Soft Computing
https://doi.org/10.1007/s00500-023-09623-6

FOCUS

A two-stage entity event deduplication method based on graph node selection and node optimization strategy

Wei Ai[1] · Jia Xu[1] · Hongen Shao[1] · Tao Meng[1] · Keqin Li[2]

Accepted: 28 December 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract
Entity event deduplication is the task of identifying all duplication entity events that have described the same entity within a set of events. However, the traditional entity event deduplication method has two challenges. First, the traditional method usually used global comparison when finding the duplication entity event, are all entity events in the dataset need to be compared, leading to low performance. Second, when the entity event evolves, the traditional method does not identify it well and reduces the effectiveness. To address these two problems and improve the performance and effectiveness, we propose a two-stage deduplication method based on graph node selection and optimization *(TS-NSNO)* strategy. In the first stage *(TS-NS)*, we propose a graph node selection strategy, which transforms the global comparison into a local comparison by selecting the leader node, greatly reduces the number of calculations and improves the performance. In the second stage *(TS-NO)*, we propose a graph node optimization strategy, by combining the spatiotemporal distance and entity event importance change of the event evolution, which optimizes the entity event with incorrect judgment to improve the effectiveness. We conduct extensive experiments on real entity event datasets of different sizes, and the results show that our method performs better in terms of performance and effectiveness.

Keywords Deduplication · Entity event · Event evolution · Entity event connected subgraph

1 Introduction

1.1 Motivation

With the exponential growth of news media, huge numbers of text documents exist, which often report similar information about the same entity. Therefore, in text document analysis, there is a large amount of duplication in event information extracted by the event extraction technology (Liu et al. 2016; Jadhav and Rajan 2018; Han et al. 2018). Given these large amounts of similar event information, event deduplication becomes an important operation. In general, entity event deduplication is the task of identifying all duplication entity events that describe the same entity within an event set, for example,"Due to the illegal use of personal information, Chinese regulators ordered the deletion of another 25 Didi-related applications", and "Administration of China ordered the Didi app removed from mobile app stores in China due to its illegal collection of customer data". Both of these examples are related to news of the Didi app being removed from a platform, and although one is from the perspective of users and the other is from the perspective of the government, they are still the repeated entity events. Repetitive information wastes time for decision-makers and may affect some decisions. Therefore, how to accurately and efficiently distinguishing entity events, which have different expressions but the same meaning, is an urgent problem to solve.

✉ Tao Meng
mengtao@hnu.edu.cn

Wei Ai
aiwei@hnu.edu.cn

Jia Xu
jiaxu.hn@foxmail.com

Hongen Shao
hongen.shao@foxmail.com

Keqin Li
lik@newpaltz.edu

[1] School of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410082, Hunan, China

[2] Department of Computer Science, State University of New York, New York 12561, New Paltz, USA

Published online: 07 February 2024

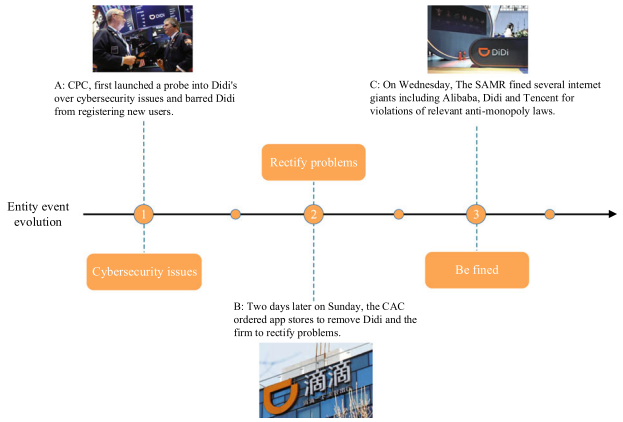


Fig. 1 Entity event evolution

In recent years, some researchers have described a few deduplication methods of entity events. The initial entity event method was based on sentence representation (McConky et al. 2012; Sharapova and Sharapov 2019), which judged whether the entity events were repeated mostly by calculating the similarity between the description sentences of the entity events. However, this method was too simplistic and could not express the complete meaning of entity events, so the deduplicaton effect of this method was poor. Thus, other more accurate expression methods were proposed that, based on attributes, could improve the expression of entity events (UzZaman and Allen 2010; Fedoryszak et al. 2019), and they work better than the method based on sentences. However, these methods do not consider the relationship between entity events. Therefore, another approach is graph-based representation (Chen 2010; Schinas et al. 2015; Liu et al. 2018). In this method, the relation between entity events is added, which allows the deduplication method effectiveness to reach a new level. However, the method involves calculating the similarity between graphs, which involves high complexity and cannot fulfill real-time deduplication requirements, and it does not consider entity event evolution. Entity event evolution is shown in Fig. 1, and in this series of events, the first two events are repeated for the third event. However, there is no corresponding event evolution feature of judgment, which makes it impossible to judge repetition. To balance and optimize the performance and effectiveness, in this paper, we combine the entity event evolution of sentences and attributes and construct a two-stage deduplication method based on graph node selection and optimization strategy.

1.2 Our contributions

In response to the above problems, this paper describes a two-stage entity event deduplication method based on graph node selection and optimization strategy, which can effectively improve performance and effectiveness. The first stage is the graph node selection strategy, which aims to improve the deduplication performance and realizes real-time deduplication detection. In this stage, we first construct entity event connected subgraph from the historical data and use the *node clustering coefficient* to select the leader node set of each subgraph. By selecting the leader node set, the computation can be greatly reduced, and high performance can be achieved. Second, we consider the relationship between the attributes of the entity event to modeling and improve the effectiveness of the deduplication method. The second stage is the graph node optimization strategy, whose main purpose is to improve the effectiveness of the deduplication method. In this stage, event evolution factors are considered for modeling and to improve the effectiveness (Yang et al. 2009). Our contributions to this paper are summarized as follows:

- We construct a node selection strategy that adopts the *node clustering coefficient* to select the leader node to reduce the complexity of the deduplication method and achieve the purpose of real-time deduplication detection.
- We propose a node optimization strategy, which models through event evolution to comprehensively improve the effectiveness of the deduplication method.
- We design a two-stage deduplication method, which aims to find duplication entity events. The experimental results show that this method can quickly and accurately obtain entity information, and provide nonrepeat information for follow-up tasks and customers.

The rest of this paper is organized as follows. In Sect. 2, we review the related works of entity event deduplication. Then, we introduce the definition and problems related to this paper in Sect. 3. In Sect. 4, we will describe the deduplication method in detail. Section 5 describes how to create a dataset, select and experimentally analyze and display relevant experimental environment configuration parameters. Finally, the conclusion of our contributions and a discussion of future work are presented in Sect. 6.

2 Related work

In terms of granularity, text deduplication techniques can be divided into two types: full-text deduplication techniques and event deduplication techniques. Full-text deduplication techniques are specifically used for near-duplicate detection, which is a coarse-grained deduplication technique. A common approach of this form is fingerprinting (Broder 1997; Charikar 2002) and hash techniques (Manku et al. 2007; Arun and Sumesh 2015). Compared with text deduplication, event deduplication is a more fine-grained text information duplication detection method. It better satisfies the current

demand for refined information. This paper is based on the perspective of events to study deduplication.

