Article

FONDUE: A Framework for Node Disambiguation and Deduplication Using Network Embeddings [†]

Ahmad Mel , Bo Kang , Jefrey Lijffijt and Tijl De Bie ^{*}

AIDA, IDLab-ELIS, Ghent University, 9052 Ghent, Belgium; ahmad.mel@ugent.be (A.M.); bo.kang@ugent.be (B.K.); jefrey.lijffijt@ugent.be (J.L.)

^{*} Correspondence: tijl.debie@ugent.be

[†] This paper is an extended version of our paper published in IEEE DSAA 2020 The 7th IEEE International Conference on Data Science and Advanced Analytics.

Featured Application: FONDUE can be used to preprocess graph structured data, in particular it facilitates detecting nodes in the graph that represent the same real-life entity, and for detecting and optimally splitting nodes that represent multiple distinct real-life entities. FONDUE does this in an entirely unsupervised fashion, relying exclusively on the topology of the network.

Abstract: Data often have a relational nature that is most easily expressed in a network form, with its main components consisting of nodes that represent real objects and links that signify the relations between these objects. Modeling networks is useful for many purposes, but the efficacy of downstream tasks is often hampered by data quality issues related to their construction. In many constructed networks, ambiguity may arise when a node corresponds to multiple concepts. Similarly, a single entity can be mistakenly represented by several different nodes. In this paper, we formalize both the node disambiguation (NDA) and node deduplication (NDD) tasks to resolve these data quality issues. We then introduce FONDUE, a framework for utilizing network embedding methods for data-driven disambiguation and deduplication of nodes. Given an undirected and unweighted network, FONDUE-NDA identifies nodes that appear to correspond to multiple entities for subsequent splitting and suggests how to split them (node disambiguation), whereas FONDUE-NDD identifies nodes that appear to correspond to same entity for merging (node deduplication), using only the network topology. From controlled experiments on benchmark networks, we find that FONDUE-NDA is substantially and consistently more accurate with lower computational cost in identifying ambiguous nodes, and that FONDUE-NDD is a competitive alternative for node deduplication, when compared to state-of-the-art alternatives.

Keywords: node disambiguation; node deduplication; node linking; entity linking; network embeddings; representation learning

Citation: Mel, A.; Kang, B.; Lijffijt, J.; De Bie, T. FONDUE: A Framework for Node Disambiguation and Deduplication Using Network Embeddings. *Appl. Sci.* **2021**, *11*, 9884. <https://doi.org/10.3390/app11219884>

Academic Editors: Paola Velardi and Stefano Faralli

Received: 2 August 2021

Accepted: 18 October 2021

Published: 22 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Increasingly, collected data naturally comes in the form of a network of interrelated entities. Examples include social networks describing social relations between people (e.g., Facebook), citation networks describing the citation relations between papers (e.g., PubMed [1]), biological networks, such as those describing interactions between proteins (e.g., DIP [2]), and knowledge graphs describing relations between concepts or objects (e.g., DBpedia [3]). Thus, new machine learning, data mining, and information retrieval methods are increasingly targeting data in their native network representation.

An important problem across all the fields of data science, broadly speaking, is data quality. For problems on networks, especially those that are successful in exploiting fine- as well as coarse-grained structure of networks, ensuring good data quality is perhaps even more important than in standard tabular data. For example, an incorrect edge can have

a dramatic effect on the implicit representation of other nodes, by dramatically changing distances on the network. Similarly, mistakenly representing distinct real-life entities by the same node in the network may dramatically alter its structural properties, by increasing the degree of the node and by merging the possibly quite distinct neighborhoods of these entities into one. Conversely, representing the same real-life entity by multiple nodes can also negatively affect the topology of the graph, possibly even splitting apart communities.

Although identifying missing edges and, conversely, identifying incorrect edges, can be tackled adequately using link prediction methods, prior work has neglected the other task: identifying and appropriately splitting nodes that are ambiguous—i.e., nodes that correspond to more than one real-life entity. We will refer to this task as *node disambiguation* (NDA). A converse and equally important problem is the problem of identifying multiple nodes corresponding to the same real-life entity, a problem we will refer to as *node deduplication* (NDD).

This paper proposes a unified and principled framework to both NDA and NDD problems, called framework for node disambiguation and deduplication using network embeddings (FONDUE). FONDUE is inspired by the empirical observation that real (natural) networks tend to be easier to embed than artificially generated (unnatural) networks, and rests on the associated hypothesis that the existence of ambiguous or duplicate nodes makes a network less natural.

Although most of the existing methods tackling NDA and NDD make use of additional information (e.g., node attributes, descriptions, or labels) for identifying and processing these problematic nodes, FONDUE adopts a more widely applicable approach that relies solely on topological information. Although exploiting additional information may of course increase the accuracy on those tasks, we argue that a method that does not require such information offers unique advantages, e.g., when data availability is scarce, or when building an extensive dataset on top of the graph data, is not feasible for practical reasons. Additionally, this approach fits the privacy by design framework, as it eliminates the need to incorporate more sensitive data. Finally, we argue that, even in cases where such additional information is available, it is both of scientific and of practical interest to explore how much can be completed without using it, instead solely relying on the network topology. Indeed, although this is beyond the scope of the current paper, it is clear that methods that solely rely on network topology could be combined with methods that exploit additional node-level information, plausibly leading to improved performance of either type of approach individually.

1.1. The Node Disambiguation Problem

We address the problem of NDA in the most basic setting: given a network, unweighted, unlabeled, and undirected, the task considered is to identify nodes that correspond to multiple distinct real-life entities. We formulate this as an inverse problem, where we use the given *ambiguous network* (which contains ambiguous nodes) in order to retrieve the *unambiguous network* (in which all nodes are unambiguous). Clearly, this inverse problem is ill-posed, making it impossible to solve without additional information (which we do not want to assume) or an inductive bias.

The key insight in this paper is that such an inductive bias can be provided by the network embedding (NE) literature. This literature has produced embedding-based models that are capable of accurately modeling the connectivity of *real-life* networks down to the node-level, while being unable to accurately model *random* networks [4,5]. Inspired by this research, we propose to use as an inductive bias the fact that the unambiguous network must be easy to model using a NE. Thus, we introduce FONDUE-NDA, a method that identifies nodes as ambiguous if, after splitting, they maximally improve the quality of the resulting NE.

Example 1. Figure 1a illustrates the idea of FONDUE for NDA applied on a single node. In this example, node i with embedding $\mathbf{x}_i$ corresponds to two real-life entities that belong to two separate

communities, visualized by either full or dashed lines, to highlight the distinction. Because node i is connected to two different communities, most NE methods would locate its embedding $\mathbf{x}_i$ between the embeddings of the nodes from both communities. Figure 1b shows a split of node i into nodes i' and i'' , each with connections only to one of both communities. The resulting network is easy to embed by most NE methods, with embeddings $\mathbf{x}_{i'}$ and $\mathbf{x}_{i''}$ close to their own respective communities. In contrast, Figure 1c shows a split where the two resulting nodes are harder to embed. Most NE methods would embed them between both communities, but substantial tension would remain, resulting in a worse value of the NE objective function.

Figure 1. (a) A node that corresponds to two real-life entities that belongs to two communities. Links that connect the node with different communities are plotted in either full lines or dashed lines. (b) an ideal split that aligns well with the communities. (c) a less optimal split.

1.2. The Node Deduplication Problem

The same inductive bias can be used also for the NDD problem. The NDD problem is that given a network, unweighted, unlabeled, and undirected, identify distinct nodes that correspond to the same real-life entity. To this end, FONDUE-NDD determines how well merging two given nodes into one would improve the embedding quality of NE models. The inductive bias considers a merge as better than another one if it results in a better value of the NE objective function.

The diagram in Figure 2 shows the suggested pipeline for tackling both problems.

Figure 2. FONDUE pipeline for both NDA and NDD. Data corruption can lead to two types of problems: node ambiguation (e.g., multiple authors sharing the same name represented with one node in the network) in the left part of the diagram, and node duplication (e.g., one author with name variation represented by more than 1 node in the network). We then define two tasks to resolve both problems separately using FONDUE.

1.3. Contributions

In this paper, we make a number of related contributions:

- We propose FONDUE, a framework exploiting the empirical observation that naturally occurring networks can be embedded well using state-of-the-art NE methods, to tackle two distinct tasks: node deduplication (FONDUE-NDD) and node disambiguation (FONDUE-NDA). The former, by identifying nodes as more likely to be duplicated if contracting them enhances the quality of an optimal NE. The latter, by identifying nodes as more likely to be ambiguous if splitting them enhances the quality of an optimal NE;
- In addition to this conceptual contribution, substantial challenges had to be overcome to implement this idea in a scalable manner. Specifically for the NDA problem, through a first-order analysis we derive a fast approximation of the expected NE quality improvement after splitting a node;
- We implemented this idea for CNE [6], a recent state-of-the-art NE method, although we demonstrate that the approach can be applied for a broad class of other NE methods as well;
- We tackle the NDA problem, with extensive experiments over a wide range of networks demonstrate the superiority of FONDUE over the state-of-the-art for the *identification* of ambiguous nodes, and this at comparable computational cost;
- We also empirically observe that, somewhat surprisingly, despite the increase in accuracy for *identifying* ambiguous nodes, no such improvement was observed for the ambiguous node *splitting* accuracy. Thus, for NDA, we recommend using FONDUE for the *identification* of ambiguous nodes, while using an existing state-of-the-art approach for optimally splitting them;
- Experiments on four datasets for NDD demonstrate the viability of FONDUE-NDD for the NDD problem based on only the topological features of a network.

2. Related Work

The problem of NDA differs from named-entity disambiguation (NED; also known as named entity linking), a natural language processing (NLP) task where the purpose is to identify which real-life entity from a list a named-entity in a text refers to. For example, in the ArnetMiner dataset [7] ‘Bin Zhu’ corresponds to more than 10+ authors. The Open Researcher and Contributor ID (ORCID) [8] was introduced to solve the author name ambiguity problem, and most NED methods rely on ORCID for labeling datasets.

NED in this context aims to match the author names to unique (unambiguous) author identifiers [7,9–11].

In [7], they exploit hidden Markov random fields in a unified probabilistic framework to model node and edge features. On the other hand, Zhang et al. [12] designed a comprehensive framework to tackle name disambiguation, using complex feature engineering approach. By constructing paper networks, using the information sharing between two papers to build a supervised model for assigning the weights of the edges of the paper network. If two nodes in the network are connected, they are more likely to be authored by the same person.

Recent approaches are increasingly relying on more complex data, Ma et al. [13] used heterogeneous bibliographic networks representation learning, by employing relational and paper-related textual features, to obtain the embeddings of multiple types of nodes, while using meta-path based proximity measures to evaluate the neighboring and structural similarities of node embedding in the heterogeneous graphs.

The work of Zhang et al. [9] focusing on preserving privacy using solely the link information in a graph, employs network embedding as an intermediate step to perform NED, but they rely on other networks (person–document and document–document) in addition to person–person network to perform the task.

Although NDA could be used to assist in NED tasks, NED typically strongly relies on the text, e.g., by characterizing the context in which the named entity occurs (e.g., paper

topic) [14]. Similarly, Ma et al. [15] proposes a name disambiguation model based on representation learning employing attributes and network connections, by first encoding the attributes of each paper using variational graph auto-encoder, then computing a similarity metric from the relationship of these attributes, and then using graph embedding to leverage the author relationships, heavily relying on NLP.

