SeMBlock: A semantic-aware meta-blocking approach for entity resolution

Intelligent Decision Technologies 15 (2021) 461–468
DOI 10.3233/IDT-200207
IOS Press

SeMBlock: A semantic-aware meta-blocking approach for entity resolution

Delaram Javdani[a], Hossein Rahmani[a,*] and Gerhard Weiss[b]
[a]*School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*
[b]*Department of Data Science and Knowledge Engineering, Maastricht University, Maastricht, The Netherlands*

Abstract. Entity resolution refers to the process of identifying, matching, and integrating records belonging to unique entities in a data set. However, a comprehensive comparison across all pairs of records leads to quadratic matching complexity. Therefore, blocking methods are used to group similar entities into small blocks before the matching. Available blocking methods typically do not consider semantic relationships among records. In this paper, we propose a Semantic-aware Meta-Blocking approach called SeMBlock. SeMBlock considers the semantic similarity of records by applying locality-sensitive hashing (LSH) based on word embedding to achieve fast and reliable blocking in a large-scale data environment. To improve the quality of the blocks created, SeMBlock builds a weighted graph of semantically similar records and prunes the graph edges. We extensively compare SeMBlock with 16 existing blocking methods, using three real-world data sets. The experimental results show that SeMBlock significantly outperforms all 16 methods with respect to two relevant measures, F-measure and pair-quality measure. F-measure and pair-quality measure of SeMBlock are approximately 7% and 27%, respectively, higher than recently released blocking methods.

Keywords: Data matching, entity resolution, meta-blocking, word embedding, locality-sensitive hashing, semantic similarity, big data integration

1. Introduction

Integrating two or more datasets in the absence of a unique identifier is a challenging problem that causes redundancy of data and inaccurate knowledge extraction [1–3]. Entity resolution is used to identify, match and integrate the records of an entity in different datasets [4–6]. However, there are challenges in entity resolution such as computing similarities between all pairs of records in a large dataset, which is problematic because the number of comparisons grows quadratically with the size of the dataset. Even for a small dataset, calculating the total similarity matrix using costly similarity functions can be extremely difficult [7].

To meet these challenges, blocking in entity resolution is used to group the records into a set of blocks so that a block contains only similar entities [8–11] and the entities in a block are more similar to each other than to entities in other blocks [12,13]. A number of blocking methods [14–28] have already been proposed which are exclusively based on the textual similarity of records while not taking semantic information into account.

In this paper, we propose a novel blocking method called SeMBlock (Semantic-aware Meta-Blocking approach) that uses a locality-sensitive hashing (LSH) method based on word embedding (BERT) to discover semantic relationships among pairs of records. Text analysis techniques [29–32] such as word embedding methods, Word2Vec [33,34] and BERT (Bidirectional Encoder Representations from Transformers) [35], map words to a new space in which semantically similar words are placed adjacent to each other. Word2Vec applies a neural network to describe each word in the text with a vector [33,34] and BERT pre-trains deep bidirectional representations from unlabeled text [35]. One of the major differences between BERT and Word2vec is that the BERT generates word embeddings based on

*Corresponding author: Hossein Rahmani, School of Computer Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran. E-mail: h_rahmani@iust.ac.ir.

ISSN 1872-4981/$35.00

the contexts in which the words appear. As a result, it generates different embeddings for each of the occurrences of a word, while in Word2Vec each instance of the word has the same representation even if it occurs in different contexts. Locality-sensitive hashing (LSH) is a popular method for nearest neighbor search in high-dimensional spaces. The basic idea is that the more similar two records are, the higher the probability is that they are hashed into the same block. LSH is a fast blocking technique over extensive data sets due to its probabilistic nature. Leveraging LSH techniques can reduce the time complexity of generating blocks to O(n). Additionally, the use of LSH techniques in semantic similarity space significantly improves the quality of blocking by removing record pairs that are textually similar but semantically different [12]. SeMBlock generates a graph of semantically similar records and applies a pruning method in order to improve the quality of blocking. The main contributions of SeMBlock are:

- Creating fast and reliable blocking in a large-scale data environment that considers the context of the text.
- Constructing a semantic-aware blocking graph.
- Outperforming 16 state-of-the-art blocking methods on three real-world data sets.

SeMBlock has been proposed to take advantage of word embedding for extracting semantic similarity and locality-sensitive hashing in order to significantly reduce the time complexity of generating blocks and to improve the run-time and accuracy of entity resolution. Accurate and fast entity resolution has huge practical implications in almost all modern data management tasks such as information extraction [36], big data analysis [37], and knowledge base construction [38] and helps businesses in unifying their customer data and in improving their decision making.