In recent years, many researchers have proposed related research methods for event deduplication. These methods are usually divided into three categories by means of representation of the event: sentence-based representation (McConky et al. 2012; Sharapova and Sharapov 2019; Liu et al. 2016), attribute-based representation (UzZaman and Allen 2010; Fedoryszak et al. 2019; Tomadaki and Salway 2005), and graph-based representation (Chen 2010; Schinas et al. 2015).

Sentence-based: This type of method is mainly used to calculate the distance between the description sentences of the event (e.g., edit distance, semantic distance, etc.) and obtain the similarity. McConky et al. proposed two kinds of event deduplication algorithms based on sentences (McConky et al. 2012). The first method extracted word combinations of specific parts of speech from event description statements by rules and then judged whether the event was repeated based on word combinations. The second method combined the semantic role of keywords for deduplication events based on the first method. The method based on sentence representation is simple in logic and does not consider the complex event attributes, so the effectiveness of deduplication is general.

Attribute-based: This kind of method calculates the attributes of the events that have been extracted to obtain the similarity between the attributes. Uzzaman et al. first defined attribute templates for different types of events, then populated event attributes through extraction methods, and finally deduplicated events through similarity calculation of event attributes (UzZaman and Allen 2010). Tomadaki et al. took the weight of different attributes into account for an event deduplication (Tomadaki and Salway 2005). Compared with the method based on sentence representation, the method based on attribute representation has better deduplication effectiveness, but the method relies on the extraction of event attributes, which has a high complexity and low deduplication efficiency. In addition, the expression form is simple, without considering the relationship between attributes.

Graph-based: In this method, the attributes of events are represented by nodes, the relationship between attributes is represented by the weight of edges, and then the event is deduplicated by calculating the similarity between graphs. Wang et al. described a method based on graph representation, constructed the graph representation of events through the proposed attribute similarity calculation equation, and then completed event deduplication through the calculation of graph similarity (Schinas et al. 2015). Based on the graph representation, Liu et al. considered the weight of different attributes and introduce *GCN* when calculating similar attributes to improve the effectiveness (Liu et al. 2018). This method has a better effectiveness, but it has higher complexity and poor deduplication efficiency, which cannot meet the requirements of real-time deduplication. At the same time, it is extremely time-consuming to compare each of the events, leading to a prohibitive $O(n^2)$ time complexity, where n is the number of documents.

Therefore, we propose a two-stage deduplication method based on graph node selection and optimization strategy. Our method uses graphs to gather similar events together and strategies to reduce the amount of calculation to improve the performance of deduplication. We use event evolution modeling to improve the effectiveness of event deduplication. Since event extraction is a mature subtask in the field of *NLP* and the main research object of this paper is deduplication, it will not be described extensively (Han et al. 2018; Hossny et al. 2020; Zhang et al. 2011).

3 Preliminaries

In this section, we put forward some necessary prerequisite knowledge to correctly understand our proposed method, because some of the concepts used are recent and meaningful. We define the entity event, describe the concept of the entity event-connected subgraph, and clarify why the above concept is proposed.

3.1 Entity event

An event is a specific action or measurement that occurs at a specific time and place, and it is defined as a five-tuple, which includes trigger words, subjects, objects, times, and locations (Zhang et al. 2011). Entities are concepts that have an identifier, and they are composed of a set of attributes and relationships to other concepts, such as the person names, organizations, locations, etc. From news reports and the definition of events, an event not only revolves around one entity but can also describe multiple entities at the same time. For example, in Fig. 1, event C describes multiple entities. When performing event deduplication for multiple entity events, it is impossible to accurately provide messages to the entity, resulting in the delay or omission of entity messages.

Therefore, this paper raises a definition for the entity event based on the need for precise deduplication. Through the entity event, we can obtain the relevant events of the entity more comprehensively (Ai et al. 2021).

Definition 3.1 (Entity Event) Occurring at a particular time or place with one role as the primary entity and the remaining entities in participating roles, called an entity event. Consisting of one or more actions and representing a change in action or state. If the event describes multiple entities, there is a separate agent event for each entity. An entity event is represented by the symbol E, and the specific expression as follows:

$$E = (W, S, C, O, T). \tag{1}$$

In the Eq. (1), W (word) represents the trigger word set, S (sentence) represents the set of sentences containing the trigger word, C (entity) represents an entity of an entity event, O (object) represents the set of objects participating in the entity event, and T (time) represents the time of the entity event. Example C of Fig. 1 contains three entity events.

It can be concluded from the definition of the entity event that an event can contain multiple entity events, and one entity event can only belong to one event. The entity event makes the information for each subject more complete. An entity event is also an event in nature, so this paper can use the event extraction method to extract it and then draw out all the entity events in the event according to Definition 3.1 (Han et al. 2018).

This paper adopts the method of combining sentence representation with the based on attributes representation and optimizes and adjusts the characteristics of financial news. How to extract the attributes of the entity event from the unstructured text is not the research content of this paper. Additionally, individual attributes can be omitted, but usually, a complete entity event representation, the three attributes of an entity, time and place, are all complete.

3.2 Entity event connected subgraph

Graph-based deduplication methods are similar to graph-based clustering. For example, there are some data points, our goal is to group the same data points into the same group, and the different data points into different groups. The graph-based processing method is the problem of converting data into a graph or modeling a real application as a graph, where vertices represent data points, and edges represent relationships between data points. This kind of graph has different forms, such as connected graphs, K-nearest neighbor graphs, and bipartite graphs (Chen 2010).

Most of the current duplication detection methods compare each pair of objects, leading to a prohibitive $O(n^2)$ time complexity (Zhang et al. 2016). Therefore, researchers have begun to optimize repeated detection methods, such as text document similarity graphs (Ge et al. 2019). The text documents in the graph are similar when there is an edge relationship and use the minimum vertex coverage to calculate to reduce the amount of calculation. However, the granularity of the text document similarity graph is large and cannot be used in entity event deduplication. Therefore, this paper proposes a concept of an entity event connected subgraph and optimizes the performance of the algorithm through node selection strategies (Ai et al. 2021).

Definition 3.2 (Entity Event Connected Subgraph) When repeated entity events of the same entity are connected as a connected subgraph, they are called entity event connected subgraphs. Among them, the node is the entity event, and the edge is the relationship between the entity events. The equation is expressed by

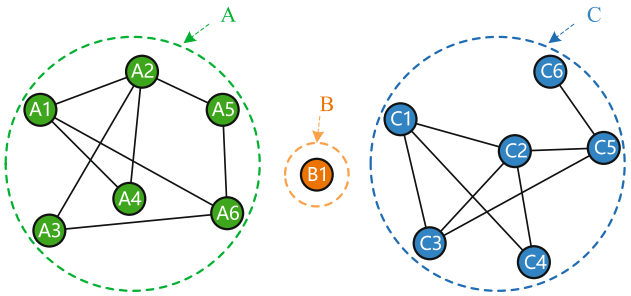


Fig. 2 Entity event connect subgraph

$$G^c = (V, A)\{E \in V \mid E(C) = c\}. \tag{2}$$

In Eq. (2), V represents the entity event set, the entity event is c, and A is the relationship between the entity events in this set. The constraint condition indicates that when entity event E belongs to connected subgraph G^c, the entity of the entity event is c. G^c represents a connected subgraph under current entity c.