In NDA, in contrast, no natural language is considered, and the goal is to rely on just the network's connectivity in order to identify which nodes may correspond to multiple distinct entities. Moreover, NDA does not assume the availability of a list of known unambiguous entity identifiers, such that an important part of the challenge is to identify which nodes are ambiguous in the first place. This offers a more privacy-friendly advantage and extends the application towards more datasets where access to additional information is restricted or not possible.

The research by Saha et al. [16], and Hermansson et al. [17] is most closely related to ours. These papers also only use topological information of the network for NDA. Yet, Ref. [16] also require timestamps for the edges, while [17] require a training set of nodes labeled as ambiguous and non-ambiguous. Moreover, even though the method proposed by [16] is reportedly orders of magnitude faster than the one proposed by [17], it remains computationally substantially more demanding than FONDUE (e.g., [16] evaluate their method on networks with just 150 entities). Other recent work using NE for NED [9,18–20] is only related indirectly as they rely on additional information besides the topology of the network.

The literature on NDD is scarce, as the problem is not well-defined. Conceptually, it is similar to that of named entity linking (NEL) [11,21] problem which aims to link instances of named entities in a text such as a newspaper, articles to the corresponding entities, often in knowledge bases (KB). Consequently, NEL heavily relies on textual data to identify erroneous entities rather than entity connection which is the core of our method. KB approaches for NEL are dominant in the field [22,23], as they make use of knowledge base datasets, heavily relying on labeled and additional graph data to tackle the named entity linking task. This also poses a challenge when it comes to benchmarking our method for NDD. No identified studies that tackles NDD from a topological approach is present in the current literature, at least without reliance on additional attributes and features.

3. Methods

Section 3.1 formally defines the NDA and NDD problems. Section 3.2 introduces the FONDUE framework in a maximally generic manner, independent of the specific NE method it is applied to, or the task (NDD or NDE) it is used for. A scalable approximation of FONDUE-NDA is described throughout Section 3.3, and applied to CNE as a specific NE method. Section 3.4 details the FONDUE-NDD method used for NDD.

Throughout this paper, a bold uppercase letter denotes a matrix (e.g., A), a bold lower case letter denotes a column vector (e.g., $\mathbf{x}_i$), $(\cdot)^\top$ denotes matrix transpose (e.g., $A^\top$), and $\|\cdot\|$ denotes the Frobenius norm of a matrix (e.g., $\|A\|$).

3.1. Problem Definition

We denote an undirected, unweighted, unlabeled graph as $\mathcal{G} = (V, E)$, with $V = \{1, 2, \dots, n\}$ the set of n nodes (or vertices), and $E \subseteq \binom{V}{2}$ the set of edges (or links) between these nodes. We also define the adjacency matrix of a graph $\mathcal{G}$, denoted $A \in \{0, 1\}^{n \times n}$, with $A_{ij} = 1$ if $\{i, j\} \in E$. We denote $\mathbf{a}_i \in \{0, 1\}^n$ as the adjacency vector for node i , i.e., the i th column of the adjacency matrix A , and $\Gamma(i) = \{j \mid \{i, j\} \in E\}$ the set of neighbors of i .

3.1.1. Formalizing the Node Disambiguation Problem

To formally define the NDA problem as an inverse problem, we first need to define the forward problem which maps an unambiguous graph onto an ambiguous one. This formalizes the 'corruption' process that creates ambiguity in the graph. In practice, this happens most often because identifiers of the entities represented by the nodes are not

unique. For example, in a co-authorship network, the identifiers could be non-unique author names. To this end, we define a node contraction:

Definition 1 (Node Contraction). *A node contraction c for a graph $\mathcal{G} = (V, E)$ with $V = \{1, 2, \dots, n\}$ is a surjective function $c : V \rightarrow \hat{V}$ for some set $\hat{V} = \{1, 2, \dots, \hat{n}\}$ with $\hat{n} \leq n$. For convenience, we will define $c^{-1} : \hat{V} \rightarrow 2^V$ as $c^{-1}(i) = \{k \in V | c(k) = i\}$ for any $i \in \hat{V}$. Moreover, we will refer to the cardinality $|c^{-1}(i)|$ as the multiplicity of the node $i \in \hat{V}$.*

A node contraction defines an equivalence relation $\sim_c$ over the set of nodes: $i \sim_c j$ if $c(i) = c(j)$, and the set $\hat{V}$ is the quotient set $V / \sim_c$. Continuing our example of a co-authorship network, a node contraction maps an author onto the node representing their name. Two authors i and j would be equivalent if their names $c(i)$ and $c(j)$ are equal, and the multiplicity of a node is the number of distinct authors with the corresponding name.

We can naturally define the concept of an ambiguous graph in terms of the contraction operation, as follows.

Definition 2 (Ambiguous graph). *Given a graph $\mathcal{G} = (V, E)$ and a node contraction c for that graph, the graph $\hat{\mathcal{G}} = (\hat{V}, \hat{E})$ defined as $\hat{E} = \{\{c(k), c(l)\} | \{k, l\} \in E\}$ is referred to as an ambiguous graph of $\mathcal{G}$. Overloading notation, we may write $\hat{\mathcal{G}} \triangleq c(\mathcal{G})$. To contrast $\mathcal{G}$ with $\hat{\mathcal{G}}$, we may refer to $\mathcal{G}$ as the unambiguous graph.*

Continuing the example of the co-authorship network, the contraction operation can be thought of as the operation that replaces author identities with their names, which may map distinct authors onto a same shared name. Note that the symbols for the ambiguous graph and its set of nodes and edges are denoted here using hats, to indicate that in the NDA problem we are interested in situations where the ambiguous graph is the empirically observed graph.

We can now formally define the NDA problem as inverting this contraction operation:

Definition 3 (The Node Disambiguation Problem). *Given an ambiguous graph $\hat{\mathcal{G}} = (\hat{V}, \hat{E})$, NDA aims to retrieve the unambiguous graph $\mathcal{G} = (V, E)$ and associated node contraction c , i.e., a contraction c for which $c(\mathcal{G}) = \hat{\mathcal{G}}$.*

To be more precise, it suffices to identify $\mathcal{G}$ up to an isomorphism, as the actual identifiers of the nodes are irrelevant.

3.1.2. Formalizing the Node Deduplication Problem

The NDD problem can be formalized as the converse of the NDA problem, also relying on the concept of node contractions. First, a duplicate graph can be defined as follows:

Definition 4 (Duplicate graph). *Given a graph $\mathcal{G} = (V, E)$, a graph $\hat{\mathcal{G}} = (\hat{V}, \hat{E})$ where $\{k, l\} \in \hat{E} \Rightarrow \{c(k), c(l)\} \in E$ for an appropriate contraction c , and where for each $\{i, j\} \in E$ there exists an edge $\{k, l\} \in \hat{E}$ for which $c(k) = i$ and $c(l) = j$, is referred to as a duplicate graph of $\mathcal{G}$. Or more concisely, using the overloaded notation from Definition 2, a duplicate graph $\hat{\mathcal{G}}$ is a graph for which $c(\hat{\mathcal{G}}) = \mathcal{G}$. To contrast $\mathcal{G}$ with $\hat{\mathcal{G}}$, we may refer to $\mathcal{G}$ as the deduplicated graph.*

Continuing the example of the co-authorship network, one node in the duplicate graph could correspond to two versions of the name of the same author, such that they are assigned two different nodes in the duplicate graph. A contraction operation that maps duplicate names to their common identity would merge such nodes corresponding to the same author. Hats on top of the symbols of the duplicate graph indicate that in the NDD problem we are interested in the situation where the duplicate graph is the empirically observed one.

The NDD problem can, thus, be formally defined as follows:

Definition 5 (The Node Deduplication Problem). *Given a duplicate graph $\hat{\mathcal{G}} = (\hat{V}, \hat{E})$, NDD aims to retrieve the deduplicated graph $\mathcal{G} = (V, E)$ and the node contraction c associated with $\hat{\mathcal{G}}$, i.e., for which $\mathcal{G} = c(\hat{\mathcal{G}})$.*

3.1.3. Real Graphs Suffer from Both Issues

Of course, many real graphs both require deduplication and disambiguation. This is particularly true for the running example of the co-authorship network. Yet, while building on the common FONDUE framework, we define and study both problems separately, and propose an algorithm for each in Section 3.3 (for NDA) and Section 3.4 (for NDD). For networks suffering from both problems, both algorithms can be applied concurrently or sequentially without difficulties, thus solving both problems simultaneously.

3.2. FONDUE as a Generic Approach

To address both the NDA and NDD problems, FONDUE uses an inductive bias that the non-corrupted (unambiguous and deduplicated) network must be easy to model using NE. This allows us to approach both problems in the context of NE. Here we first formalize the inductive bias of FONDUE (Section 3.2). This will later allow us to present both the FONDUE-NDA (Section 3.3) and FONDUE-NDD (Section 3.4) algorithms, each tackling one of the data corruption tasks (NDA and NDD, respectively).

The FONDUE Induction Bias

Clearly, both the NDA and NDD problems are inverse problems, with NDA an ill-posed one. Thus, further assumptions, inductive bias, or priors are inevitable in order to solve them. The key hypothesis in FONDUE is that the unambiguous and deduplicated $\mathcal{G}$, considering it is a ‘natural’ graph, can be embedded well using state-of-the-art NE methods. This hypothesis is inspired by the empirical observation that NE methods embed ‘natural’ graphs well.

NE methods find a mapping $f : V \rightarrow \mathbb{R}^d$ from nodes to d -dimensional real vectors. An embedding is denoted as $X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^\top \in \mathbb{R}^{n \times d}$, where $\mathbf{x}_i \triangleq f(i)$ for $i \in V$ is the embedding of each node. Most well-known NE methods aim to find an optimal embedding $X_{\mathcal{G}}^*$ for given graph $\mathcal{G}$ that minimizes a continuous differentiable cost function $\mathcal{O}(\mathcal{G}, X)$.

Thus, given an ambiguous graph $\hat{\mathcal{G}}$, FONDUE-NDA will search for the graph $\mathcal{G}$, such that $c(\mathcal{G}) = \hat{\mathcal{G}}$ for an appropriate contraction c , while optimizing the NE cost function on $\mathcal{G}$:

Definition 6 (NE-based NDA problem). *Given an ambiguous graph $\hat{\mathcal{G}}$, NE-based NDA aims to retrieve the unambiguous graph $\mathcal{G}$ and the associated contraction c :*

$$\begin{aligned} \underset{\mathcal{G}}{\operatorname{argmin}} \quad & \mathcal{O}(\mathcal{G}, X_{\mathcal{G}}^*) \\ \text{s.t.} \quad & c(\mathcal{G}) = \hat{\mathcal{G}} \text{ for some contraction } c. \end{aligned} \quad (1)$$

Ideally, this optimization problem can be solved by simultaneously finding optimal splits for all nodes (i.e., an inverse of the contraction c) that yield the smallest embedding cost after re-embedding. However, this strategy requires to (a) search splits in an exponential search space that has the combinations of splits (with arbitrary cardinality) of all nodes, (b) to evaluate each combination of the splits, the embedding of the resulting network needs to be recomputed. Thus, this ideal solution is computationally intractable and more scalable solutions are needed (see Section 3.3).