The structure of this article is as follows: Section 2 overviews available blocking methods. Section 3 provides a formal description of the entity resolution problem. SeMBlock is described in detail in Section 4. The experimental results are described and discussed in Section 5. Section 6 provides summarizing conclusions and future work.

2. Background

Entity Resolution is the process of identifying different entity records that represent the same real-world object. Performing entity resolution tasks over large data sets is computationally challenging due to the quadratic complexity $O(n^2)$ of pairwise comparisons, i.e., as every entity record has to be compared with all others. To reduce its computational cost, blocking techniques [10,39,40] have been proposed. Blocking efforts to identify which entity pairs are likely to match in order to limit comparisons only between them without knowledge of the matching function. Blocking leads to a time complexity $O(m^2 * |B|)$ for the size of the maximal block m and the number of blocks $|B|$ in the worst-case [10,12]. In the following, we discuss five categories of blocking approaches and point to state-of-the-art representatives of these categories.

Traditional schema-based blocking techniques such as standard blocking [18,41], sorted neighborhood [42], extended sorted neighborhood [19], q-grams blocking [43], extended q-grams blocking [19], MFIBlocks [44], canopy clustering [45,46], extended canopy clustering [19], suffix arrays [47] and extended suffix arrays [19] generate blocks based on the blocking keys, which depend on the schema of the dataset [48]. The main drawback of these methods is the choice of selecting features for the blocking keys, which is a laborious and error-prone process and requires domain expert knowledge [21,49].

In comparison with schema-based blocking techniques, schema-agnostic blocking approaches such as token blocking [50–52], attribute-clustering blocking [22], TYPiMatch [20] do not use any schema information [48]. They place each record in multiple blocks and create overlapping blocks which decrease the probability of a matching loss and increase the probability of inserting non-matching records in the same block (high recall but at the expense of low precision) [21].

Meta-blocking approaches such as WNP and CNP meta-blocking [14,24], BLAST [21], BLOSS [17], supervised meta-blocking [15] and multi-core meta-blocking [53] have been proposed which rebuild a set of blocks to keep the most promising comparisons [14]. To do so, the set of blocks are shown by a weighted graph, called a blocking graph [14]. In this graph, each record is represented by a node and an edge exists between two nodes if the corresponding records together appear in at least one block. The weights of the edges are calculated for the matching probability. Then, a pruning algorithm is applied based on the weights of the edges. Eventually, each pair of nodes connected by edge create a new block [24,54].

3. Problem definition

At the core of entity resolution lies the concept of the entity record, which forms a unique set of at-

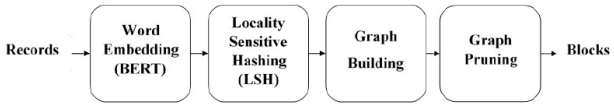

Fig. 1. The four main steps of SeMBlock.

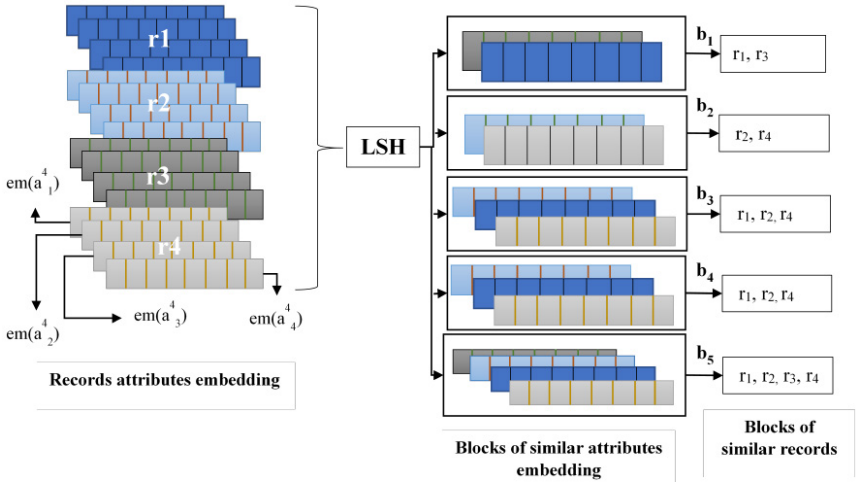

Fig. 2. An illustration of the semantic-aware LSH method.

tribute name-value pairs. An individual entity record is specified by r_i, where i stands for its unique identity in a record collection R ($R = \{r_1, \ldots, r_n\}$). Each record $r_i \in R$ contains a set of attributes denoted by $A_i = \{a_1^i, \ldots, a_m^i\}$. Two different records r_i and r_j ($\{r_i, r_j\} \subseteq R$ and $i \neq j$) are called duplicates if they represent the same real-world entity. Blocking is used for scaling entity resolution to extensive data collections. It groups similar records into a set of blocks B ($B = \{b_1, \ldots, b_{n'}\}$) so that for any three records $r_i \in b_u$, $r_j \in b_u$ and $r_k \in b_v$ with $u \neq v$, the probability that r_i and r_j refer to the same real-world entity is higher than the probability that r_i and r_k refer to the same real-world entity. The problem of blocking is to discover a grouping of a given set of records so that the mentioned criteria are fulfilled *and* the number of required record comparisons is kept as low as possible [48].