According to Definitions 3.1 and 3.2, there are multiple connected subgraphs under entity c, as shown in Fig. 2. Therefore, we denote all connected subgraphs under entity c as a set D^c, whose expression is shown in Eq. (3) as follows:

$$D^c = \left\{G_1^c, G_2^c, \ldots, G_q^c\right\}(q = 1, 2, \ldots). \tag{3}$$

The entity event connected subgraph is shown in Fig. 2. There are three connected subgraphs of A, B, and C, and they all have the same entity c. The entity events in the subgraph are repeated, such as $A1$ and $A2$ being repeated, and $C1$ and $C2$ also like this. The entity events in different subgraphs are not repeated, such as $A1$ and $C1$. In general, when the entity events have an edge relationship or are in the same connected subgraph, they are repeated; otherwise they are not repeated.

4 The proposed method

In this section, we present the basic step for a two-stage deduplication method based on a graph, which is the construct graph and comparison of two arbitrary events. Specifically, we first outline our proposed framework to easily understand the overall process in Sect. 4.1. In Sects. 4.2 and 4.3, we describe the first stage and the second stage in detail. At the same time, we also explain why these methods are proposed and give a code description of the method.

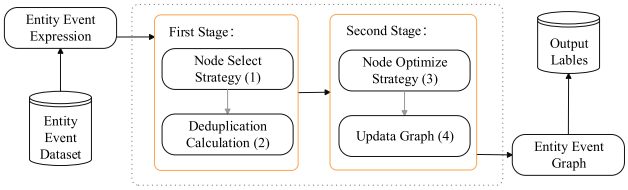


Fig. 3 Process framework

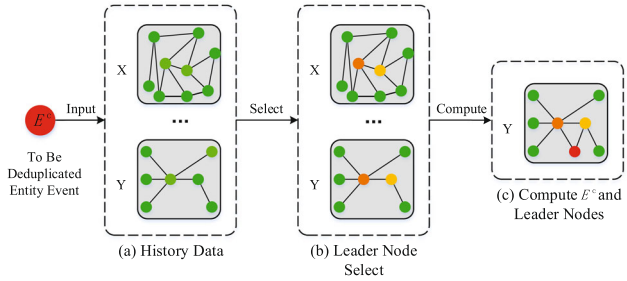


Fig. 4 The first stage process

4.1 TS-NSNO description

In this paper, we propose a two-stage deduplication framework, termed *TS-NSNO*, which aims to improve the deduplication method's performance and effectiveness. We describe the stages of the *TS-NSNO* method and model them mathematically.

As shown in Fig. 3, we first represent and vectorize the entity event and use *TF-IWF* to complete this step. The solid wireframe in the figure represents the main steps of *TS-NSNO*. The first stage is the node selection strategy, which mainly includes two steps. In the node selection step, we select the leader node set of the connected subgraph. The purpose is to reduce the number of deduplication calculations in the second step, which is to directly reduce the complexity of the algorithm. In the deduplication calculations step, we establish an entity event attribute correlation model and combine it with similarity to judge the event similarity, which can improve the effectiveness of the deduplication method.

The second stage is the node optimization strategy, which consists of two steps: node optimization and graph updating. The node optimization step mainly includes two modeling processes. Therefore, we establish the temporal and spatial distance model and the importance degree model through event development factors. Through the node optimization step, we can optimize the connection to the entity event that has been detected and can obtain the entity event more accurately and raise the recall. In the graph updating step, we update and calculate the leader nodes of the entity event new connected subgraph so that the next detection can perform the calculation in the latest environment to ensure the timeliness of the message. According to the proposed two-stage strategy, this paper forms a two-stage deduplication method based on graph node selection and optimization. We will introduce each stage in detail in the following sections.

4.2 The first stage: node selection

In the first stage, we first proposed selecting the leader node by the *node clustering coefficient* (Watts and Strogatz 1998), and second, we adopted a new deduplication calculation method that combines the event similarity with the event attribute correlation. The complete schematic diagram of the first stage is shown in Fig. 4. As shown in the figure, when the entity event E^c of the red node to be detected occurs, our first step is to read historical data to find all connected subgraphs under the current entity c. Second, we will select the top-K leader nodes according to the calculated node aggregation coefficient. We assume that the number of leader node sets top-K is 2, and there are two connected subgraphs of X and Y. Completing this step, we can see from the figure that the orange node is the first leader node, the yellow node is the second leader node, and the green nodes are not leader nodes. Third, the entity event E^c is calculated separately with the leader node set of each connected subgraph. Through calculation, we find that the entity event E^c is more similar to the leader node in the connected subgraph Y, so we connect it with the Y. We will introduce the specific calculation process in the next two subsections.

4.2.1 Leader node selection

In a subgraph, the leader node can represent most of the information of the subgraph. Therefore, the event deduplication model can reduce a large number of unnecessary calculations through the subgraph leader node selection strategy, thereby improving the performance of event deduplication. The traditional method needs to calculate the similarity with all nodes in all subgraphs. After introducing the subgraph leader node selection strategy, it only needs to calculate with the top-K leader nodes in all subgraphs.

This paper intends to use the *node clustering coefficient* to select the leader node of the subgraph. The *node clustering coefficient* mainly indicates the closeness of the node and the neighbor node. The larger the coefficient value is, the closer the connection between the node and the neighboring node, and vice versa. An undirected graph can be expressed as $G^c = (V, A)$, where V represents the set of all nodes, A represents the set of all edges, and the edge $a = (u, v) \in A$ represents the interconnection between node u and node v. The neighbor set $N(u)$ of node u is a group of nodes, which can be expressed by

$$N(u) = \{v \in V \mid (u, v) \in A\}. \tag{4}$$

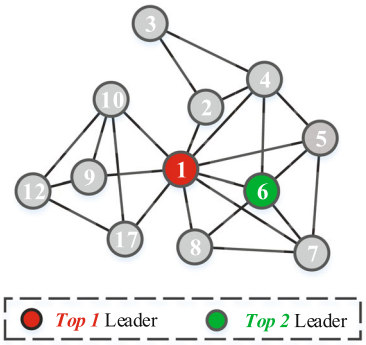


Fig. 5 Leader node selecting

Based on the neighbor set of the node, the *node clustering coefficient* is expressed by *NC(u)*, and the specific expression is shown in the following Eq. (5) by

$$NC\,(u) = \sum_{x \in N(u)} \frac{Z_{u,x}}{deg\,(x) + 1}. \tag{5}$$

In Eq. (5), $Z_{u,x}$ represents the number of triangles actually formed by edge *a(u, x)* and neighboring nodes, and deg*(x)* represents the degree of node *x*. *a(u, x)* represents the influence of node *v* on neighboring nodes. The higher the value of *a(u, x)*, the stronger the leadership of node *v* to neighbors.

As shown in Fig. 5, if the *K* value is 2, red node 1 has the largest clustering coefficient value. Therefore, node 1 is the top-*1* leader node of the subgraph, followed by green node 6, which is the top-*2* leader node of the subgraph. By adjusting the *K* value, the size of the lead node of the subgraph can be adjusted to ensure that the model can represent from rough to fine the characterization information of subgraphs.

From Eq. (5), the *node clustering coefficient* of each node of each connected subgraph can be obtained. According to the *node clustering coefficient*, the top-*K* leader nodes are selected. The set of leader nodes in each subgraph is expressed as follows:

$$D_q^c = \{g_1^c, g_2^c, g_3^c, \ldots, g_n^c\}(n = 1, 2, \ldots, q). \tag{6}$$

In Eq. (6), *c* represents an entity. *n* represents a connected subgraph under the entity, and the value range is (1, 2, 3,..., *K*). *K* represents the number of leader nodes. The D_q^c set represents *K* leader nodes of the *n-th* connected subgraph under entity *c*.