Similarly, for NDD, given a duplicate graph $\hat{\mathcal{G}}$, FONDUE-NDD will search for a graph $\mathcal{G}$, such that $c(\hat{\mathcal{G}}) = \mathcal{G}$ for an appropriate contraction c , again while optimizing the NE cost function on $\mathcal{G}$:

Definition 7 (NE-based NDD problem). Given a duplicate graph $\hat{\mathcal{G}}$, NE-based NDD aims to retrieve the deduplicated graph $\mathcal{G}$ and the associated contraction c of $\hat{\mathcal{G}}$:

$$\begin{aligned} \underset{\mathcal{G}}{\operatorname{argmin}} \quad & \mathcal{O}(\mathcal{G}, \mathbf{X}_{\mathcal{G}}^*) \\ \text{s.t.} \quad & c(\hat{\mathcal{G}}) = \mathcal{G} \text{ for some contraction } c. \end{aligned} \quad (2)$$

Generally speaking, to solve this optimization problem, we would want to find the optimal merging for all the nodes that would reduce the cost of the embedding after computing the re-embedding. Yet, a thorough optimization of this problem is beyond the scope of this paper, and as an approximation we rely on a ranking-based approach where we rank networks with randomly merged nodes depending on the value of the objective function after re-embedding. This may be suboptimal, but it highlights the viability of the concept if used for NDD as shown in the results of the experiments.

Although the principle underlying both methods is thus very similar, we will see below that the corresponding methods differ considerably. In common to them is the need for a basic understanding of NE methods.

3.3. FONDUE-NDA

From the above section, it is clear that the NDA problem can be decomposed into two subproblems:

1. Estimating the multiplicities of all $i \in \hat{\mathcal{G}}$ —i.e., the number of unambiguous nodes from $\mathcal{G}$ represented by the node from $\hat{\mathcal{G}}$. This essentially amounts to estimation the contraction c . Note that the number of nodes n in V is then equal to the sum of these multiplicities, and arbitrarily assigning these n nodes to the sets $c^{-1}(i)$ defines c^{-1} and, thus, c ;
2. Given c , estimating the edge set E . To ensure that $c(\mathcal{G}) = \hat{\mathcal{G}}$, for each $\{i, j\} \in \hat{E}$ there must exist at least one edge $\{k, l\} \in E$ with $k \in c^{-1}(i)$ and $l \in c^{-1}(j)$. However, this leaves the problem underdetermined (making this problem ill-posed), as there may also exist multiple such edges.

As an inductive bias for the second step, we will additionally assume that the graph $\mathcal{G}$ is sparse. Thus, FONDUE-NDA estimates $\mathcal{G}$ as the graph with the smallest set E for which $c(\mathcal{G}) = \hat{\mathcal{G}}$. Practically, this means that an edge $\{i, j\} \in \hat{E}$ results in exactly one edge $\{k, l\} \in E$ with $k \in c^{-1}(i)$ and $l \in c^{-1}(j)$, and that equivalent nodes $k \sim_c l$ with $k, l \in V$ are never connected by an edge, i.e., $\{k, l\} \notin E$. This bias is justified by the sparsity of most ‘natural’ graphs, and our experiments indicate it is justified.

We approach the NE-based NDA Problem 6 in a greedy and iterative manner. In each iteration, FONDUE-NDA identifies the node that has a split which will result in the smallest value of the cost function among all nodes. To further reduce the computational complexity, FONDUE-NDA only splits one node into two nodes at a time (e.g., Figure 1b), i.e., it splits node i into two nodes i' and i'' with corresponding adjacency vectors $\mathbf{a}_{i'}, \mathbf{a}_{i''} \in \{0, 1\}^n$, $\mathbf{a}_{i'} + \mathbf{a}_{i''} = \mathbf{a}_i$. We refer to such a split as a *binary split*. Note that repeated binary splits can of course be used to achieve the same result as a single split into several nodes, so this assumption does not imply a loss of generality or applicability. Once the best binary split of the best node is identified, FONDUE-NDA splits that node and starts the next iteration. The evaluation of each split requires recomputing the embedding, and comparing the resulting optimal NE cost functions with each other.

Unfortunately, this naive strategy is computationally intractable: computing a single NE is already computationally demanding for most (if not all) NE methods. Thus, having to compute a re-embedding for all possible splits, even binary ones (there are $O(n2^d)$ of them, with n the number of nodes and d the maximal degree), is entirely infeasible for practical networks.

3.3.1. A First-Order Approximation for Computational Tractability

Thus, instead of recomputing the embedding, FONDUE-NDA performs a first-order analysis by investigating the effect of an infinitesimal split of a node i around its embedding $\mathbf{x}_i$, on the cost $\mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})$ obtained after performing the splitting, with $\hat{\mathcal{G}}_{si}$ and $\hat{\mathbf{X}}_{si}$ referring to the ambiguous graph and its embeddings' representation, respectively, after splitting node i .

Drawing intuition from Figure 1, when two distinct authors share the same name in a given collaboration network, their respective separate community (ego-network) are lumped into one big cluster. Yet, from a topological point of view, that ambiguous node (author name) is connected to both communities that are generally different, meaning they share very few, if any, links. This stems from the observation that it is highly unlikely that two authors with the same exact name would belong to the same community, i.e., collaborate together. Furthermore, splitting this ambiguous node into two different ones (distinguishing the two authors), would ideally separate these two communities. Thus, to do so, we consider that each community, that is supposed to be embedded separately, is pulling the ambiguous node towards its own embedding region, and once separated, the embeddings of each of the resolved nodes will be improved. So our main goal is to quantify the amount of improvements in the embedding cost function by separating the two nodes i' and i'' by a unit distance in a certain direction. We propose to split the assignment of the edges of i between i' and i'' , such that all the links from i are distributed to either i' or i'' in such way to maximize the embedding cost function, which could be evaluated by computing the gradient with respect to the separation distance δ_i .

Specifically, FONDUE-NDA seeks the split of node i that will result in embedding $\mathbf{x}_{i'}$ and $\mathbf{x}_{i''}$ with infinitesimal difference δ_i (where $\delta_i = \mathbf{x}_{i'} - \mathbf{x}_{i''}$, $\mathbf{x}_{i'} = \mathbf{x}_i + \frac{\delta_i}{2}$, $\mathbf{x}_{i''} = \mathbf{x}_i - \frac{\delta_i}{2}$, and $\delta_i \rightarrow 0$, e.g., Figure 1b), such that $\|\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})\|$ is large, with $\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})$ being the gradient of $\mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})$ with respect to δ_i . This can be completed analytically. Indeed, applying the chain rule, we find:

$$\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) = \frac{1}{2} \nabla_{\mathbf{x}_{i'}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) - \frac{1}{2} \nabla_{\mathbf{x}_{i''}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}). \quad (3)$$

Many recent NE methods like LINE [24] and CNE [6], aim to embed 'similar' nodes in the graph closer to each other, and 'dissimilar' nodes further away from each other (for a particular similarity notion depending on the NE method). For such methods, Equation (3) can be further simplified. Indeed, as such NE methods focus on modeling a property of pairs of nodes (their similarity), their objective functions can be typically decomposed as a summation of node-pair interaction losses over all node-pairs. For example, this can be seen in Section 3.3.3 of the current paper for CNE [6], and in Equations (3) and (6) of [24] for LINE. Each of these node-pair interaction losses quantifies the extent to which the proximity between nodes' embeddings reflects their 'similarity' in the network. For methods where this decomposition is possible, we can thus write the objective function as follows:

$$\begin{aligned} \mathcal{O}(\mathcal{G}, \mathbf{X}) &= \sum_{j: \{i,j\} \in V \times V} \mathcal{O}^p(A_{ij}, \mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j: \{i,j\} \in E} \mathcal{O}^p(A_{ij} = 1, \mathbf{x}_i, \mathbf{x}_j) + \sum_{l: \{k,l\} \notin E} \mathcal{O}^p(A_{kl} = 0, \mathbf{x}_k, \mathbf{x}_l), \end{aligned}$$

where $\mathcal{O}^p(A_{ij}, \mathbf{x}_i, \mathbf{x}_j)$ denotes the node-pair interaction loss for the nodes i and j , $\mathcal{O}^p(A_{ij} = 1, \mathbf{x}_i, \mathbf{x}_j)$ the part of objective function that corresponds to node i and node j with an edge between them ($A_{ij} = 1$) and $\mathcal{O}^p(A_{kl} = 0, \mathbf{x}_k, \mathbf{x}_l)$ is the part of objective function, where node k and node l are disconnected.

Given that $\Gamma(i) = \Gamma(i') \cup \Gamma(i'')$ and $\Gamma(i') \cap \Gamma(i'') = \emptyset$, we can apply the same decomposition approach on $\nabla_{\mathbf{x}_{i'}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})$,

$$\begin{aligned} \nabla_{\mathbf{x}_{i'}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) &= \nabla_{\mathbf{x}_{i'}} \sum_{j:j \in \Gamma(i')} \mathcal{O}^p(A_{i'j} = 1, \mathbf{x}_{i'}, \mathbf{x}_j) + \nabla_{\mathbf{x}_{i'}} \sum_{l:l \notin \Gamma(i')} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l) \\ &= \nabla_{\mathbf{x}_{i'}} \sum_{j:j \in \Gamma(i')} \mathcal{O}^p(A_{i'j} = 1, \mathbf{x}_{i'}, \mathbf{x}_j) + \nabla_{\mathbf{x}_{i'}} \sum_{l:l \in \Gamma(i'')} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l) \\ &\quad + \nabla_{\mathbf{x}_{i'}} \sum_{l:l \notin \Gamma(i)} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l), \end{aligned}$$

and similarly on $\nabla_{\mathbf{x}_{i''}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})$,

$$\begin{aligned} \nabla_{\mathbf{x}_{i''}} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) &= \nabla_{\mathbf{x}_{i''}} \sum_{j:j \in \Gamma(i'')} \mathcal{O}^p(A_{i''j} = 1, \mathbf{x}_{i''}, \mathbf{x}_j) + \nabla_{\mathbf{x}_{i''}} \sum_{l:l \in \Gamma(i')} \mathcal{O}^p(A_{i''l} = 0, \mathbf{x}_{i''}, \mathbf{x}_l) \\ &\quad + \nabla_{\mathbf{x}_{i''}} \sum_{l:l \notin \Gamma(i)} \mathcal{O}^p(A_{i''l} = 0, \mathbf{x}_{i''}, \mathbf{x}_l). \end{aligned}$$

Additionally, as both nodes i' and i'' share the same set of non-neighbors of node i , we can write the following:

$$\nabla_{\mathbf{x}_{i'}} \sum_{l:l \notin \Gamma(i)} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l) = \nabla_{\mathbf{x}_{i''}} \sum_{l:l \notin \Gamma(i)} \mathcal{O}^p(A_{i''l} = 0, \mathbf{x}_{i''}, \mathbf{x}_l).$$

Furthermore, incorporating the previous two decompositions, we can rewrite Equation (3) as follows:

$$\begin{aligned} \nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) &= \frac{1}{2} \left[\sum_{j:j \in \Gamma(i')} \nabla_{\mathbf{x}_{i'}} \mathcal{O}^p(A_{i'j} = 1, \mathbf{x}_{i'}, \mathbf{x}_j) - \sum_{l:l \in \Gamma(i')} \nabla_{\mathbf{x}_{i''}} \mathcal{O}^p(A_{i''l} = 0, \mathbf{x}_{i''}, \mathbf{x}_l) \right. \\ &\quad \left. + \sum_{l:l \in \Gamma(i'')} \nabla_{\mathbf{x}_{i'}} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l) - \sum_{j:j \in \Gamma(i'')} \nabla_{\mathbf{x}_{i''}} \mathcal{O}^p(A_{i''j} = 1, \mathbf{x}_{i''}, \mathbf{x}_j) \right] \\ &= \frac{1}{2} \sum_{j:j \in \Gamma(i')} [\nabla_{\mathbf{x}_{i'}} \mathcal{O}^p(A_{i'j} = 1, \mathbf{x}_{i'}, \mathbf{x}_j) - \nabla_{\mathbf{x}_{i''}} \mathcal{O}^p(A_{i''l} = 0, \mathbf{x}_{i''}, \mathbf{x}_l)] \\ &\quad - \frac{1}{2} \sum_{l:l \in \Gamma(i'')} [\nabla_{\mathbf{x}_{i''}} \mathcal{O}^p(A_{i''j} = 1, \mathbf{x}_{i''}, \mathbf{x}_j) - \nabla_{\mathbf{x}_{i'}} \mathcal{O}^p(A_{i'l} = 0, \mathbf{x}_{i'}, \mathbf{x}_l)]. \end{aligned}$$

Given that $\Gamma(i')^c = \Gamma(i) - \Gamma(i') = \Gamma(i'')$ and $\Gamma(i'')^c = \Gamma(i) - \Gamma(i'') = \Gamma(i')$, the above equation could be simplified to:

$$\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) = \frac{1}{2} \sum_{j:j \in \Gamma(i)} (-1)^m [\nabla_{\mathbf{x}_i} \mathcal{O}^p(A_{ij} = 1, \mathbf{x}_i, \mathbf{x}_j) - \nabla_{\mathbf{x}_i} \mathcal{O}^p(A_{ij} = 0, \mathbf{x}_i, \mathbf{x}_j)], \quad (4)$$

with $m = 0$ if $j \in \Gamma(i')$ and $m = 1$ if $j \in \Gamma(i'')$.