4. SeMBlock

SeMBlock comprises four sequential main steps, as overviewed in Fig. 1. These steps are as follows:

1. For each record $r_i \in R$, its attribute set A_i is considered and BERT [35] is applied to each attribute $a_{i'}^i \in A_i$. As a result, each record r_i with m attributes $\{a_1^i, \ldots, a_m^i\}$ is represented by m BERT embedding $(em(a_1^i), \ldots, em(a_m^i))$ where
$$em(a_{i'}^i) = BERT(a_{i'}^i) \quad (1)$$
2. To avoid the quadratic complexity of calculating the pairwise similarity of attributes' embedding, SeMBlock applies Locality-Sensitive Hashing (LSH), which hashes similar attributes' embedding into the same block with high probability and creates a set of blocks $B = \{b_1, \ldots, b_{n'}\}$. SeMBlock uses LSH for the angular distance proposed by Andoni et al. [55] to calculate cosine similarity between attributes' embedding of records in record collection R and to put similar attributes' embedding into the same block b_l. This is expressed by
$$B = LSH(EM) \quad (2)$$
where $EM = \{em(a_1^1), \ldots, em(a_m^n)\}$ and n is the number of records, m is the number of attributes and the size of EM is equal to n multiplied by m.

Example 1. For the records r1-r4 in Fig. 2, which each contains 4 attributes (e.g., r4 contains a_1^4, a_2^4, a_3^4 and a_4^4), attribute embeddings are obtained using BERT. Then, all attribute embeddings extracted (EM = $\{em(a_1^1), em(a_2^1), em(a_3^1), em(a_4^1), em(a_1^2), em(a_2^2), em(a_3^2), em(a_4^2), em(a_1^3),$

Table 1
Basic statistical information on the three chosen datasets

Datasets	Usage	Number of records	Number of attributes	Attributes	Number of matches
DBLP-ACM	Record linkage	2,616 + 2,294	4	Title, authors, venue and year	2,220
DBLP-Scholar	Record linkage	2,616 + 64,263	4	Title, authors, venue and year	5,347
Cora	Deduplication	1,295	12	Author, volume, title, institution, venue, address, publisher, year, pages, editor, note and month	17,184

$em(a_2^3), em(a_3^3), em(a_4^3), em(a_1^4), em(a_2^4), em(a_3^4)$ and $em(a_4^4)\}$) are given to the LSH in order to calculate cosine similarity among them and put similar embeddings into the same blocks. As shown in Fig. 2, LSH constructs 5 blocks, each containing similar attribute embeddings. Finally for each block, similar records are obtained (e.g., if $em(a_1^1)$ and $em(a_1^3) \in b_1$ then r_1 and $r_3 \in b_1$).
After building the set of blocks B, SeMBlock calculates the similarity between two records r_i and r_j as follows:

$$Sim(r_i, r_j) = |B_{r_i} \cap B_{r_j}| \quad (3)$$

where B_{r_i} and B_{r_j} are the set of blocks associated with r_i and r_j, respectively, and $|B_{r_i} \cap B_{r_j}|$ corresponds to the number of blocks r_i and r_j have in common.

Example 2. For the records r_1 and r_2 in Fig. 2, $B_{r_1} = \{b_1, b_3, b_4, b_5\}$ and $B_{r_2} = \{b_2, b_3, b_4, b_5\}$. Therefore, $|B_{r_1} \cap B_{r_2}| = |\{b_3, b_4, b_5\}| = 3$.

3. SeMBlock constructs a blocking graph in which each $r_i \in R$ is represented by a node and an edge e_{ij} exists between two nodes r_i and r_j if the corresponding records r_i and r_j are in the same block (which is the case if $|B_{r_i} \cap B_{r_j}| \neq 0$). The weight of the edge w_{ij} is equal to the similarity of two records r_i and r_j:

$$w_{ij} = Sim(r_i, r_j) \quad (4)$$

4. SeMBlock prunes the graph in order to further increase accuracy. If the weight of an edge e_{ij} (i.e., w_{ij}) is higher than some predefined integer value α, the edge will be kept; otherwise, it will be removed:

$$\text{Pruning}(e_{ij}) = \begin{cases} \text{keep}(e_{ij}), & \text{if } w_{ij} > \alpha \\ \text{remove}(e_{ij}), & \text{otherwise} \end{cases} \quad (5)$$

5. Experimental results

This section is divided into three subsections, describing the three real-world data sets used for evaluation, the blocking evaluation measures used to evaluate SeMBlock, and the performance of SeMBlock in comparison to the 16 blocking methods.