4.2.2 Deduplication calculation

In the first deduplication calculation, we combine the entity event similarity and the entity event correlation to make judgments. First, we calculate the similarity between the entity event pairs according to the literature (Wang et al. 2020). The equation for calculating the similarity between the entity event pair is as follows:

$$FS(E^c, V_{qk}^c) = \frac{E^c \cdot V_{qk}^c}{\|E^c\| \left\| V_{qk}^c \right\|}. \tag{7}$$

In Eq. (7), E^c represents the vectorized entity event to be detected, and V_{qk}^c is the *k-th* leader node in the set of leader nodes of the *q-th* connected subgraph and belongs to the set D_q^c. *FS* is the similarity between two entity events. When the value of *FS* is greater than the set threshold, it means that the two entity events are similar, and vice versa.

However, there are some shortcomings in the calculation of similarity and the accuracy of the deduplication method is low. It cannot capture the similarities and differences between words in detail, and there are some noise words in the sentence. Therefore, we propose a method to calculate the correlation of events according to the feature word attributes of events, to reduce the wrong judgment and improve the deduplication effectiveness.

We can analyze the attribute trigger words of entity events, and find that there are many different grammars (e.g., homonyms, heteronyms) or spelling orders (e.g., word order, expression, part of speech) in the trigger words. When these differences are present in vectorized sentences, the difference is very large, which leads to the low accuracy of the deduplication method based only on the similarity. Therefore, we analyze the trigger words and propose the concept of feature word mapping and modeling. When the trigger words are mapped to the same feature word, we consider these trigger words to be synonyms. One trigger word can only be mapped to one feature word, but one feature word can correspond to multiple trigger words. Therefore, this paper constructs a feature lexicon *Le*, which contains trigger words W_x, feature words l_y and the mapping relationship between them, and our mapping relationship is as follows:

$$f : W_x \to l_y. \tag{8}$$

In financial news, if the trigger words of the two entity events in comparison belong to the same feature words, it describes the same entity event. Second, if the trigger words of the two entity events compared do not belong to the same feature words, they are not described in the same feature word. Through the analysis of financial news, it is found that the mapping of trigger words and feature words can distinguish the differences between feature words more accurately. Therefore, this paper analyzes the feature word attributes of the entity event and models an entity event attribute correlation model with the following expression:

$$FW = \frac{f\left(l_y\left(E^c\right) \cup l_y\left(V_{qk}^c\right)\right)}{f\left(l_y\left(E^c\right) \cap l_y\left(V_{qk}^c\right)\right)}. \tag{9}$$

In Eq. (9), the function of f is to record the number of sets. $l_y(E^c)$ represents the mapping feature word set of the current entity events, and V_{qk}^c represents the mapping feature word set of the leader nodes. The numerator represents the sum of the characteristic coefficients of the two entity events having the same category. The denominator represents the sum of all the category characteristic coefficients of the two entity events. The higher the relevance of the trigger words, the more similar feature words, and the closer the FW value is to 1; vice versa, the closer it is to 0. Therefore, when the FW falls within the set threshold interval, the attribute correlation between the two entity events is high, and vice versa.

From Eqs. (7) and (9), the similarity and the correlation between the entity event pairs can be obtained. The judgment is as follows:

- If both FS and FW of the two entity events are greater than the threshold, then the two entity events are repeated;
- If the FS *or* FW of the two entity events is less than the threshold, then the two entity events are nonrepeated.

A detailed example of the algorithm is shown below in *Algorithm* 1.

Algorithm 1 *: TS-NS Algorithm*

Input:
The entity event data set of N samples, D;
Formula symbols: $NC(v)$; $FS(x, y)$; $FW(x, y)$;
Parameters:
Feature word correlation threshold δ,
Sentence similarity threshold z ,
Number of leader nodes top_k.
Output:
$v(E_p^c, V_{qk}^c)$, Edges constructed between nodes;
y_i, duplication label;

while true **do**
  $D^c \leftarrow D$ projected according to the history entity c
  for all $G_q^c \in D^c$ **do**
    $NC(v) \leftarrow \sum_{x \in N(v)} Z(v, x)/(deg(x) + 1)$
    $D_q^c \leftarrow Node\_Selection(NC(v), G_q^c, top_k)$
  end for
  for all $V_{qk}^c \in D_q^c$ **do**
    $E^c \leftarrow$ The current calculation node
    $SenSim \leftarrow FS(E^c, V_{qk}^c)$
    $FeaRela \leftarrow FW(E^c, V_{qk}^c)$
    if $FeaRela \geq \delta$ ***and*** $SenSim \geq z$ **then**
      $v(E^c, V_{qk}^c) \leftarrow 1\ (G_q^c(\{..., E^c, V_{qk}^c\}, v))$
      $y_i \leftarrow True$
    end if
  end for
end while
return $v(E_p^c, V_{qk}^c)$, y_i

4.3 The second stage: node optimization

In the second stage, we establish the temporal and spatial distance model and the importance degree model through event development factors. The detailed process of the specific strategy is shown in Fig. 6. First, the input of the second stage strategy is the output of the first stage, and we only need to optimize the updated connected subgraph. Second, we perform optimization calculations on the leader node connected to the red node. As shown in the figure, the red node is connected to the yellow and orange nodes, so we need to perform optimization calculations on them. Third, we can see from the figure that there is no development between events, so the original graph structure is returned. If development occurs, the connection to the node is deleted. If both the entity event and the connected leader node are developed, then all the relationships of the entity event will be deleted, and a new connected subgraph will be established. Finally, we update the graph in the database, calculate the *node clustering coefficient* of all nodes in the updated graph, and return to our label.

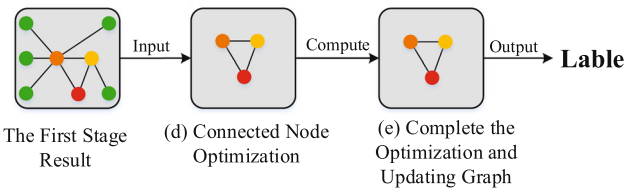


Fig. 6 The second stage process

4.3.1 Deduplication optimization

According to the first stage, entity events E^c are connected with similar leader nodes to form a new connected subgraph. However, the relationship between these nodes is not necessarily similar. When the entity event develops, that is the event has changed and the degree of importance has changed, it cannot be identified in the first stage. Therefore, we need to detect and optimize the connected nodes to make the effectiveness of this method reach a better state.

Combining the judgment of event development in the literature (Huang et al. 2014), we establish the temporal and spatial distance model and the importance degree model. The temporal and spatial distance between the two entity events is used to determine whether the space of the entity event has changed and the degree of change. We define the temporal and spatial distance of the entity events E^c and D_q^c as SP. The

calculation method of *SP* is as follows:

$$SP = exp\left(-\gamma \times \frac{d\left(t_i, t_j\right)}{T}\right). \tag{10}$$

In Eq. (10), t_i and t_j respectively represent the occurrence times of entity events E^c and V^c_{qk}, respectively, T is the time interval value set to take historical data, and λ is the time attenuation coefficient. The parameters of T and λ are changed in pairs to control the error caused by the excessively large time difference, and their values will be introduced in detail in the experiment see Sect. 5 for details, $d\left(t_i, t_j\right)$ represents the time difference between two entity events and the time difference between two entity events is shown as follows:

$$d\left(t_i, t_j\right) = \begin{cases} t_i - t_j, & \left(if : t_i \geq t_j\right); \\ 0, & \left(if : t_i < t_j\right). \end{cases} \tag{11}$$

When there is a time difference between the two entity events, the temporal and spatial distance between the two entity events can be calculated. If *SP* is between (0, *U*), then the two entity events produce temporal and spatial distance and the entity event to be detected is developing. If *SP* is between (*U*, 1), then the two entity events do not produce temporal and spatial distance. When there is no time difference between the two entity events, the temporal and spatial distance *SP* value is 1.