Let $\mathbf{F}_i^1 \in \mathbb{R}^{d \times |\Gamma(i)|}$ be a matrix of which the columns correspond to the gradient vectors $\nabla_{\mathbf{x}_i} \mathcal{O}^p(A_{ij} = 1, \mathbf{x}_i, \mathbf{x}_j)$ (one column for each $j \in \Gamma(i)$), and let $\mathbf{F}_i^0 \in \mathbb{R}^{d \times |\Gamma(i)|}$ be the matrix of which the columns correspond to the gradient vectors $\nabla_{\mathbf{x}_i} \mathcal{O}^p(A_{ij} = 0, \mathbf{x}_i, \mathbf{x}_j)$ (also one column for each $j \in \Gamma(i)$). Moreover, let $\mathbf{b}_i \in \{1, -1\}^{|\Gamma(i)|}$ be a vector with a dimension corresponding to each of the neighbors $j \in \Gamma(i)$ of i , with value equal to 1 if that neighbor is a neighbor of i' and equal to -1 if it is a neighbor of i'' after splitting i . Then the gradient Equation (4) can be written more concisely and transparently as follows:

$$\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) = \frac{1}{2} (\mathbf{F}_i^1 - \mathbf{F}_i^0) \mathbf{b}_i.$$

The aim of FONDUE-NDA is to identify node splits for which the embedding quality improvement is maximal. As argued above, we propose to approximately quantify this by means of a first order approximation, by considering the two-norm squared of this gradient, namely by maximizing $\|\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si})\|^2$. Denoting $\mathbf{M}_i = (\mathbf{F}_i^1 - \mathbf{F}_i^0)^\top (\mathbf{F}_i^1 - \mathbf{F}_i^0)$ (and recognizing that $\mathbf{b}_i^\top \mathbf{b}_i = |\Gamma(i)|$ is independent of $\mathbf{b}_i$), FONDUE-NDA can thus be formalized in the following compact form:

$$\operatorname{argmax}_{i, \mathbf{b}_i \in \{1, -1\}^{|\Gamma(i)|}} \frac{\mathbf{b}_i^\top \mathbf{M}_i \mathbf{b}_i}{\mathbf{b}_i^\top \mathbf{b}_i}. \quad (5)$$

Note that $\mathbf{M}_i \succeq \mathbf{0}$ for all nodes and all splits, such that this is an instance of Boolean quadratic maximization problem [25,26]. This problem is NP-hard, thus it requires further approximations to ensure tractability in practice.

3.3.2. Additional Heuristics for Enhanced Scalability

In order to efficiently search for best split on a given node, we developed two approximation heuristics.

First, we randomly split the neighborhood $\Gamma(i)$ into two and evaluate the objective (Equation (5)). Repeat the randomization procedure for a fixed number of times, pick the split that gives the best objective value as output.

Second, we find the eigenvector $\mathbf{v}$ that corresponds to the largest absolute eigenvalue of matrix $\mathbf{M}_i$. Sort the element in vector $\mathbf{v}$ and assigning top k corresponding nodes to $\Gamma(i')$ and the rest to $\Gamma(i'')$. Evaluating the objective value for $k = 1 \dots |\Gamma(i)|$ and pick the best split.

Finally, we combine these two heuristics and use the split that gives the best objective Equation (5) as the final split of the node i .

3.3.3. FONDUE-NDA Using CNE

We now apply FONDUE-NDA to conditional network embedding (CNE). CNE proposes a probability distribution for network embedding and finds a locally optimal embedding by maximum likelihood estimation. CNE has objective function:

$$\begin{aligned} \mathcal{O}(\mathcal{G}, \mathbf{X}) &= \log(P(\mathbf{A}|\mathbf{X})) \\ &= \sum_{i,j:A_{ij}=1} \log P_{ij}(A_{ij} = 1|\mathbf{X}) + \sum_{i,j:A_{ij}=0} \log P_{ij}(A_{ij} = 0|\mathbf{X}). \end{aligned} \quad (6)$$

Here, the link probabilities P_{ij} conditioned on the embedding are defined as follows:

$$P_{ij}(A_{ij} = 1|\mathbf{X}) = \frac{P_{A,ij} \mathcal{N}_{+,\sigma_1}(\|\mathbf{x}_i - \mathbf{x}_j\|)}{P_{A,ij} \mathcal{N}_{+,\sigma_1}(\|\mathbf{x}_i - \mathbf{x}_j\|) + (1 - P_{A,ij}) \mathcal{N}_{+,\sigma_2}(\|\mathbf{x}_i - \mathbf{x}_j\|)},$$

where $\mathcal{N}_{+,\sigma}$ denotes a half-normal distribution [27] with spread parameter σ , $\sigma_2 > \sigma_1 = 1$, and where $P_{A,ij}$ is a prior probability for a link to exist between nodes i and j as inferred from the degrees of the nodes (or based on other information about the structure of the network [28]). First, we derive the gradient:

$$\nabla_{\mathbf{x}_i} \mathcal{O}(\mathcal{G}, \mathbf{X}) = \gamma \sum_{j \neq i} (\mathbf{x}_i - \mathbf{x}_j) (P(A_{ij} = 1|\mathbf{X}) - A_{ij}) = \mathbf{0},$$

where $\gamma = \frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}$. This allows us to further compute gradient

$$\nabla_{\delta_i} \mathcal{O}(\hat{\mathcal{G}}_{si}, \hat{\mathbf{X}}_{si}) = -\frac{\gamma}{2} \begin{pmatrix} \vdots & \mathbf{x}_i - \mathbf{x}_j & \vdots \end{pmatrix} \mathbf{b}_i$$

Thus, the Boolean quadratic maximization problem has form:

$$\operatorname{argmax}_{i, \mathbf{b}_i \in \{1, -1\}^{|\Gamma_i|}} \frac{\mathbf{b}_i^\top \sum_{k, l \in \Gamma(i)} (\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_l)^\top \mathbf{b}_i}{\mathbf{b}_i^\top \mathbf{b}_i}. \quad (7)$$

3.4. FONDUE-NDD

Using the inductive bias for the NDD problem, the goal is to minimize the embedding cost after merging the duplicate nodes in the graph (Equation (2)). This is motivated by the fact that natural networks tend to be modeled using NE methods, better than corrupted (duplicate) networks, thus their embedding cost should be lower. Thus, merging (or contracting) duplicate nodes (nodes that refer to the same entity) in a duplicate graph $\hat{\mathcal{G}}$ would result in a contracted graph $\hat{\mathcal{G}}_c$ that is less corrupt (resembling more a “natural” graph), thus with a lower embedding cost.

Contrary to NDA, NDD is more straightforward, as it does not deal with the problem of reassigning the edges of the node after splitting, but rather simply determining the duplicate nodes in a duplicate graph. FONDUE-NDD applied on $\hat{\mathcal{G}}$, aims to find duplicate node-pairs in the graph to combine them into one node by reassigning the union of their edges, which would result in contracted graph $\hat{\mathcal{G}}_c$.

Using NE methods, FONDUE-NDD aims to iteratively identify a node-pair $\{i, j\} \in \hat{\mathcal{V}}_{cand}$, where $\hat{\mathcal{V}}_{cand}$ is the set of all possible candidate node-pairs, that if merged together to form one node i_m , would result in the smallest cost function value among all the other node-pairs. Thus, problem 6 can be further rewritten as:

$$\operatorname{argmin}_{\{i, j\} \in \hat{\mathcal{V}}_{cand}} \mathcal{O}(\hat{\mathcal{G}}_{c_{ij}}, \hat{\mathbf{X}}_{c_{ij}}), \quad (8)$$

where $\hat{\mathcal{G}}_{c_{ij}}$ is a contracted graph from $\hat{\mathcal{G}}$ after merging the node-pair $\{i, j\}$, and $\hat{\mathbf{X}}_{c_{ij}}$ its respective embeddings.

Trying this for all possible node-pairs in the graph is an intractable solution. It is not obvious what information could be used to approximate Equation (8), thus we approach the problem simply by randomly selecting node-pairs, merging them, observing the values of the cost function, and then ranking the result. The lower the cost score, the more likely that those merged nodes are duplicates.

Lacking a scalable bottom-up procedure to identify the best node pairs, in the experiments our focus will be on evaluation whether the introduced criterion for merging is indeed useful to identify *whether* node pairs appear to be duplicates.

FONDUE-NDD Using CNE

Similarly to the previous section, we proceed by applying CNE as a network embedding method, the objective function of FONDUE-NDD is thus the one of CNE evaluated on the tentatively deduplicated graph after attempting a merge:

$$\begin{aligned} \mathcal{O}(\hat{\mathcal{G}}_{c_{ij}}, \hat{\mathbf{X}}_{c_{ij}}) &= \log(P(\hat{\mathbf{A}}_{c_{ij}} | \hat{\mathbf{X}}_{c_{ij}})) = \sum_{k, l: \hat{\mathbf{A}}_{c_{ij}, kl} = 1} \log \left(1 + \frac{\sigma_1}{\sigma_2} \frac{1 - P_{kl}}{P_{kl}} \exp \left(\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} \right) \frac{d_{kl}^2}{2} \right) \right) \\ &+ \sum_{k, l: \hat{\mathbf{A}}_{c_{ij}, kl} = 0} \log \left(1 + \frac{\sigma_2}{\sigma_1} \frac{P_{kl}}{1 - P_{kl}} \exp \left(\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2} \right) \frac{d_{kl}^2}{2} \right) \right), \end{aligned} \quad (9)$$

with the link probabilities P_{kl} conditioned on the embedding are defined as follows:

$$P_{kl}(\hat{\mathbf{A}}_{c_{ij}, kl} = 1 | \mathbf{X}) = \frac{P_{\hat{\mathbf{A}}_{c_{ij}, kl}} \mathcal{N}_{+, \sigma_1}(\|\mathbf{x}_k - \mathbf{x}_l\|)}{P_{\hat{\mathbf{A}}_{c_{ij}, kl}} \mathcal{N}_{+, \sigma_1}(\|\mathbf{x}_k - \mathbf{x}_l\|) + (1 - P_{\hat{\mathbf{A}}_{c_{ij}, kl}}) \mathcal{N}_{+, \sigma_2}(\|\mathbf{x}_k - \mathbf{x}_l\|)}.$$

Similarly to Section 3.3.3, $\mathcal{N}_{+, \sigma}$ denotes a half-Normal distribution with spread parameter σ , $\sigma_2 > \sigma_1 = 1$, and where $P_{\hat{A}_{c_{ij}, kl}}$ is a prior probability for a link to exist between nodes k and l as inferred from the network properties.