5.1. Data sets

We use three real-world data sets: Cora [56], DBLP-ACM [57] and DBLP-Scholar [57]: Cora contains bibliographic records of machine learning papers, DBLP-ACM contains bibliographic data from DBLP and ACM, and DBLP-Scholar contains bibliographic data from DBLP and Google Scholar. Table 1 shows details of each data set. The experimental results of our baseline models are only available on these three datasets, so we use these three datasets for a fair comparison.

5.2. Evaluation measures

We use three common measures [9,10,18,58,59] to evaluate blocking quality: Pair Completeness (*PC*), Pairs Quality (*PQ*), and F-Measure (*FM*).

PC, which is also known as recall, estimates the portion of the duplicate entities that co-occur at least once in a set of blocks B [18]:

$$PC = \frac{D_B}{D_R} \quad (6)$$

where D_B is the number of duplicates appearing in B and D_R is the number of all duplicates in the record collection R.

PQ, also known as precision, measures the portion of comparisons that correspond to real duplicates (see, e.g., [18]) and is given by

$$PQ = \frac{D_B}{||B||} \quad (7)$$

where $||B|| = \sum_{b_i \in B} ||b_i||$ and $||b_i||$ is the number of comparisons implied by the block b_i.

Finally, FM is the harmonic mean of PC and PQ (see, e.g., [9]):

$$FM = \frac{2 * PC * PQ}{PC + PQ} \quad (8)$$

5.3. SeMBlock vs. Existing blocking methods

We evaluated SeMBlock with the datasets and measures described above. According to SeMBlock, for each dataset the BERT embeddings of the records are calculated, LHS is applied, a blocking graph is constructed, and graph pruning is executed. As described in

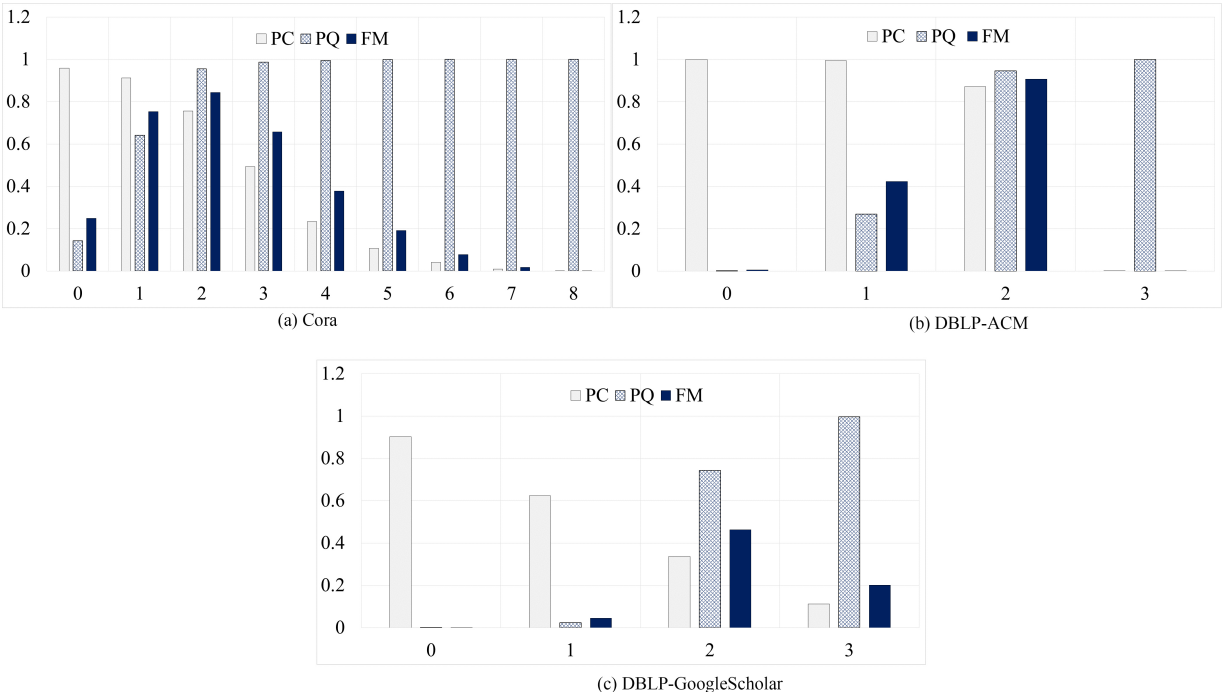

Fig. 3. PC, PQ, and FM for SeMBlock and different values of α. For these datasets the best pruning threshold is $\alpha = 2$.