Although the temporal and spatial distance allows us to judge whether an event develops by the time it occurs, an entity event episode may appear over a temporal interval. Therefore, we also need to use the model of the degree of importance of the development of the event to judge. According to (Fedoryszak et al. 2019), the feature word list is manually graded to classify the importance degree. The importance degree of each feature word is classified under the first-level label. Second, the importance degree of event attributes is compared. The importance degree of an entity event is represented by *r*, and the importance degree of the entity event attribute of detected and connected subgraphs is rated. The specific calculation equation is as follows:

$$SI = \frac{r\left(l_y\left(E^c\right)\right)}{r\left(l_y\left(V^c_{qk}\right)\right)}. \tag{12}$$

In Eq. (12), $l_y\left(E^c\right)$ represents the importance value of the attribute of the entity event to be detected, $l_y\left(V^c_{qk}\right)$ on behalf of the entity events connected to the subgraph of the nodes in the attribute importance value. *SI* represents the change in the importance of two entity events and is detected by calculating the importance values of all attributes. When the importance degree value *r* of the entity event is obtained, the change in their importance degree is obtained by calculating the ratio of the two entity events. If *SI* is greater than or less than 1, the importance of the two entity events changes. Otherwise, the importance of the two entity events does not change.

From Eqs. (10) and (12), the temporal and spatial distance between the entity event pair and the change in importance degree can be obtained. Combining the two models to make repeated judgments are as follows:

- If *SP* is within the set threshold interval *and SI* is not equal to 1, then the current two entity events have evolved;
- If *SP* is not within the set threshold interval *or SI* is equal to 1, then the current two entity events have not evolved.

The detailed algorithm is shown in *Algorithm* 2.

Algorithm 2 *: TS-NO Algorithm*

Input:
$v(E^c_p, V^c_{qk})$, Edges constructed between similar nodes;
$SpaceTimeDistance$, space time distance parameter;
y_i, duplication label;
Output:
$\hat{y}_i$ two stage duplication label;
if $v(E^c_p, V^c_{qk}) == 1$ **then**
  $t_i \leftarrow$ Occurrence time of node E^c_p time
  $t_j \leftarrow$ Occurrence time of node V^c_{qk} time
  if $t_i - t_j \geq 0$ **then**
    $d(t_i, t_j) \leftarrow t_i - t_j$
  else
    $d(t_i, t_j) \leftarrow 0$
  end if
  $SP \leftarrow exp(-\lambda \times \frac{d(t_i, t_j)}{T})$
  $SI \leftarrow \sum_1^n r(E^c_p) / \sum_1^m r(V^c_{qk})$
  if $SP \leq SpaceTimeDistance$ ***and*** $SI \neq 1$ **then**
    $v(E^c_p, V^c_{qk}) \leftarrow 0$
    $y_i \leftarrow$ **False**
  end if
end if
return $\hat{y}_i$

4.4 Complexity analysis

One of the main goals of this method is to improve the efficiency of deduplication, so we analyze the time complexity of this method.

Assumption: First, suppose that there are *d* entity events in this dataset, and they all belong to the same entity. When constructing the subgraph, the number of leader nodes selected is *K*, and $K \leq d$.

Analysis: Since different deduplication methods require calculations on the content of the entity event, we do not consider it, only the number of calculations. In the first stage, when *d* pieces of data are nonrepetitive data, this time has the worst time complexity; then *d* subgraphs are constructed,

and the maximum time complexity is:

$$O_{first\text{-}max} = O\left(d \times (d-1)\right) = O\left(d^2\right), \tag{13}$$

when d pieces of data are repetitive data, this time has the worst time complexity; then, only one subgraph is constructed, and the minimum time complexity is:

$$O_{first\text{-}min} = O\left(d \times k\right) = O\left(d\right). \tag{14}$$

The second stage is to optimize the connected nodes. In a subgraph, it is calculated at most K times, and k is a constant. At this time, the maximum time complexity is:

$$O_{second} = O\left(d \times k\right) = O\left(d\right). \tag{15}$$

Therefore, the worst time complexity of the *TS-NSNO* algorithm proposed in this paper is shown in Eq. (16) as follows:

$$\begin{aligned} O_{TS\text{-}NSNO\text{-}max} &= O_{first\text{-}max} + O_{second} \\ &= O\left(d^2\right). \end{aligned} \tag{16}$$

The best time responsibility is shown in Eq. (17) as follows:

$$\begin{aligned} O_{TS\text{-}NSNO\text{-}min} &= O_{first\text{-}min} + O_{second} \\ &= O\left(d\right). \end{aligned} \tag{17}$$

According to the investigation of news reports, there are approximately 60% to 80% repetitions on the *Web*. We assume that the repetition rate is B, so the time complexity of this method is approximately:

$$\begin{aligned} T\left(d\right) &= O\left((1-B)\,d\right) \cdot O\left((1-B)\,d-1\right) \\ &= O\left((1-B)^2\,d^2 - d\right) \\ &= O\left(d \log_2 d\right). \end{aligned} \tag{18}$$

The time complexity of the general calculation method is $O\left(d \log_2 d\right)$, so this method can improve the efficiency of deduplication.

5 Experiments

This section describes the related preparations of the dataset, the environmental configuration, the most important comparison test design, and the selection of experimental parameters and results for display and analysis. We compare the most commonly used deduplication method on the entity event dataset and comprehensively evaluate the model we proposed.

Table 1 Data set items

Field name	Illustration
id	News number
Title	Title of the news release
Content	Content of the news
Release_time	News release time
Crawling_time	News crawl time
Company_entity	Extract the entity of the event
Company_subject	The remaining subjects
Trigger_words	Trigger word for extraction event
Sentences	Sentences representing entity events
Deduplicate_lable	Manually predict labels

Table 2 Data set

Data set name	Entity-num	Data-num
Entity event data-small	10	1356
Entity event data-middle	40	5203
Entity event data-big	70	8762
Entity event data-lager	100	12,330

5.1 Dataset description

Since there is no public data set for entity event deduplication, we analyze the event datasets *(e.g. ACE 2005 corpus, ASTRE [NTFB16])*, and we find that the data in these datasets did not all contain the fields required for this experiment (such as entity, time, etc.). Furthermore, it was not convenient to manually repeat the labeling of the data. Therefore, we choose to crawl data from several major financial news websites, for example, *Sohu News and Finance column*, *Toutiao Business column*, *Netease News and Finance column*, and *Sina Finance column*. For the analysis of the crawled text data, we first use the currently popular methods for entity recognition (Devlin et al. 2018). Second, we use the event extraction method based on trigger words and entities to extract events from the text and obtain all the entity events of each text according to the definition of entity events (Han et al. 2018). Finally, the main items contained in the dataset are shown in Table 1.

This experiment needs to test the performance. Therefore, we construct four datasets of the same type and different sizes, and a detailed display is shown in Table 2. In this experiment, a total of one hundred entities are included. The number of entity events in the largest dataset is 12,230, and the time difference is one month. This method is a real-time calculation method, so our dataset is sorted by release time (Navarro-Colorado and Saquete 2016).