4. Experiments

In this section, we investigate quantitatively and qualitatively the performance of FONDUE on both semi-synthetic and real-world datasets, compared to state-of-the-art methods tackling the same problems. In Section 4.1, we introduce and discuss the different datasets used in our experiments, in Section 4.2 we discuss the performance of FONDUE-NDA, and FONDUE-NDD in Section 4.3. Finally, in Section 4.4, we summarize and discuss the results. All code used in this section is publicly available from the GitHub repository <https://github.com/aida-ugent/fondue>, accessed on 20 October 2021.

4.1. Datasets

One main challenge for assessing the evaluation of disambiguation tasks is the scarcity of availability of ambiguous (contracted) graph datasets with reliable ground truth. Furthermore, other studies that focus on ambiguous node identification often do not publish their heavily processed dataset (e.g., DBLP datasets [16]), which makes it harder to benchmark different methods. Thus, to simulate data corruption in real world datasets, we opted to create a contracted graph given a source graph, and then use the latter as ground truth to assess the accuracy of FONDUE compared to other baselines. To do so, we used a simple approach for node contraction, for both NDA (Section 4.2.1) and NDD (Section 4.3.1). Below, in Table 1 we list the details of the different datasets used after post-processing in our experiments.

Additionally, we also use real-world networks containing ambiguous and duplicate nodes, mainly part of the PubMed collaboration network, analyzed in Appendix A. The PubMed data are released in independent issues, so to build a connected network from the PubMed data, we select issues that contain ambiguous and duplicate nodes. We then select the largest connected component of that network. One main limitation to this dataset is that not every author has an associated Orcid ID, which affects the false positive and false negative labels in the network (author names that might be ambiguous would be ignored). This is further highlighted in the subsequent sections.

4.2. Node Disambiguation

In this section, we investigate the following questions: (Q_1) Quantitatively, how does our method perform in identifying ambiguous nodes compared to the state-of-the-art and other heuristics? (Section 4.2.2); (Q_2) Qualitatively, how reliable is the quality of the detected ambiguous nodes compared to other methods when applied to real world datasets? (Section 4.2.3); (Q_3) Quantitatively, how does our method perform in terms of splitting the ambiguous nodes? (Section 4.2.4); (Q_4) How does the behavior of the method change when the degree of contraction of a network varies? (Section 4.2.5); (Q_5) Does the proposed method scale? (Section 4.2.6); (Q_6) Quantitatively, how does our method perform in terms of node deduplication? (Section 4.3.1).

4.2.1. Data Processing

Before conducting the experiments, the processing of the data to generate semi-synthetic networks was needed. This was completed by contracting each of the thirteen datasets mentioned in Table 2. More specifically, for each network $\mathcal{G} = (V, E)$, a graph contraction was performed to create a contracted graph $\hat{\mathcal{G}} = (\hat{V}, \hat{E})$ (ambiguous) by randomly merging a fraction r of total number of nodes, to create a ground truth to test our proposed method. This is completed by first specifying the fraction of the nodes in the graph to be contracted ($r \in \{0.001, 0.01, 0.1\}$), and then sampling two sets of vertices, $\hat{V}^i \subset \hat{V}$ and $\hat{V}^j \subset \hat{V}$, such that $|\hat{V}^i| = |\hat{V}^j| = \lfloor r \cdot |\hat{V}| \rfloor$ and $\hat{V}^i \cap \hat{V}^j = \emptyset$. Then, every vertex $v_j \in \hat{V}^j$ is merged with the corresponding vertex $v_i \in \hat{V}^i$ by reassigning the links connected

to v_j to v_i and removing v_j from the network. The node-pairs (v_i, v_j) later serve as ground truths. We have also tested the case where the set of the candidate contracted vertices have no common neighbors (instead of a uniform selection at random). This mimics some types of social networks where two authors that share the same name, often their ego-networks do not intersect. Further analysis of the PubMed dataset Table 1, revealed that none of the ambiguous nodes shared edges with the neighbors of another ambiguous node.

We have tested the performance of FONDUE-NDA, as well as that of the competing methods listed in the following section, on fourteen different datasets listed in Table 1, with their properties shown in Table 2.

Table 1. The different datasets used in our experiments (Section 4.1).

Datasets	Description
FB-SC	Facebook Social Circles network [29] Consists of anonymized friends list from Facebook.
FB-PP	Page-Page graph of verified Facebook pages [29]. Nodes represent official Facebook pages while the links are mutual likes between pages.
email	Anonymized network generated using email data from a large European research institution modeling the incoming and outgoing email exchange between its members [29].
STD	A database network of the Computer Science department of the University of Antwerp that represent the connections between students, professors and courses [30].
PPI	A subnetwork of the BioGRID Interaction Database [31], that uses PPI network for Homo Sapiens.
lesmis	A network depicting the coappearance of characters in the novel Les Misérables [32].
netscience	A coauthorship network of scientists working on network theory and experiments [29].
polbooks	Network of books about US politics, with edges between books representing frequent copurchasing of books by the same buyers. http://www-personal.umich.edu/~mejn/netdata/ , accessed on 20 October 2021
CondMat	Collaboration network of Arxiv Condensed Matter Physics [33]
GrQc	Collaboration network of Arxiv General Relativity [33]
HepTh	Collaboration network of Arxiv Theoretical High Energy Physics [33]
CM03	Collaboration network of Arxiv Condensed Matter till 2003 [33]
CM05	Collaboration network of Arxiv Condensed Matter till 2005 [33]
PubMed	Collaboration network extracted from the PubMed database (analyzed in Appendix A), containing 2122 nodes with ground truth of 31 ambiguous nodes (6 of which maps to more than 2 entities) and 1 duplicate node [1]

Table 2. Various properties of each semi-synthetic network used in our experiments.

	fb-sc	fb-pp	email	lesmis	polbooks	STD	
# Nodes	4039	22,470	986	77	105	395	
# Edges	88,234	170,823	16,064	254	441	3423	
Avg degree	43.7	15.2	32.6	6.6	8.4	17.3	
Density	1.1×10^2	6.8×10^4	3.3×10^2	8.7×10^2	8.1×10^2	4.4×10^2	
	ppi	netscience	GrQc	CondMat	HepTh	CM05	CM03
# Nodes	3852	379	4158	21,363	8638	36,458	27,519
# Edges	37,841	914	13,422	91,286	24,806	171,735	116,181
Avg degree	19.6	4.8	6.5	8.5	5.7	9.4	8.4
Density	5.1×10^3	1.3×10^2	1.6×10^3	4.0×10^4	6.6×10^4	2.6×10^4	3.1×10^4

4.2.2. Quantitative Evaluation of Node Identification

In this section, we focus on answering Q_1 , namely, given a contracted graph, FONDUE-NDA aims to identify the list of contracted (ambiguous) nodes present in it. We first discuss the datasets used in the experiments in the following section.

Baselines. As mentioned earlier in Section 1, most entity disambiguation methods in the literature focus on the task of re-assigning the edges of an already predefined set of ambiguous nodes, and the process of identifying these nodes in a given non-attributed network, is usually overlooked. Thus, there exists very few approaches that tackle the latter case. In this section, we compare FONDUE-NDA with three different competing approaches that focus on the identification task, one existing method, and two heuristics.

Normalized-Cut (NC) The work of [16] comes close to ours, as their method also aims to identify ambiguous nodes in a given graph by utilizing Markov clustering to cluster an ego network of a vertex u with the vertex itself removed. NC favors the grouping that gives small cross-edges between different clusters of u 's neighbors. The result is a score reflecting the quality of the clustering, using normalized-cut (NC):

$$NC = \sum_{i=1}^k \frac{W(C_i, \overline{C_i})}{W(C_i, C_i) + W(C_i, \overline{C_i})},$$

with $W(C_i, C_i)$ as the sum of all the edges within cluster C_i , $W(C_i, \overline{C_i})$ the sum of the for all the edges between cluster C_i and the rest of the network $\overline{C_i}$, and k being the number of clusters in the graph. Although [17] also worked on identifying nodes based on topological features, their method (which is not publicly available) performed worse in all the cases when compared to [16], so we only chose the latter as a competing baseline.

Connected-Component Score (CC) We also include another baseline, connected-component score (CC), relying on the same approach used in [16], with a slight modification. Instead of computing the normalized cut score based on the clusters of the ego graph of a node, we account for the number of connected components of a node's ego graph, with the node itself removed.

Degree Finally, we use node degree as a baseline. As contracted nodes usually tend to have a higher degree, by inheriting more edges from combined nodes, degree is a simple predictor for the node ambiguity.

Evaluation Metric. FONDUE-NDA ranks nodes according to their calculated ambiguity score (how likely is that node to be ambiguous). The same process goes for NC and CC. At first glance, the evaluation can be approached from a binary classification perspective, by considering the top X ranked nodes as ambiguous (where X is the actual number true-positive), and, thus, we can use the usual metrics for binary classification, such as F1-score, precision, recall and AUC. However, this requires knowing beforehand the number of true-positive, i.e., the number of actual ambiguous nodes (or setting a clear cutoff value), which is only possible in labeled datasets and controlled experiments. In real world settings, if FONDUE-NDA is to be used to detect ambiguous nodes in unlabeled networks, practical application is rather more restricted, as it is more useful to have relevant nodes (ambiguous) ranked more highly than non-relevant nodes. Thus, it is necessary to extend the traditional binary classification evaluation methods, that are based on binary relevance judgments, to more flexible graded relevance judgments, such as, for example, cumulative gain, which is a form of graded precision, as it is identical to the precision when rating scale is binary. However, as our datasets are highly imbalanced by nature, mainly because ambiguous nodes are by definition a small part of the network, a better take on the cumulative gain metric is needed. Hence, we employ the normalized discounted gain to evaluate our method, alongside the traditional binary classification methods listed above. Below, we detail each metric.

Precision The number of correctly identified positive results divided by the number of all positive results

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall The number of correctly identified positive results divided by the number of all positive samples

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score It is the weighted average of the precision where an F1 score reaches its best value at 1 and worst score at 0.

$$F1 = \frac{2 * \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

Note that, due to the fact that in the binary classification case, the number of false positive is equal to the number of false negative, the value of the recall, precision and F1-score will be the same.

Area Under the ROC curve (AUC) A ROC curve is a 2D depiction of a classifier performance, which could be reduced to a single scalar value, by calculating the value under the curve (AUC). Essentially, the AUC computes the probability that our measure would rank a randomly chosen ambiguous node (positive example), higher than a randomly chosen non-ambiguous node (negative example). Ideally, this probability value is 1, which means our method has successfully identified ambiguous nodes 100% of the time, and the baseline value is 0.5, where the ambiguous and non-ambiguous nodes are indistinguishable. This accuracy measure has been used in other works in this field, including [16], which makes it easier to compare to their work.