Section 4, pruning the parameter α is crucial. Figure 3 shows the impact of this parameter on the three measures PQ, PC, and FM. Specifically, this figure shows for the three datasets how these measures vary for different values of α (note that $\alpha = 0$ means that no pruning happens). As can be seen from Fig. 3, $\alpha = 2$ is the best choice for graph pruning with respect to FM for all three datasets. As α increases, the PQ of SeMBlock increases but PC decreases dramatically, and accordingly, FM decreases.

We compared SeMBlock with the following standard blocking approaches (The parameters of each method are determined based on the best values specified for them in the associated papers):

- Schema-based approaches: Sorted neighborhood blocking (SN) [42], Extended sorted neighborhood blocking (ESN) [19], Canopy clustering (CC) [45,46], Extended canopy clustering (ECC) [19], Suffix arrays blocking (SA) [47], Extended suffix arrays blocking (ESA) [19], Q-grams blocking (Qg) [43], and Extended q-grams blocking (EQg) [19].
- Schema-agnostic approaches: Token blocking (TB) [50], Attribute clustering blocking (AC) [22], and Unsupervised blocking technique (BL) [60].
- Meta-blocking approaches: BLAST [21], Redefined WNP (Wnp1) [24], Reciprocal WNP (Wnp2) [24], Redefined CNP (Cnp1) [24], and Reciprocal CNP (Cnp2) [24].

We calculated the PC, PQ, and FM measures for these 16 methods in the three datasets. The results of this comparison are shown in Table 2 (where SeMBlock was executed with $\alpha = 2$). As can be seen from this table, SeMBlock outperformed all other methods with respect to FM and PQ. However, the PC values of SeMBlock are not the highest ones (though they are not significantly below the highest PC values) over the three datasets. This is not a serious drawback because for cases where the PC measure is more important, changing the α value (e.g., $\alpha = 1$) can increase this measure.

Additionally, we evaluated the efficiency of the blocking methods by the number of record pairs each method generates, as blocking aims to reduce the number of pairs to be compared in entity resolution. Table 2 shows the number of record pairs (i.e., $||B||$) generated by different blocking methods. As it is shown in Table 2, the number of record pairs of SeMBlock is the lowest in all three datasets.

Moreover, we compared SeMBlock with recently released blocking methods (Rebo-I and Rebo-II) [61] in the Cora dataset. The results showed that SeMBlock outperformed both methods with respect to FM and PQ (i.e., Rebo-I (PC: 0.928, PQ: 0.694 and FM: 0.794),

Table 2
PC, PQ, and FM for SeMBlock and 16 other blocking methods in the Cora, DBLP-ACM, and DBLP-Scholar datasets and comparison on the number of record pairs generated by different approaches

