churches. "A Comparison of Fast Blocking Methods for Record Linkage." In Proceedings of the Workshop on Data Cleaning, Record Linkage and Object Consolidation at the Ninth ACM SIGKDD international conference on Knowledge Discovery and data mining, Washington DC, 2003.
[25] Aizawa, Akiko, and Keizo Oyama. "A fast linkage detection scheme for multi-source information integration." In International Workshop on Challenges in Web Information Retrieval and Integration, pp. 30-39. IEEE, 2005.
[26] Simonini, Giovanni, Sonia Bergamaschi, and H. V. Jagadish. "BLAST: a loosely schema-aware meta-blocking approach for entity resolution." Proceedings of the VLDB Endowment 9, no. 12 (2016): 1173-1184.
[27] Papadakis, George, Ekaterini Ioannou, Themis Palpanas, Claudia Niederee, and Wolfgang Nejdl. "A blocking framework for entity resolution in highly heterogeneous information spaces." IEEE Transactions on Knowledge and Data Engineering 25, no. 12 (2013): 2665-2682.
[28] Ma, Yongtao, and Thanh Tran. "Typimatch: Type-specific unsupervised learning of keys and key values for heterogeneous web data integration." In Proceedings of the sixth ACM international conference on Web search and data mining, pp. 325-334. ACM, 2013.
[29] Efthymiou, Vasilis, Kostas Stefanidis, and Vassilis Christophides. "Benchmarking blocking algorithms for web entities." IEEE Transactions on Big Data (2016).
[30] Papadakis, George, Georgia Koutrika, Themis Palpanas, and Wolfgang Nejdl. "Meta-blocking: Taking entity resolution to the next level." IEEE Transactions on Knowledge and Data Engineering 26, no. 8 (2014): 1946-1960.
[31] Dal Bianco, Guilherme, Marcos André Gonçalves, and Denio Duarte. "BLOSS: Effective meta-blocking with almost no effort." Information Systems 75 (2018): 75-89.
[32] Papadakis, George, George Papastefanatos, and Georgia Koutrika. "Supervised meta-blocking." Proceedings of the VLDB Endowment 7, no. 14 (2014): 1929-1940.
[33] Papadakis, George, Konstantina Bereta, Themis Palpanas, and Manolis Koubarakis. "Multi-core meta-blocking for big linked data." In Proceedings of the 13th International Conference on Semantic Systems, pp. 33-40. ACM, 2017.
[34] Papadakis, George, Ekaterini Ioannou, Claudia Niederée, Themis Palpanas, and Wolfgang Nejdl. "Beyond 100 million entities: large-scale blocking-based resolution for heterogeneous data." In Proceedings of the fifth ACM international conference on Web search and data mining, pp. 53-62. ACM, 2012.
[35] Papadakis, George, George Papastefanatos, Themis Palpanas, and Manolis Koubarakis. "Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking." In EDBT, pp. 221-232. 2016.
[36] Fisher, Jeffrey, Peter Christen, Qing Wang, and Erhard Rahm. "A clustering-based framework to control block sizes for entity resolution." In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 279-288. ACM, 2015.
[37] Papadakis, George, Ekaterini Ioannou, Claudia Niederée, Themis Palpanas, and Wolfgang Nejdl. "Eliminating the redundancy in blocking-based entity resolution methods." In Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries, pp. 85-94. ACM, 2011.
[38] Whang, Steven Euijong, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina. "Entity resolution with iterative blocking." In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 219-232. ACM, 2009.
[39] Kolb, Lars, Andreas Thor, and Erhard Rahm. "Dedoop: Efficient deduplication with Hadoop." Proceedings of the VLDB Endowment 5, no. 12 (2012): 1878-1881.
[40] Efthymiou, Vasilis, George Papadakis, George Papastefanatos, Kostas Stefanidis, and Themis Palpanas. "Parallel meta-blocking: Realizing scalable entity resolution over large, heterogeneous data." In 2015 IEEE International Conference on Big Data (Big Data), pp. 411-420. IEEE, 2015.
[41] Efthymiou, Vasilis, George Papadakis, George Papastefanatos, Kostas Stefanidis, and Themis Palpanas. "Parallel meta-blocking for scaling entity resolution over big heterogeneous data." Information Systems 65 (2017): 137-157.
[42] KNUTH, DE. "The Art of Computer Programming 1: Fundamental Algorithms 2: Seminumerical Algorithms 3: Sorting and Searching. 1968." Cité en: 168.
[43] Levenshtein, Vladimir. "Binary codes capable of correcting spurious insertions and deletion of ones." Problems of Information Transmission 1, no. 1 (1965): 8-17.
[44] Sankoff, D., and J. B. Kruskal. "The theory and practice of sequence comparison." In Time Warps, String Edits, and Macromolecules. Addison-Wesley, 1983.
[45] Winkler, William E. "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." (1990).
[46] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv: 1301.3781 (2013).
[47] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.
[48] Kusner, Matt, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. "From word embeddings to document distances." In International Conference on Machine Learning, pp. 957-966. 2015.
[49] Sen, Prithviraj, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. "Collective classification in network data." AI Magazine 29, no. 3 (2008): 93-93.