Discounted Gain (DCG) The main limitation of the previous method, as we discussed earlier, is inability to account for graded scores, but rather only binary classification. To account for this, we utilize different cumulative gain based methods. Given a search result list, cumulative gain (CG) is the sum of the graded relevance values of all results.

$$CG = \sum_{i=1}^n \text{relevance}_i$$

On the other hand, DCG [34] takes position significance into account, and adds a penalty if a highly relevant document is appearing lower in a search result list, as the graded relevance value is logarithmically reduced proportionally to the position of the result. Practically, it is the sum of the true scores ranked in the order induced by the predicted scores, after applying a logarithmic discount. The higher the better is the ranking.

$$DCG = \sum_{i=1}^n \frac{\text{relevance}_i}{\log_2(i+1)}$$

Normalized Discounted Gain (NDCG) It is commonly used in the information retrieval field to measure effectiveness of search algorithms, where highly relevant documents being more useful if appearing earlier in search result, and more useful than marginally relevant documents which are better than non-relevant documents. It improves upon DCG by accounting for the variation of the relevance, and providing a proper upper and lower bounds to be averaged across all the relevance scores. Thus, it is computed by summing the true scores ranked in the order induced by the predicted scores, after applying a logarithmic discount, then dividing by the best possible score ideal DCG (IDCG, obtained for a perfect ranking) to obtain a score between 0 and 1.

$$NDCG = \frac{NDCG}{IDCG}$$

Evaluation pipeline. We first perform network contraction on the original graph, by fixing the ratio of ambiguous nodes to r . We then embed the network using CNE, and compute the disambiguation measure of FONDUE-NDA (Equation (7)), as well as the baseline measures for each node. Then, the scores yield by the measures are compare to the ground truth (i.e., binary labels indicating whether a node is a contracted node). This is completed for 3 different values of $r \in \{0.001, 0.01, 0.1\}$. We repeat the processes 10 times using a different random seed to generate the contracted network and average the scores. For the embedding configurations, we set the parameters for CNE to $\sigma_1 = 1$, $\sigma_2 = 2$, with dimensionality limited to $d = 8$.

Results. are illustrated in Figure 3 and shown in detail in Table 3 focusing on NDCG mainly for being a better measure for assessing the ranking performance of each method. FONDUE-NDA outperforms the state-of-the-art method, as well as non-trivial baselines in terms of NDCG in most datasets. It is also more robust with the variation of the size of the network, and the fraction of the ambiguous nodes in the graph. NC seems to struggle to identify ambiguous nodes for smaller networks (Table 2). Additionally, as we tested against multiple network settings, with randomly uniform contraction (randomly selecting a node-pair and merging them together), or a conditional contraction (selecting a node pair that do not share common neighbors to mimic realistically collaboration networks), we did not observe any significant changes in the results.

Table 3. Performance evaluation (NDCG) on multiple datasets for our method compared with other baselines, for two different contraction methods. Note that for some datasets with small number of nodes, we did not perform any contraction for 0.001 as the number of contracted nodes in this case is very small, thus we replaced the values for those methods by “—”.

Ambiguity Rate		10%				1%				0.1%			
Method		FONDUE-NDA	NC	CC	Degree	FONDUE-NDA	NC	CC	Degree	FONDUE-NDA	NC	CC	Degree
Randomly Uniform Contraction	fb-sc	0.954	0.962	0.768	0.776	0.767	0.875	0.569	0.423	0.679	0.535	0.272	0.199
	fb-pp	0.899	0.825	0.821	0.804	0.649	0.532	0.528	0.511	0.374	0.268	0.268	0.253
	email	0.783	0.661	0.619	0.704	0.529	0.305	0.264	0.310	—	—	—	—
	student	0.778	0.664	0.568	0.652	0.396	0.328	0.235	0.257	—	—	—	—
	lesmis	0.906	0.570	0.499	0.622	—	—	—	—	—	—	—	—
	polbooks	0.972	0.604	0.534	0.698	1.000	0.310	0.267	0.318	—	—	—	—
	ppi	0.759	0.670	0.724	0.741	0.420	0.353	0.381	0.387	0.194	0.138	0.147	0.151
	netscience	0.886	0.784	0.731	0.721	0.508	0.378	0.323	0.288	—	—	—	—
	GrQc	0.857	0.805	0.796	0.768	0.603	0.447	0.437	0.415	0.249	0.195	0.184	0.168
	CondMat	0.864	0.855	0.843	0.816	0.601	0.553	0.543	0.520	0.367	0.278	0.269	0.255
	HepTh	0.860	0.798	0.823	0.796	0.582	0.466	0.494	0.470	0.325	0.201	0.224	0.208
	cm05	0.884	0.873	0.859	0.827	0.627	0.590	0.582	0.545	0.471	0.312	0.307	0.288
	cm03	0.888	0.869	0.852	0.823	0.635	0.577	0.562	0.534	0.335	0.297	0.281	0.272
no common neighbors	fb-sc	0.953	0.989	0.768	0.764	0.730	0.933	0.591	0.418	0.399	0.665	0.321	0.172
	fb-pp	0.895	0.826	0.820	0.801	0.650	0.532	0.529	0.510	0.389	0.266	0.267	0.253
	email	0.676	0.696	0.625	0.604	0.303	0.319	0.288	0.256	—	—	—	—
	student	0.659	0.726	0.531	0.587	0.368	0.447	0.201	0.229	—	—	—	—
	lesmis	0.755	0.591	0.498	0.486	—	—	—	—	—	—	—	—
	polbooks	0.981	0.620	0.544	0.696	1.000	0.268	0.420	0.363	—	—	—	—
	ppi	0.725	0.673	0.721	0.700	0.398	0.352	0.381	0.373	0.166	0.139	0.147	0.144
	netscience	0.877	0.797	0.714	0.705	0.622	0.372	0.304	0.290	—	—	—	—
	GrQc	0.861	0.806	0.794	0.766	0.580	0.445	0.435	0.416	0.280	0.197	0.183	0.173
	CondMat	0.863	0.855	0.843	0.815	0.585	0.554	0.542	0.516	0.317	0.274	0.273	0.257
	HepTh	0.856	0.798	0.824	0.796	0.581	0.467	0.494	0.480	0.285	0.204	0.212	0.213
	cm05	0.883	0.874	0.858	0.825	0.633	0.591	0.582	0.543	0.414	0.310	0.312	0.289
	cm03	0.884	0.869	0.853	0.822	0.651	0.577	0.561	0.533	0.439	0.295	0.279	0.271

Figure 3. Bar plots for the different metrics (higher is better) for each dataset listed in Table 3, for each of the 4 measures, FONDUE-NDA, Degree, NC, CC, for network with 10% uniformly contracted nodes. More detailed results are listed in Appendix B.

4.2.3. Qualitative Evaluation of Nodes Identification

Now that we have demonstrated that FONDUE-NDA outperforms current state-of-the-art methods for identifying ambiguous nodes, in this section we investigate Q_3 , the quality of the results produced by FONDUE-NDA and how do they compare to other competing methods. Additionally, we investigate the question, whether nodes with a higher ambiguity number (nodes that map to a larger number of entities) are more easily identified as ambiguous compared to nodes with lower ambiguity number (that map to a lower number of entities). To do so, semi-synthetic datasets do not constitute valid sources, and we need to rely on real-world dataset to assess the quality of the ground truth of the ambiguous nodes. We collected data from the National Center for Biotechnology Information that provides the PubMed datasets, that comprises citations for biomedical literature from MEDLINE, life science journals, and online books, that may include links to full-text content from PubMed Central and publisher websites. Snippets of those data are released periodically and can be used to build author–author collaboration networks. Us-

ing Orcid ID, a persistent digital identifier, usually provided in the metadata of the datasets to distinguish ambiguous nodes, we can build reliable ground truth for our qualitative experiment and investigate the top ranked nodes by FONDUE-NDA, and other baselines.

Datasets. PubMed datasets are publicly available, and updated regularly. Few data snippets are selected to extract author information and build an author–author collaboration network based on the metadata included in the datasets. The biggest connected component of the network is then selected. To assess the ambiguity of the network, we also extract the Orcid ID, if available, for each author in the network. If more than Orcid ID is associated with one author, the author-name is labeled as ambiguous. Note that not all authors have their Orcid ID listed in the datasets, which might cause a possible degradation in the quality of the labeled datasets, as some author-names might be ambiguous, but due to their Orcid ID not being recorded, they will not be accounted for in the labels, thus affecting the final results.

Metrics. Similarly to Section 4.2.2, we first take the binary classification approach, where we consider the top 31 ranked nodes to be classified as ambiguous (as the network contains 31 ambiguous nodes). The same goes for NC. We use the same metrics from Section 4.2.2. Additionally, to assess the quality of the ranked nodes, we utilize the number of true positives (TP) (nodes that are correctly identified as ambiguous) and false positive (FP) (nodes that are incorrectly identified as ambiguous). We also consider NDCG for graded relevance.

Results. After building the PubMed collaboration network, that contains 2122 nodes with 31 ambiguous nodes (6 of which are names that refer to more than 2 authors), it is embedded using CNE, and we compute the score measure of FONDUE-NDA (Equation (7)), as well as the baseline measures for each node. As explained earlier in Section 4.2.2, FONDUE-NDA ranks the nodes by ambiguity score. For comparison, we use the work of [16] as a baseline for qualitative comparison. Table 4 shows the performance of FONDUE-NDA against NC, for AUC, TP, FP, and F1-score. FONDUE-NDA clearly outperforms NC for binary classification of ambiguous nodes. This result is also highlighted when inspecting further the results in Table 5, that list the top 10 classified nodes by each method, with FONDUE-NDA successfully classifying 90% of the author names. Note that the results are quite intuitive, and conform with our findings in the earlier analysis of the dataset in Appendix A, as authors with Asian names are more likely to share common names, due to shorter name length, and simplified transcription (from Mandarin to English for example).

We also investigated the ranks of nodes that maps to more than 2 entities (highlighted with an asterisk in both tables). Again FONDUE-NDA outperforms NC, and ranks 3 out of 6 highly ambiguous names in the top 10.

This confirms the results obtained in Section 4.2.2, as FONDUE-NDA outperforms NC in as a state-of-the-art method for identifying ambiguous nodes.

Table 4. Performance of FONDUE-NDA compared to NC for the PubMed dataset containing 31 ambiguous author-names of which 6 are associated with more than 2 Orcid IDs (highly ambiguous). NDCG* reflects the ranking score of those highly ambiguous nodes.

	FONDUE-NDA	NC [16]
AUC	0.971	0.944
TP	17	11
FP	14	22
F1-score	0.55	0.35
NDCG	0.877	0.681
NDCG*	0.743	0.38

Table 5. The top 10 ranked nodes of FONDUE-NDA and NC, against the ground truth (1 if the node is ambiguous and 0 otherwise). FONDUE-NDA correctly classifies 9 out of the 10 nodes as ambiguous. Names with * are highly ambiguous, referring to more than 2 Orcid IDs.

NC Rank	Ambiguous	FONDUE-NDA Rank	Ambiguous
Jie_Li	0	Tao_Wang *	1
Lei_Liu	0	Wei_Zhang	1
Jiawei_Wang	0	Jing_Li	1
Jun_Liu	0	Bin_Zhang *	1
Ni_Wang	0	Yan_Li	1
Xin_Liu	0	Rui_Li	0
Yao_Chen	0	Ying_Liu	1
Huanhuan_Liu	0	Feng_Li	1
Jun_Yan	0	Yang_Yang *	1
Lei_Chen	0	Ying_Sun	1

4.2.4. Quantitative Evaluation of Nodes Splitting

Following the identification of the ambiguous nodes, in this section, we focus on the task of node splitting, and answering Q_3 of how well does FONDUE-NDA perform when it comes to partitioning the set of edges into two separate ambiguous nodes. Simply put, given an ambiguous node v_i , we refer to node splitting the process of replacing this particular node with two different nodes v'_i, v''_i and re-assigning the edges of v_i , such that $\Gamma(v'_i) \cup \Gamma(v''_i) = \Gamma(v_i)$.