	DBLP-Scholar				DBLP-ACM				Cora			
Methods	*PC*	*PQ*	*FM*	$\|\|B\|\|$	*PC*	*PQ*	*FM*	$\|\|B\|\|$	*PC*	*PQ*	*FM*	$\|\|B\|\|$
TB	0.43	0.00	0.00	$9.10*10^7$	1.00	0.00	0.00	$6.61*10^6$	1.00	0.00	0.01	$4.84*10^6$
AC	0.42	0.00	0.00	$8.99*10^7$	1.00	0.00	0.00	$6.42*10^6$	1.00	0.00	0.01	$4.68*10^6$
BL	0.99	0.00	0.00	$6.90*10^6$	0.99	0.04	0.07	$6.16*10^4$	0.86	0.38	0.53	$3.84*10^4$
SN	0.00	0.00	0.00	$4.33*10^5$	0.96	0.01	0.02	$2.43*10^5$	0.65	0.06	0.10	$2.01*10^5$
ESN	0.44	0.00	0.00	$1.84*10^8$	1.00	0.00	0.00	$1.47*10^7$	1.00	0.00	0.00	$1.02*10^7$
CC	0.00	0.00	0.00	$3.39*10^3$	0.98	0.84	**0.90**	**$2.58*10^3$**	0.97	0.02	0.04	$8.78*10^5$
ECC	0.00	0.00	0.00	$1.10*10^5$	0.05	0.01	0.02	$9.51*10^3$	0.36	0.36	0.36	$1.70*10^4$
SA	0.00	0.00	0.00	$3.91*10^5$	1.00	0.01	0.01	$3.95*10^5$	0.40	0.09	0.15	$7.41*10^4$
ESA	0.00	0.00	0.00	$5.34*10^5$	1.00	0.00	0.01	$6.42*10^5$	0.26	0.06	0.10	$7.34*10^4$
Qg	0.46	0.00	0.00	$1.41*10^8$	1.00	0.00	0.00	$1.26*10^7$	1.00	0.00	0.00	$8.82*10^6$
EQg	0.44	0.00	0.00	$1.23*10^8$	1.00	0.00	0.00	$1.20*10^7$	1.00	0.00	0.00	$9.30*10^6$
BLAST	0.95	0.05	0.10	$4.10*10^4$	0.99	0.61	0.75	$3.60*10^3$	0.82	0.84	**0.83**	**$1.68*10^4$**
Wnp1	0.98	0.01	0.02	$2.90*10^5$	0.99	0.14	0.24	$1.70*10^4$	0.90	0.54	0.68	$2.86*10^4$
Wnp2	0.95	0.03	0.07	$6.30*10^4$	0.98	0.24	0.38	$9.20*10^3$	0.81	0.69	0.75	$2.01*10^4$
Cnp1	0.94	0.02	0.04	$1.10*10^5$	0.99	0.10	0.18	$2.20*10^4$	0.67	0.66	0.66	$1.74*10^4$
Cnp2	0.88	0.31	**0.46**	$6.50*10^4$	0.98	0.20	0.34	$1.10*10^4$	0.46	0.82	0.59	$9.64*10^3$
SeMBlock	0.33	**0.78**	**0.46**	**$2.26*10^3$**	0.87	**0.95**	**0.91**	**$2.03*10^3$**	0.76	**0.96**	**0.85**	**$1.36*10^4$**

Rebo-II (PC: 0.935, PQ: 0.656 and FM: 0.771), and SeMBlock (PC: 0.76, PQ: 0.96 and FM: 0.85)).

6. Conclusions and future work

Entity resolution is the process of identifying records in a data set that refer to the same entity across different data sources. To avoid the quadratic complexity of entity resolution, many attempts have been made to group similar records into blocks, prior to record matching, using blocking techniques. Available blocking methods typically do not exploit semantic criteria for the task of blocking. We introduced a semantic-aware Meta-Blocking approach called SeMBlock that exploits word-embedding based locality-sensitive hashing (LSH) for calculating semantic similarity and identifying relationships among records. The experimental results show that considering semantic relationships in the blocking process can significantly improve the quality of blocking. The size of the blocks generally gets smaller because semantic features can effectively eliminate record pairs that are textually similar but semantically different. We also compared SeMBlock with 16 available standard blocking methods. Overall, SeMBlock can be an effective blocking technique compared to these methods when the priority is to achieve high pair-quality and f-measure, without giving up a high level of pair-completeness.

In the future, first, we plan to extend SeMBlock by leveraging context information within a network environment for enhancing the applicability of SeMBlock to real-world ER problems. Second, we want to investigate the effect of combining several semantic features on the quality of blocking. Third, we intend to find an alternative to the LSH method that increases the accuracy of our method.

References

[1] Bhattacharya I, Getoor L. Collective entity resolution in relational data. ACM Transactions on Knowledge Discovery from Data (TKDD). 2007; 1(1): 5.

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

[3] Lin Y, Wang H, Li J, Gao H. Efficient entity resolution on heterogeneous records. IEEE Transactions on Knowledge and Data Engineering, 2019.

[4] Tauer G, Date K, Nagi R, Sudit M. An incremental graph-partitioning algorithm for entity resolution. Information Fusion. 2019; 46: 171–183.

[5] Christophides V, Efthymiou V, Palpanas T, Papadakis G, Stefanidis K. End-to-End Entity Resolution for Big Data: A Survey. arXiv preprint arXiv190506397, 2019.

[6] Kwashie S, Liu L, Liu J, Stumptner M, Li J, Yang L. Certus: an effective entity resolution approach with graph differential dependencies (GDDs). Proceedings of the VLDB Endowment. 2019; 12(6): 653-666.

[7] Bilenko M, Kamath B, Mooney RJ. Adaptive blocking: Learning to scale up record linkage. In: Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 87–96.

[8] Papadakis G, Tsekouras L, Thanos E, Pittaras N, Simonini G, Skoutas D, et al. JedAI3: beyond batch, blocking-based Entity Resolution. In: EDBT, 2020, pp. 603–606.

[9] Wang Q, Cui M, Liang H. Semantic-aware blocking for entity resolution. IEEE Transactions on Knowledge and Data Engineering. 2016; 28(1): 166--180.