As seen from Sect. 4, a lexicon *Le* is established in this paper, which can be both a mapping feature word and a trig-

Table 3 Environment configuration

Environment	Parameters
System version	Windows server 2012
CPU	Bronze 3160 CPU @ 1.70GHz
GPU	Nvidia Tesla V100
Python	Python 3.7

ger word lexicon. There are three types of trigger words in the lexicon: positive, negative, and neutral. Here, we would like to emphasize that we established the lexicon with professional data analysts and classified and rated the importance of trigger words. The mapping feature words in the final lexicon *Le* have 22 categories of positive words, 23 categories of negative words, and 3 categories of neutral words, and depending on the trigger word, the same feature word will have different levels. Our datasets and *Le* lexicon are publicly available at www.github.com/jiaxu-git/TS-NSNO.[1]

5.2 Experimental setup

We use standard measurements such as precision, recall, and *F1-score*. To test the efficiency of the model, the processing time of each subdataset is analyzed statistically, and the performance is compared according to the total processing time of each method. Each experiment is run 10 times with four datasets of different sizes, and the average result is taken as the final result of the model.

All the algorithms in this paper are run under the following environmental configuration. See Table 3 for details.

5.3 Algorithm design for comparison

This section mainly introduces the relevant comparative experiments in this paper. To highlight the accuracy and efficiency of the algorithm in this paper, we selected the field of entity event deduplication common deduplication methods and combined deduplication methods to design comparative experiments. The datasets tested in this paper are all entity event datasets constructed in this experiment.

We will classify and select methods according to different deduplication components, including the method of mixing components with better results. In this article, we selected five comparison algorithms. Among them, three methods are for different components, including a mixed method, and the other two are graph-based deduplication methods, as shown in Table 4.

[1] You can see the data set and dictionary built in this experiment through this link www.github.com/jiaxu-git/TS-NSNO.

Table 4 Comparison method

Methods	Condition	Vec method
Sen-method	Only based on sentence of entity event (McConky et al. 2012)	tf-iwf
Attr-method	Only based on attribution of entity event (Tomadaki and Salway 2005)	tf-iwf
Gra1-method	Based on attribution and relationships between attribution of entity event (Schinas et al. 2015)	tf-iwf
Gra2-method	Based on sentence, attribution of entity event and introduce *GCN* (Liu et al. 2018)	tf-iwf
Mix-method	Based on sentence and attribution of entity event (Bodankar and Waghmare 2020)	tf-iwf
TS-NSNO	Based on sentence and attribution of entity event	tf-iwf

5.3.1 Sen-method

This method performs repetitive detection based on the sentence description of the entity event. The entity event is extracted from the sentence-level document, therefore, the sentence descriptions are the most basic part of the entity event. This method directly uses sentence description for similarity calculation, which is one of the simplest and most convenient methods. This method can distinguish different entity events from sentence descriptions.

5.3.2 Attr-method

This method is based on the attributes of the entity event for duplication detection, such as the time, location, and object of the entity event. This method is also based on attribute-based entity event deduplication, and the entity event needs to be stored as a structured template. This method can perform duplication detection of entity events in a more detailed manner. Compared with the *Sen-method*, this method can improve the accuracy of deduplication.

5.3.3 Gra1-method

This method is based on the attributes of the entity event and the relationship between the attributes for duplication detection. In actual entity events, there are various relationships between attributes, but they are not considered in the

Attr-method. Therefore, this method is proposed and the relationship between attributes is considered for repeated detection, and the accuracy of deduplication is also improved.

5.3.4 Gra2-method

This method is based on sentence, attribution of entity event and the introduction of *GCN* to calculate the graph. First, the structure information of the graph is enriched by multi-content composition. Second, the use of *GCN* for graph representation can improve the extraction of semantic information of the graph and more accurately represent the semantic information of the text, thus improving the effect of deduplication.

5.3.5 Mix-method

This method is a hybrid method that combines the *Sen-method* and *Attr-method*. *Gra-method* has no advantage in calculation. Therefore, *Sen-method* is not combined and compared. We use different thresholds to set while considering the complete sentence and related attributes of the entity event, and it can better obtain accurate deduplication results.

5.4 TS-NSNO experimental parameter selection

This section will introduce the related parameter selection of the *TS-NSNO* experiment. The core parameters of this method are the leader node parameter K, the attribute correlation parameter Z, the space-time distance control parameter pair (T, λ), and the space-time distance parameter U. The rest are related parameters, which can be set directly without the literature. For example, the similarity threshold Y can be directly set to 0.7 according to the literature, and the change parameter of the importance of the entity event is 1 or 0. We will describe the selection of each core parameter separately see the following subsections for details.

5.4.1 Parameter K selection

The leader node parameter K is the main factor that affects the effectiveness and efficiency of the deduplication method. When the value of K is larger, the effectiveness of *TS-NSNO* is better and the efficiency is lower. When the value of K is smaller, the effectiveness of *TS-NSNO* is worse and the efficiency is higher. The smaller the value of K, the leader node set is not sufficient to represent all of the information sub-node graph, and the entity events leading to errors in judgment. The larger the value of K is, the longer the calculation time, resulting in lower efficiency. Therefore, it is necessary to consider the effectiveness and efficiency of *TS-NSNO* at the same time and select the best K value. As shown in Fig. 7, assuming that the best choice of Z is 0.7, we can see from Fig. 7a that as the value of K increases, the *F1-score* of *TS-NSNO* continues to increase. When the value of K reaches 9, the growth rate decreases. Figure 7b shows that as the value of K increases, the overall time consumption of *TS-NSNO* increases gradually. Therefore, combining the effectiveness and efficiency of *TS-NSNO*, the optimal value of K is 9.

5.4.2 Parameter Z selection

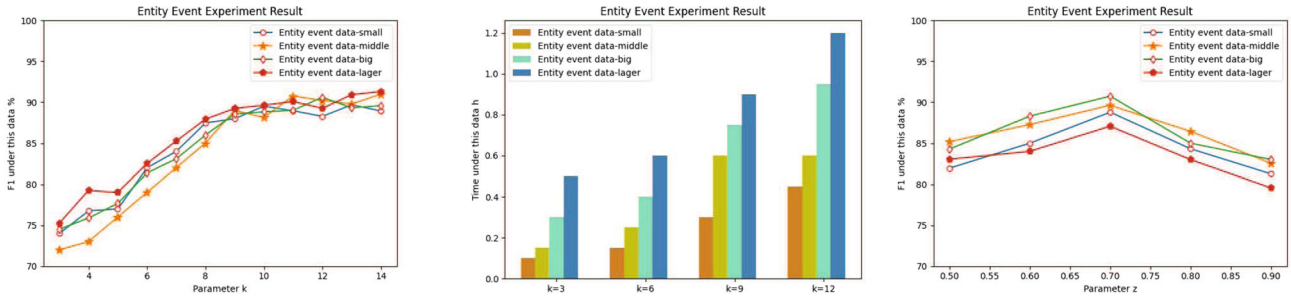
It can be seen from Fig. 7c that as the parameter Z value increases, the *F1-score* value gradually rises. When the value of Z is 0.7, the value of *F1-score* is the highest, and when Z increases again, the value of *F1-score* is accompanied by a downward trend. The parameter Z is the threshold of the relevance of the attributes of the entity event. The higher the relevance of the attributes of the two entity events, the greater the two entity events belong to the same type. When the threshold is gradually increased, there are fewer entity events within the range of correlation, and when the optimal value is reached, the *F1-score* value will decrease accordingly. Therefore, as shown in experimental Fig. 7c, the optimal value of parameter Z should be 0.7.