Baselines. For the node splitting task, the three baselines previously discussed in Section 4.2.2 are not immediately applicable. However, we adopt the Markov clustering (MCL) approach utilized in normalized cut measure for splitting. Namely, a splitting is given by the MCL clustering on the ego network of an ambiguous node, with the node itself removed.

Evaluation Metric. Given a list of ambiguous nodes, we evaluate the splitting given by FONDUE-NDA and MCL against the ground truth (node splitting according to the original network). This is quantified by computing the adjusted Rand index (ARI) score between FONDUE-NDA and the ground truth, as well as between MCL and the ground truth. ARI score is a similarity measure between two clusterings. ARI ranges between -1 and 1 , the higher the score the better the alignment between the two compared clusterings.

Pipeline. First we compute the ground truth. Then, for each ambiguous node, we evaluate the quality (based on ARI) of the split from FONDUE-NDA and MCL compared to the original partition. We repeat the experiments for three different contraction ratios $r \in \{0.001, 0.01, 0.1\}$ for each dataset. For each ratio, the experiment is repeated 3 times with different seeds.

Results. Despite outperforming NC in ambiguous node identification, FONDUE-NDA seems to underperform compared to MCL on nearly every dataset (Table 6). This shows that for node splitting, FONDUE-NDA is not adequately optimized for such task. Many factors contribute to the poor performance, mainly the quality of the embeddings which would lead to a poor performance for the objective function for edge assignment. Nonetheless, given the modular approach for our method, and for a complete framework for node identification and node splitting, we recommend using FONDUE-NDA for former task and MCL for the latter.

Table 6. Adjusted Rand index score for FONDUE-NDA and MCL.

Ambiguity Rate		10%		1%		0.1%	
	Method	FONDUE-NDA	MCL	FONDUE-NDA	MCL	FONDUE-NDA	MCL
Uniformly Random Contraction	fb-sc	0.582	0.781	0.781	0.829	0.947	0.936
	fb-pp	0.405	0.583	0.504	0.583	0.610	0.600
	email	0.122	0.248	0.214	0.276	—	—
	student	0.215	0.124	0.131	0.183	—	—
	lesmis	0.597	0.304	—	—	—	—
	polbooks	0.391	0.338	—	—	—	—
	ppi	0.033	0.120	0.055	0.115	0.338	0.321
	netscience	0.610	0.777	0.642	0.809	—	—
	GrQc	0.498	0.665	0.661	0.682	0.762	0.784
	CondMat	0.471	0.709	0.582	0.713	0.574	0.718
	HepTh	0.447	0.556	0.521	0.552	0.465	0.542
	cm05	0.471	0.722	0.674	0.732	0.488	0.741
	cm03	0.479	0.724	0.575	0.730	0.513	0.690
No Common Neighbors	fb-sc	0.609	0.952	0.887	0.985	0.862	0.998
	fb-pp	0.409	0.586	0.509	0.564	0.546	0.637
	email	0.136	0.315	0.278	0.413	—	—
	student	0.197	0.139	0.079	0.078	—	—
	lesmis	0.405	0.289	—	—	—	—
	polbooks	0.499	0.577	0.638	0.655	—	—
	ppi	0.036	0.139	0.067	0.151	0.331	0.399
	netscience	0.725	0.819	0.722	0.897	—	—
	GrQc	0.534	0.678	0.665	0.683	0.895	0.872
	CondMat	0.470	0.708	0.548	0.712	0.475	0.686
	HepTh	0.469	0.571	0.529	0.561	0.404	0.455
	cm05	0.466	0.726	0.549	0.777	0.450	0.723
	cm03	0.482	0.722	0.570	0.727	0.567	0.697

4.2.5. Parameter Sensitivity

In this section, we study the robustness of our FONDUE-NDA against different network settings. Mainly, how does the percentage of ambiguous nodes in a graph affect the node identification. In the previous experiment (Section 4.2.2), we have fixed the ratio of ambiguous nodes to $\{0.001, 0.01, 0.10\}$, we follow the same pipeline (generate, embed, evaluate for 10 different random seeds), for different ratios of ambiguous nodes. As listed in Table 3 FONDUE-NDA outperforms MCL and other baselines across nearly all networks with different contraction ratios. We also accounted for different ways the node contraction is conducted as specified in Section 4.2.1. As previously mentioned, the node contraction process is assumed to be a form of corruption of the data, i.e., in real-life (outside an evaluation setup). To simulate this corruption process (so as to generate semi-synthetic test data with known ground-truth), merged nodes were selected uniformly at random. Even though this does not guarantee that the network becomes unnatural in a way that suits the embedding objective function. However, this further strengthens our empirical results: FONDUE-NDA works even when the assumptions are not guaranteed to be satisfied. We also studied different ways to select nodes for contraction in generating semi-synthetic data: only selecting contracted nodes that have common neighbors. FONDUE-NDA consistently outperformed the baselines also with these different network contraction approaches in most datasets, as shown in Table 3 and Figure 3.

4.2.6. Execution Time Analysis

The runtime of FONDUE-NDA is linear $\mathcal{O}(n)$ on each node, and is trivially parallelizable. In Figure 4, we show the execution speed of FONDUE-NDA and baselines in node identification and splitting. FONDUE-NDA is faster than CC, and by nearly one order of magnitude than NC in most datasets. Note that FONDUE-NDA approximates Equation (5) by aggregating two different approximation heuristics (i.e., randomized sampling and eigenvector thresholding as listed in Section 3.3.2). Although the best results (in terms of NDCG and ARI) were obtained by the latter approximation. Thus, the runtime results reflect only the execution time of latter heuristic. This is listed in details in Table 7. All the experiments have been conducted on a Intel i7 – 7700K CPU 4.20GHz, running the Ubuntu 18.04 distribution of Linux, with 32GB of RAM.

Figure 4. Bar plots for the runtime performance in seconds (lower is better) for each dataset listed in Table 7, for each of the 3 measures, FONDUE-NDA, NC, CC for different percentage of contracted nodes, 0.1%, 1%, and 10%, respectively.

Table 7. Execution time (in seconds) comparison table for the different datasets averaged over 10 different experiments. The embedding time of the networks was not taken into account for the computation of the runtime results.

	10%			1%			0.1%		
	FONDUE-NDA	NC	CC	FONDUE-NDA	NC	CC	FONDUE-NDA	NC	CC
fb-sc	10.99	94.72	52.92	5.16	90.43	51.62	7.66	90.99	51.63
fb-pp	15.03	161.60	46.83	18.02	185.64	46.48	29.65	190.25	46.06
email	0.81	10.18	4.77	0.84	10.26	4.62	—	—	—
student	0.35	2.88	0.69	0.37	4.45	0.59	—	—	—
lesmis	0.14	0.21	0.03	—	—	—	—	—	—
polbooks	0.04	0.34	0.05	0.05	0.36	0.05	—	—	—
ppi	4.09	49.23	15.64	4.17	50.89	15.03	3.53	50.73	14.95
netscience	0.12	0.89	0.08	0.13	0.92	0.09	0.13	0.91	0.09
GrQc	1.53	13.07	2.06	1.66	13.36	2.07	1.69	13.29	2.04
CondMat	11.41	82.63	13.15	12.36	86.42	13.11	13.76	88.26	12.75
HepTh	3.47	28.04	2.85	3.65	27.22	2.81	4.00	27.68	2.77
cm05	23.04	143.23	30.02	24.69	144.98	25.12	26.01	137.92	25.51
cm03	16.10	116.26	17.82	20.33	137.57	17.12	18.96	124.25	17.32

4.3. Node Deduplication

4.3.1. Quantitative Evaluation of Node Deduplication

In this section we describe the details of the experimental setup used to tackle the NDD problem. Referring to the data listed in Table 1, we performed our experiments on three semi-synthetic datasets. We also used the PubMed dataset with metadata containing Orcid ID to also build a duplicate network with ground truth derived from the author names with common Orcid ID.

Evaluation Pipeline. As we indicated in Section 4.1, we perform post-processing on the graph data to conduct controlled experiments for comparison with other baselines. For the semi-synthetic datasets (Table 2), to simulate the data corruption process caused by the NDD problem, we perform node splitting on different datasets. For a given graph $G = (V, E)$ with $n = |V|$, we compute the embedding cost function of G . We then randomly choose one node i to split into two i' and i'' , i.e., adding a new node to the graph, which results in a corrupted (duplicate) graph $\hat{G} = (\hat{V}, \hat{E})$ with $n_d = |\hat{V}| = n + 1$.

We then randomly choose one node-pair from $\hat{V} \times \hat{V}$, and perform node contraction, by merging these 2 nodes such that we end up with 1 node containing all the edges of the previous 2, which results in contracted graph $\hat{G}_c = (\hat{V}_c, \hat{E}_c)$ with $|\hat{V}_c| = n$. We then compute the embedding cost function of G'_c . We repeat the process by choosing a different random node-pair from V_c and computing the resulting embedding cost function, 99 times. Lastly, we compute the embedding cost function after merging the duplicate node pair $\{i', i''\}$. We compare the value of the objective cost function with that of graph G , and display the ranking in Table 8. Although the process of data corruption seems simple, there

are many parameter variables at hand that can affect it. Thus, we introduce few parameters to better model node splitting:

- Edge distribution: We employ 2 different ways to reassign the edges, from 1 node to 2 different nodes, either by randomly distributing the edges with at least 1 edge per node, or by ensuring equal distribution for each one;
- Minimum degree: We only choose to split nodes with a degree larger than a specified minimum;
- Overlap: We specify if there is an overlap in the edge reassignment for the different nodes, i.e., percentage of common edges for each node.

Datasets. As indicated earlier, we have used the three semi-synthetic datasets (Table 2), *lesmis*, *polbooks*, and *netscience*, mainly for their relatively small size, additionally we tested FONDUE-NDD on part of the PubMed dataset (described earlier in Section 4.2.3), a network of 2122 nodes, including one duplicate node.

Baselines. As all competing methods [22,23] require additional labels and attributes for NDD, so we opted to use a simple baseline to assess the performance of FONDUE-NDD, namely the L2-norm of the embedding distance between the candidate duplicate node-pairs $\{i', i''\}$ using CNE (referred to as ED).

$$ED_{\{i', i''\}} = ||\mathbf{x}_{i'} - \mathbf{x}_{i''}||_2$$

Metric. As described in the previous section, we compute the value of the objective function after the re-embedding for the different merged nodes. We then rank the different node pairs by their value. We use that rank as a metric to predict whether our approach can successfully predict which node-pair is a duplicate. The best value would be 1, which means that 100% of the time, FONDUE-NDD is able to identify the node pairs, as the cost of the re-embedding is the lowest.

Table 8. Results of the controlled experiments for each dataset. The average ranking of objective cost function over 100 different trials. The lower, the better. Bold numbers indicate that the difference in averages are significant ($p \leq 0.05$).