[10] Papadakis G, Skoutas D, Thanos E, Palpanas T. A Survey of Blocking and Filtering Techniques for Entity Resolution. arXiv preprint arXiv190506167, 2019.

[11] Araújo TB, Pires CES, Mestre DG, Nóbrega TPD, Nascimento DCD, Stefanidis K. A noise tolerant and schema-agnostic blocking technique for entity resolution. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. ACM, 2019, pp. 422–430.

[12] Wang Q, Cui M, Liang H. Semantic-aware blocking for entity resolution. IEEE Transactions on Knowledge and Data Engineering. 2015; 28(1): 166-180.

[13] De Vries T, Ke H, Chawla S, Christen P. Robust record linkage blocking using suffix arrays and Bloom filters. ACM Transactions on Knowledge Discovery from Data (TKDD). 2011; 5(2): 9.

[14] Papadakis G, Koutrika G, Palpanas T, Nejdl W. Meta-blocking: Taking entity resolutionto the next level. IEEE Transactions on Knowledge and Data Engineering. 2014; 26(8): 1946–1960.

[15] Papadakis G, Papastefanatos G, Koutrika G. Supervised meta-blocking. Proceedings of the VLDB Endowment. 2014; 7(14): 1929-1940.

[16] Araújo TB, et al., Parallel blocking for entity resolution in the context of semi-structured data, 2020.

[17] Dal Bianco G, Gonçalves MA, Duarte D. BLOSS: Effective meta-blocking with almost no effort. Information Systems. 2018; 75: 75-89.

[18] Papadakis G, Svirsky J, Gal A, Palpanas T. Comparative analysis of approximate blocking techniques for entity resolution. Proceedings of the VLDB Endowment. 2016; 9(9): 684-695.

[19] Christen P. A survey of indexing techniques for scalable record linkage and deduplication. IEEE Transactions on Knowledge and Data Engineering. 2012; 24(9): 1537-1555.

[20] Ma Y, Tran T. 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. ACM, 2013; pp. 325–334.

[21] Simonini G, Bergamaschi S, Jagadish H. BLAST: a loosely schema-aware meta-blocking approach for entity resolution. Proceedings of the VLDB Endowment. 2016; 9(12): 1173-1184.

[22] Papadakis G, Ioannou E, Palpanas T, Niederee C, Nejdl W. A blocking framework for entity resolution in highly heterogeneous information spaces. IEEE Transactions on Knowledge and Data Engineering. 2013; 25(12): 2665-2682.

[23] Fisher J, Christen P, Wang Q, Rahm E. A clustering-based framework to control block sizes for entity resolution. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 279–288.

[24] Papadakis G, Papastefanatos G, Palpanas T, Koubarakis M. Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking. In: EDBT, 2016, pp. 221–232.

[25] Whang SE, Menestrina D, Koutrika G, Theobald M, Garcia-Molina H. Entity resolution with iterative blocking. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. ACM, 2009, pp. 219–232.

[26] Efthymiou V, Stefanidis K, Christophides V. Benchmarking blocking algorithms for web entities. IEEE Transactions on Big Data, 2016.

[27] Efthymiou V, Papadakis G, Papastefanatos G, Stefanidis K, Palpanas T. Parallel meta-blocking for scaling entity resolution over big heterogeneous data. Information Systems. 2017; 65: 137-157.

[28] Efthymiou V, Papadakis G, Papastefanatos G, Stefanidis K, Palpanas T. Parallel meta-blocking: Realizing scalable entity resolution over large, heterogeneous data. In: 2015 IEEE International Conference on Big Data (Big Data). IEEE, 2015, pp. 411–420.

[29] Piryani R, Gupta V, Singh VK. Generating aspect-based extractive opinion summary: Drawing inferences from social media texts. Computación y Sistemas. 2018; 22(1): 83-91.

[30] Gupta V, Singh VK, Mukhija P, Ghose U. Aspect-based sentiment analysis of mobile reviews. Journal of Intelligent & Fuzzy Systems. 2019; 36(5): 4721-4730.

[31] Piryani R, Gupta V, Singh VK, Ghose U. A linguistic rule-based approach for aspect-level sentiment analysis of movie reviews. In: Advances in Computer and Computational Sciences. Springer, 2017, pp. 201–209.

[32] Allahgholi M, Rahmani H, Javdani D, Weiss G, Módos D. ADDI: Recommending alternatives for drug–drug interactions with negative health effects. Computers in Biology and Medicine. 2020; 125: 103969.

[33] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv13013781, 2013.

[34] Ma L, Zhang Y. Using Word2Vec to process big text data. In: 2015 IEEE International Conference on Big Data (Big Data). IEEE, 2015, pp. 2895–2897.

[35] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv181004805, 2018.

[36] Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, et al. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 601–610.