5.4.3 Parameter T and λ selection

In the temporal and spatial distance model, time is an important factor. With the development of entity events, the time intervals included in the diagram increase. Second, the development times of different event types are not consistent. How to balance the relationship between the new entity event and the old entity event requires careful consideration. Therefore, we propose the time range of the entity event value T and the parameter λ that controls the change in the model value caused by the increase in the dependent variable T. For the development of entity events, the time parameter T can have different parameter values. The common time parameter in practice is 7, 14, or 21 days. When the selected time range is different, the focus of the selected entity events is different. For the short time range parameter T, more timely information can be obtained, and for a long time range, more complete information can be obtained. The values of the corresponding time range control coefficient parameter λ are 0.7, 0.4, and 0.1. Therefore, the final time parameter pair (T, λ) is (7,0.7), (14,0.4), and (21,0.1).

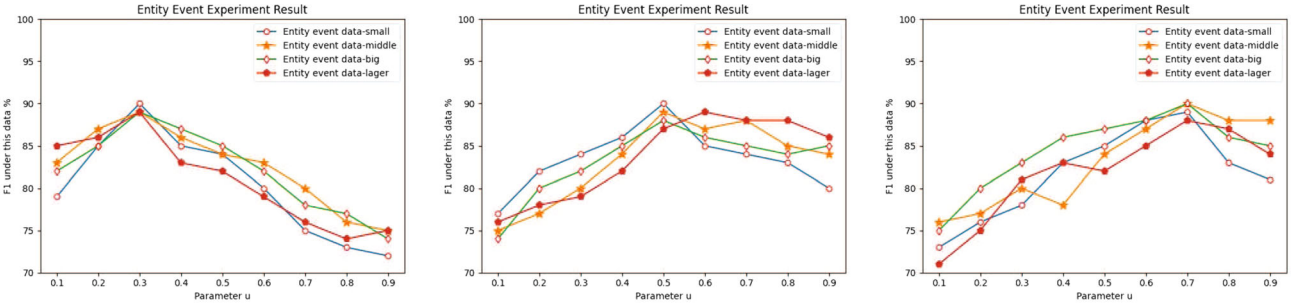
5.4.4 Parameter U selection

The parameter U is the threshold of the temporal and spatial distance SP. If the temporal and spatial distance exceeds the set threshold interval, there is no connection between the entity events; otherwise, there is a connection. In other words, the two entity events have not developed in time and space.



(a) *F1-score* value graph of parameter K (b) Performance graph of parameter K (c) Optimal results of parameters Z

Fig. 7 Optimal results of parameters K and Z



(a) when (T,λ)=(7,0.7) (b) when (T,λ)=(14,0.4) (c) when (T,λ)=(21,0.1)

Fig. 8 When taking different parameters (T, λ), the result of parameter U

Table 5 Display of the results of TS-NSNO

Method	Data sets	Acc	Recall	F1-score	Times (m)
TS-NS	Entity event data-small	86.13	82.28	84.11	19.80
	Entity event data-middle	86.02	81.28	83.58	33.60
	Entity event data-big	86.10	82.36	84.19	50.40
	Entity event data-lager	85.11	82.32	83.69	76.80
TS-NSNO	Entity event data-small	88.76	91.34	90.03	21.20
	Entity event data-middle	89.23	90.95	90.08	34.10
	Entity event data-big	88.94	91.11	90.01	50.40
	Entity event data-lager	88.67	91.00	89.82	77.30

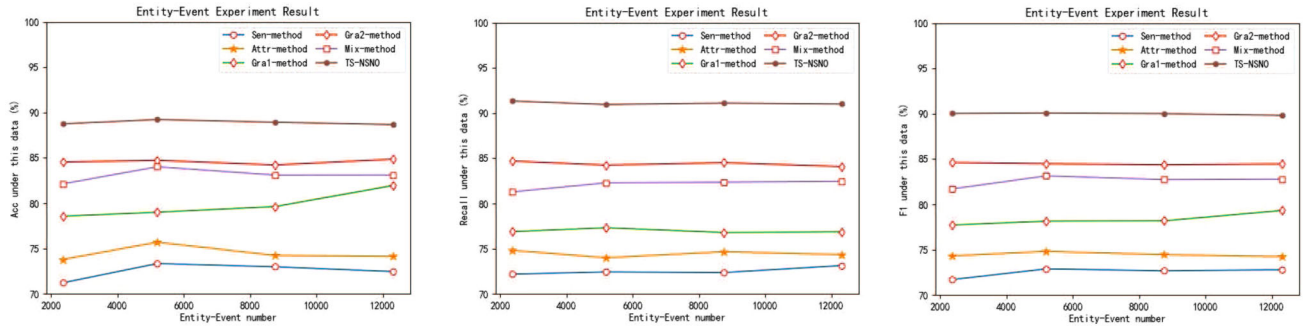
Figure 8a–c shows that under different (T,λ) parameter pairs, the value of U is different, and when the value of U is on both sides of the peak, duplication detection of entity events will be a large-scale misjudgment occurred, resulting in a drop in the *F1-score* value. Therefore, through experiments, we select the U values under different parameter pairs, respectively: 0.3, 0.5, and 0.7.

5.5 Effectiveness analysis

In this section, we first analyze the effect of the deduplication method *TS-NSNO* proposed in this paper and compare the accuracy, recall, and *F1-score* value of the two stages. Second, the *TS-NSNO* is compared and analyzed with five comparative experiments, and the final conclusion is drawn.

5.5.1 TS-NSNO experiment effectiveness analysis

From the previous section, we can conclude that the best parameter selection of this method and the time range parameter T we choose is 21. Therefore, the final results are obtained through experiments, as shown in Table 5. As we can see from the table, first, the accuracy of the first stage of the *TS-NSNO* algorithm is only 3% lower than that of the complete algorithm, but the recall is 9% different. Therefore, the second stage of the *TS-NSNO* can be greatly improved. Improve the recall and make the final *F1-score* average reach 90%. Second, the overall running time of the first stage of the *TS-NSNO* algorithm is not much different from the complete running time, so the second stage will not reduce the performance of the algorithm. Finally, we can conclude that



(a) accuracy (b) recall (c) F1-score

Fig. 9 The evaluation value of various algorithms, from the three aspects of precision, recall, and *F1-score*

both the first stage and the second stage of the *TS-NSNO* algorithm can reach the preset goal, and the *F1-score* value for deduplication can reach the current optimal value.

5.5.2 Comparative experiment analysis

Figure 9 is a line chart of the accuracy, recall and *F1-score* value of the five comparison algorithms and the *TS-NSNO* algorithm. Among them, in Fig. 9a, we can see six deduplication methods the highest accuracy is the *TS-NSNO* method, and the lowest is the *Sen-method*. In Fig. 9b, the recall rate of *TS-NSNO* is much higher than the other methods. In Fig. 9c, the largest *F1-score* is *TS-NSNO*, and the smallest is the *attr-metnod* deduplication method. Because the algorithm in this paper is also a combined method, the accuracy is not much different, and this paper also proposes two models based on event development factors to improve the recall rate of the deduplication method. Therefore, the *F1-score* value of our proposed *TS-NSNO* method is also the current optimal value.