[37] Jagadish HV, Gehrke J, Labrinidis A, Papakonstantinou Y, Patel JM, Ramakrishnan R, et al., Big data and its technical challenges. Communications of the ACM. 2014; 57(7): 86-94.

[38] De Sa C, Ratner A, Ré C, Shin J, Wang F, Wu S, et al., Incremental knowledge base construction using DeepDive. The VLDB Journal. 2017; 26(1): 81--105.

[39] Papadakis G, Skoutas D, Thanos E, Palpanas T. Blocking and Filtering Techniques for Entity Resolution: A Survey. ACM Comput Surv. 2020 Mar; 53(2). Available from: 10.1145/3377455.

[40] Vidhya K, Geetha T. Entity Resolution and Blocking: A Review. In: 2019 IEEE 9th International Conference on Advanced Computing (IACC). IEEE, 2019, pp. 133–140.

[41] Fellegi IP, Sunter AB. A theory for record linkage. Journal of the American Statistical Association. 1969; 64(328): 1183--1210.

[42] Hernández MA, Stolfo SJ. The merge/purge problem for large databases. In: ACM Sigmod Record. vol. 24. ACM, 1995, pp. 127–138.

[43] Gravano L, Ipeirotis PG, Jagadish HV, Koudas N, Muthukrishnan S, Srivastava D, et al., Approximate string joins in a database (almost) for free. In: VLDB. vol. 1, 2001, pp. 491–500.

[44] Kenig B, Gal A. MFIBlocks: An effective blocking algorithm for entity resolution. Information Systems. 2013; 38(6): 908--926.

[45] McCallum A, Nigam K, Ungar LH. 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. Citeseer, 2000, pp. 169–178.

[46] Baxter LR, Baxter R, Christen P, et al., A comparison of fast blocking methods for record, 2003.
[47] Aizawa A, Oyama K. A fast linkage detection scheme for multi-source information integration. In: International Workshop on Challenges in Web Information Retrieval and Integration. IEEE, 2005, pp. 30–39.
[48] Simonini G, Papadakis G, Palpanas T, Bergamaschi S. Schema-agnostic progressive entity resolution. IEEE Transactions on Knowledge and Data Engineering. 2018; 31(6): 1208--1221.
[49] Rahmani H, Ranjbar-Sahraei B, Weiss G, Tuyls K. Entity resolution in disjoint graphs: an application on genealogical data. Intelligent Data Analysis. 2016; 20(2): 455--475.
[50] Papadakis G, Ioannou E, Niederée C, Fankhauser P. Efficient entity resolution for large heterogeneous information spaces. In: Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011, pp. 535–544.
[51] Efthymiou V, Papadakis G, Stefanidis K, Christophides V. MinoanER: Schema-Agnostic, Non-Iterative, Massively Parallel Resolution of Web Entities. arXiv preprint arXiv190506170, 2019.
[52] Gagliardelli L, Simonini G, Beneventano D, Bergamaschi S. SparkER: Scaling Entity Resolution in Spark. In: EDBT 2019: 22nd International Conference on Extending Database Technology, 2019.
[53] Papadakis G, Bereta K, Palpanas T, Koubarakis M. Multi-core meta-blocking for big linked data. In: Proceedings of the 13th International Conference on Semantic Systems. ACM, 2017, pp. 33–40.
[54] Simonini G, Papadakis G, Palpanas T, Bergamaschi S. Schema-agnostic Progressive Entity Resolution (extended version). arXiv preprint arXiv190506385, 2019.
[55] Andoni A, Indyk P, Laarhoven T, Razenshteyn I, Schmidt L. Practical and optimal LSH for angular distance. In: Advances in Neural Information Processing Systems, 2015, pp. 1225–1233.
[56] McCallum A. Cora Dataset. Texas Data Repository Dataverse, 2017. Available from 10.18738/T8/HUIG48.
[57] Mudgal S, Li H, Rekatsinas T, Doan A, Park Y, Krishnan G, et al., Deep learning for entity matching: A design space exploration. In: Proceedings of the 2018 International Conference on Management of Data. ACM, 2018, pp. 19–34.
[58] Shao J, Wang Q. Active Blocking Scheme Learning for Entity Resolution. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2018, pp. 350–362.
[59] Shao J, Wang Q, Lin Y. Skyblocking: Learning Blocking Schemes on the Skyline. arXiv preprint arXiv180512319, 2018.
[60] O'Hare K, Jurek-Loughrey A, de Campos C. An unsupervised blocking technique for more efficient record linkage. Data & Knowledge Engineering. 2019; 122: 181–195.
[61] Yu SQ. Entity Resolution with Recursive Blocking. Big Data Research, 2020, p. 100134.