In general, the *TS-NSNO* method can achieve better results on the deduplication task—with improvement of 10% on recall and 5% on *F1-score* score—which means that our proposed model can effectively improve recall and achieve high-efficiency deduplication.

5.6 Performance analysis

From the performance column of each method in Fig. 10, we can see that the total time increases as the amount of data increases. However, the growth rates of different methods are different. The fastest growth rate was obtained with the *Gra2-method*, and the slowest growth rate was obtained with the *TS-NSNO*. It can be explained that the selection of the leader node greatly reduces the amount of calculation, and the *TS-NSNO* method we proposed can effectively reduce the calculation time and achieve the real-time detection target. Moreover, *TS-NSNO* runs more than 50% faster than the other four algorithms on average.

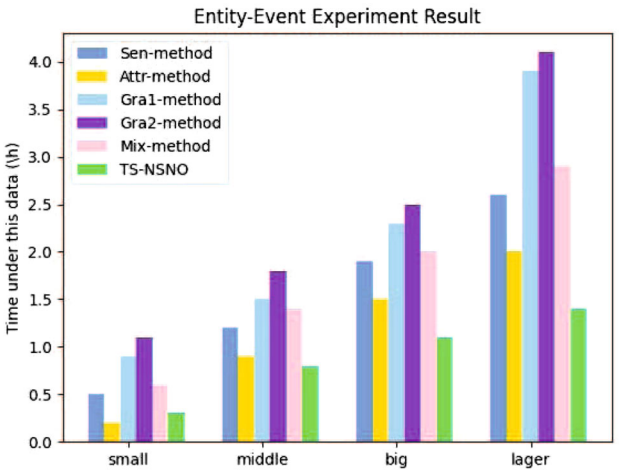


Fig. 10 The evaluation value of various algorithms efficiency

6 Conclusion and future work

In this paper, a two-stage deduplication strategy *TS-NSNO* is proposed to improve the accuracy and performance of the deduplication algorithm. In the first stage, the node selection strategy selects representative leader nodes from each connected subgraph and calculates the similarity between them and the data to be detected. In the second stage, the nodes are optimized to eliminate the nodes with incorrect connections and improved recall. In this paper, we test the accuracy and performance of the algorithm by creating the entity event dataset. The experimental results show that the two-stage optimization algorithm can better improve the accuracy and performance of deduplication.

In our future work, we will develop an improvement plan in three directions. The first is to increase the sample size of the dataset, which can more comprehensively detect the performance of the method. The second is the strategy selected in the optimization stage, aiming to find a more representative leader node. Finally, the characteristics of entity events

can be further refined and optimized, and a higher accuracy rate can be obtained.

Author Contributions The authors contributed to each part of this paper equally.

Funding This work was supported by National Natural Science Foundation of China (Grant No. 61802444), the Research Foundation of Education Bureau of Hunan Province of China (Grant No. 22B0275, No. 20B625, No. 18B196), and Local Community Structure Detection Algorithms in Complex Networks (Grant No. 2020YJ009).

Data Availability Data will be made available on request.

Declarations

Conflict of interest All authors declare that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Ai W, Xu J, Shao H et al (2021) An entity event deduplication method based on connected subgraph. In: 2021 7th international conference on systems and informatics (ICSAI), IEEE, pp 1–6

Arun P, Sumesh M (2015) Near-duplicate web page detection by enhanced TDW and simHash technique. In: 2015 international conference on computing and network communications (CoCoNet), IEEE, pp 765–770

Bodankar R, Waghmare M (2020) Int J Sci Res Sci Eng Technol. Identification and effective summary extraction with deduplication of data in news articles 7:96–102

Broder AZ (1997) On the resemblance and containment of documents. In: Proceedings. compression and complexity of SEQUENCES 1997 (Cat. No. 97TB100171), IEEE, pp 21–29

Charikar MS (2002) Similarity estimation techniques from rounding algorithms. In: Proceedings of the thirty-fourth annual ACM symposium on theory of computing, pp 380–388

Chen Z (2010) Graph-based clustering and its application in coreference resolution. In: Proceedings of the 2010 workshop on graph-based methods for natural language processing, pp 1–9

Devlin J, Chang MW, Lee K et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. pp 4171–4186

Fedoryszak M, Frederick B, Rajaram V et al (2019) Real-time event detection on social data streams. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2774–2782

Ge Y, Wu J, Dai G et al (2019) Text deduplication with minimum loss ratio. In: Proceedings of the 2019 11th international conference on machine learning and computing, pp 310–316

Han S, Hao X, Huang H (2018) An event-extraction approach for business analysis from online Chinese news. Electron Commerc Res Appl 28:244–260

Hossny AH, Mitchell L, Lothian N et al (2020) Feature selection methods for event detection in twitter: a text mining approach. Soc Netw Anal Min 10(1):1–15

Huang D, Hu S, Cai Y et al (2014) Discovering event evolution graphs based on news articles relationships. In: 2014 IEEE 11th international conference on e-business engineering, IEEE, pp 246–251

Jadhav A, Rajan V (2018) Extractive summarization with SWAP-NET: sentences and words from alternating pointer networks. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers), pp 142–151

Liu S, Liu K, He S et al (2016) A probabilistic soft logic based approach to exploiting latent and global information in event classification. In: Thirtieth AAAI conference on artificial intelligence, p 2993–2999

Liu B, Niu D, Wei H et al (2018) Matching article pairs with graphical decomposition and convolutions. arXiv preprint arXiv:1802.07459

Manku GS, Jain A, Das Sarma A (2007) Detecting near-duplicates for web crawling. In: Proceedings of the 16th international conference on World wide web, pp 141–150

McConky K, Nagi R, Sudit M et al (2012) Improving event coreference by context extraction and dynamic feature weighting. In: 2012 IEEE international multi-disciplinary conference on cognitive methods in situation awareness and decision support, IEEE, pp 38–43

Navarro-Colorado B, Saquete E (2016) Cross-document event ordering through temporal, lexical and distributional knowledge. Knowl Based Syst 110:244–254

Schinas M, Papadopoulos S, Petkos G et al (2015) Multimodal graph-based event detection and summarization in social media streams. In: Proceedings of the 23rd ACM international conference on multimedia, pp 189–192

Sharapova E, Sharapov R (2019) Detection of fuzzy duplicate texts in news feeds. 2019 systems of signal synchronization. Generating and processing in telecommunications (SYNCHROINFO), IEEE, pp 1–5

Tomadaki E, Salway A (2005) Matching verb attributes for cross-document event co-reference. In: Proceedings of interdisciplinary workshop on the identification and representation of verb features and verb classes, pp 127–132

UzZaman N, Allen JF (2010) Extracting events and temporal expressions from text. In: 2010 IEEE fourth international conference on semantic computing, IEEE, pp 1–8

Wang X, Dong X, Chen S (2020) Text duplicated-checking algorithm implementation based on natural language semantic analysis. In: 2020 IEEE 5th information technology and mechatronics engineering conference (ITOEC), IEEE, pp 732–735

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442

Yang CC, Shi X, Wei CP (2009) Discovering event evolution graphs from news corpora. IEEE Trans Syst Man Cybern Part A Syst Hum 39(4):850–863

Zhang X, Yao Y, Ji Y et al (2016) Effective and fast near duplicate detection via signature-based compression metrics. Math Probl Eng 10:1–12

Zhang X, Liu Z, Liu W et al (2011) Event similarity computation in text. In: 2011 International conference on internet of things and 4th international conference on cyber. Physical and social computing, IEEE, pp 419–423

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.