Edge Distribution	Minimum Degree	Edge Overlap	Lemis		Polbooks		Netscience	
			FONDUE-NDD	ED	FONDUE-NDD	ED	FONDUE-NDD	ED
Balanced	None	0%	18.775	18.2	30.025	17.85	6.975	4.2
		20%	15.55	8.75	22.475	10.65	3.9	2.825
		30%	14.125	9.35	20.4	8.5	3.225	1.775
	Graph Average	0%	10	15.806	17.676	20.941	5.325	5.7
		20%	10.167	11.083	11.471	12.941	2.775	3.025
		30%	8.611	9.333	10.794	11.176	2.725	2.625
	2x Graph Average	0%	3.857	24.857	5.727	23.364	3.471	5.029
		20%	5	17.429	3.818	15.909	1.735	3.412
		30%	2.857	13.429	2.545	14.091	1.735	3.206
Unbalanced	None	0%	25.9	22.75	36.425	26.325	6.9	2.85
		20%	16.75	10.75	25.875	10	3.75	2.55
		30%	16.3	10.525	22.875	12.075	3.65	2.775
	Graph Average	0%	13.5	17.306	27	23.176	5.125	5.075
		20%	12.417	13.278	13.029	11.706	3.1	3.15
		30%	12.944	12.167	14.265	12.029	3.025	2.675
	2x Graph Average	0%	18.143	40.143	11.545	22.545	5.735	8.147
		20%	9.429	18.429	8.364	16.182	2.118	3.5
		30%	5.714	19.286	7	12.182	1.735	2.794

Results. The results in Table 8, represent the average ranking of objective cost function over 100 different trial. We ran a 2-side Fisher test to test if the differences between the averages for the two methods are significantly different ($p < 0.05$), and the averages are highlight

in bold when it is the case. The results show that for high degree nodes (higher than the average), FONDUE-NDD outperforms ED, but its performance degrades for low degree nodes. Additionally, the more connected a corrupted node is, the better the improvement of the objective function of the recovered network compared to that of the corrupted network. This shows that some parameters identified in the previous section plays a large role in the identification of the duplicate nodes using FONDUE-NDD. Overall the intuition behind FONDUE-NDD is highlighted in the results of the experiments. For the PubMed dataset, we find that the average rank is equal to 4 out of 100, while ED ranked 6th. This also confirms the result to semi-synthetic data, as the degree of the duplicate node was above the average of the graph.

Execution time. As we do not account for the time of embedding of the initial duplicate network as part of execution time for FONDUE-NDD, the baseline ED has an execution time of 0, as it is directly derived from the embedding of the duplicate graph. FONDUE-NDD performs additional repeated uniform random node contraction then embedding, as specified in the pipeline section, thus the execution time for FONDUE-NDD varies depending on the size of the network and the number of embeddings executed. Results are shown in Table 9.

Table 9. Runtime for FONDUE-NDD in seconds, for 100 iterations (contracting each time a different random node pair and computing its embeddings).

Dataset	Execution Time (Seconds)
lesmis	609.34
polbooks	712.29
netscience	1198.63
pubmed	3474.82

4.4. Discussions

Despite its state-of-the-art performance in identifying ambiguous nodes (Section 4.2.2), FONDUE-NDA's node splitting functionality falls short compared to that of MCL (Section 4.2.4). Nonetheless, we argue that FONDUE-NDA's main feature is to facilitate the identification of ambiguous nodes, which is one of the highlight contributions of this paper, as its results are consistent across different datasets and contraction ratio, rendering it a versatile tool for network ambiguity detection in the challenging situation when besides the network topology itself no additional information (such as node attributes, descriptions, or labels) is available or may be used.

For node deduplication, FONDUE-NDA performed well in settings where the duplicate nodes have a higher than average degree compared to the network, which is arguably the case for this NDD, as duplicate nodes tend to have higher degree.

The main limitation of FONDUE is its reliance on the scalability of the embedding method. The current backend NE method being CNE, the scalability is limited to medium-sized networks with sub-100,000 nodes.

Implementing additional NE methods for FONDUE-NDA and FONDUE-NDD could be one future areas for exploring and improving the state-of-the-art of NDA and NDD.

5. Conclusions

In this paper, we formalized both the node deduplication problem and the node disambiguation problem as inverse problems. We presented FONDUE as a novel method that exploits the empirical fact that naturally occurring networks can be embedded well using state-of-the-art network embedding methods, such that the embedding quality of the network after node disambiguation or node deduplication can be used as an inductive bias.

For node deduplication, we showed that FONDUE-NDD, using only the topological properties of a graph, can help identify nodes that are duplicate, with experiments on four different datasets successfully demonstrating the viability of the method. Despite it not

being an end-to-end solution, it can facilitate filtering out the best candidate nodes that are duplicates.

For tackling node disambiguation, FONDUE-NDA decomposes this task into two sub-tasks: identifying ambiguous nodes, and determining how to optimally split them. Using an extensive experimental pipeline, we empirically demonstrated that FONDUE-NDA outperforms the state-of-the-art when it comes to the accuracy of identifying ambiguous nodes, by a substantial margin and uniformly across a wide range of benchmark datasets of varying size, proportion of ambiguous nodes, and domain, while keeping the computational cost lower than that of the best baseline method, by nearly one order of magnitude.

On the other hand, the boost in ambiguous node identification accuracy was not observed for the node splitting task, where FONDUE-NDA underperformed compared to the competing baseline, Markov clustering. Thus, we suggested a combination of FONDUE for node identification, and Markov clustering on the ego-networks of ambiguous nodes for node splitting, as the most accurate approach to address the full node disambiguation problem.

Author Contributions: Conceptualization, B.K. and T.D.B.; methodology, A.M., B.K. and T.D.B.; software, A.M. and B.K.; validation, A.M., B.K., J.L. and T.D.B.; formal analysis, A.M. and B.K.; investigation, A.M. and B.K.; resources, J.L. and T.D.B.; data curation, A.M. and B.K.; writing—original draft preparation, A.M., B.K., J.L. and T.D.B.; writing—review and editing, A.M., B.K., J.L. and T.D.B.; visualization, A.M. and B.K.; supervision, B.K., J.L. and T.D.B.; project administration, B.K.; funding acquisition, J.L. and T.D.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the European Union’s Horizon 2020 research and innovation programme (ERC Grant Agreement No. 963924), from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme, and from the FWO (project no. G091017N, G0F9816N, 3G042220).

Data Availability Statement: All data used in this study are publicly available, from other sources. See Section 3.1 and Table 1 for sources.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

NDA	Node Disambiguation
NDD	Node Deduplication
NE	Network Embeddings
CNE	Conditional Network Embeddings

Appendix A. Real-World Example: PubMed Dataset

To help illustrate the aforementioned problems in the real-world, we processed the publicly available PubMed dataset courtesy of the U.S. National Library of Medicine, that comprises more than 31 million citations for biomedical literature from various sources, with metadata ranging from author names to their Orcid IDs. An Orcid ID is a unique author identifier used to distinguish researchers. It is useful information to uncover ambiguous author names (an author name that has multiple Orcid IDs) or author name duplicates (distinct names that correspond to the same Orcid ID). From a quick analysis of the 17 millions author data present in the dataset, there exist 154,247 duplicate author names (different names that refer to the same author), and 62,506 ambiguous author names (different authors with the same name). A further visualization in Figure A1 shows the frequency histogram of ambiguous author names with multiple Orcid IDs which ranges

from 2 up to more than 200 per name. The most common cause for this problem is that some distinct authors share the same name, which is very common for names for authors of Asian background, as the romanization of their name is often ambiguous. The most ambiguous names in the dataset are ‘Wei Zhang’ and ‘Wei Wang’, both with more than 200 different authors associated with each of them.

Figure A1. The frequency of ambiguous author names with multiple Orcid IDs. It represents the frequency of names that are shared with distinct authors. ‘Wei Zhang’ and ‘Wei Wang’ are the most common names, being shared with over than 200 distinct authors each.

As for duplicate author names, shown in Figure A2, the main cause is multiple name variations for the same author, for example, the Orcid ID 0000000240600292 belongs to 4 variations of the name ‘Robert Henry’ (Robert J. Henry, Robert James Henry, and R. Henry) that refer to the same author in the dataset. The second cause is erroneously parsed metadata provided by the PubMed baseline dataset, where the coauthors share the same Orcid ID as one of the other authors.

Figure A2. The frequency of different author names that share the same Orcid ID. It represents Orcid IDs that are present in more than 1 author name entry, due to author name variation.

Both of those problems are the result of both the processing and parsing tasks, which are very common for such data type.

Appendix B. Tabulated Results for Ambiguous Node Identification

As discussed in Section 4.2.2, we argued that the best metric for evaluating the task was NDCG. Nonetheless, as many approaches use the classical binary classification metrics, we present below detailed results of the experiments.

Table A1. Performance evaluation for AUC score, DCG and F1-score on multiple datasets for our FONDUE-NDA compared with other baselines. Note that for some datasets with a small number of nodes, we did not perform any contraction for 0.001 as the number of contracted nodes in this case is very small, thus we replaced the values for those methods by “—”.

Ambiguity Rate		10%				1%				0.1%			
	Method	FONDUE-NDA	NC	CC	Degree	FONDUE-NDA	NC	CC	Degree	FONDUE-NDA	NC	CC	Degree
AUC	fb-sc	0.951	0.793	0.163	0.737	0.968	0.836	0.517	0.713	0.989	0.936	0.747	0.690
	fb-pp	0.871	0.720	0.744	0.724	0.895	0.699	0.762	0.726	0.888	0.673	0.768	0.673
	email	0.768	0.491	0.280	0.730	0.862	0.587	0.380	0.736	—	—	—	—
	student	0.785	0.555	0.549	0.724	0.878	0.316	0.708	0.839	—	—	—	—
	lesmis	0.833	0.556	0.366	0.757	—	—	—	—	—	—	—	—
	polbooks	0.952	0.590	0.308	0.841	1.000	0.813	0.327	0.877	—	—	—	—
	ppi	0.746	0.559	0.641	0.726	0.792	0.555	0.647	0.739	0.943	0.522	0.630	0.803
	netscience	0.933	0.877	0.807	0.805	0.952	0.821	0.669	0.842	—	—	—	—
	GrQc	0.872	0.807	0.796	0.740	0.890	0.812	0.785	0.719	0.944	0.794	0.675	0.696
	CondMat	0.861	0.837	0.797	0.757	0.867	0.839	0.810	0.744	0.873	0.840	0.793	0.709
	HepTh	0.853	0.754	0.780	0.743	0.861	0.746	0.775	0.733	0.902	0.741	0.799	0.764
	cm05	0.867	0.852	0.811	0.754	0.889	0.867	0.815	0.758	0.910	0.870	0.824	0.770
	cm03	0.875	0.851	0.815	0.758	0.886	0.843	0.822	0.742	0.909	0.818	0.802	0.751
DCG	fb-sc	56.843	57.324	45.767	46.226	8.504	9.709	6.313	4.695	1.738	1.371	0.697	0.509
	fb-pp	216.187	198.421	197.420	193.395	24.629	20.177	20.028	19.358	2.798	2.005	2.008	1.894
	email	16.150	13.644	12.779	14.533	2.249	1.297	1.124	1.317	—	—	—	—
	student	8.481	7.243	6.196	7.105	0.845	0.698	0.501	0.548	—	—	—	—
	lesmis	3.297	2.073	1.816	2.262	—	—	—	—	—	—	—	—
	polbooks	4.415	2.746	2.427	3.170	1.000	0.310	0.267	0.318	—	—	—	—
	ppi	43.601	38.479	41.609	42.611	4.504	3.778	4.086	4.152	0.414	0.294	0.313	0.323
	netscience	9.324	8.255	7.696	7.594	1.083	0.806	0.688	0.613	—	—	—	—
	GrQc	52.236	49.072	48.513	46.8	6.797	5.041	4.928	4.676	0.639	0.499	0.471	0.431
	CondMat	199.255	197.187	194.376	188.157	21.936	20.186	19.841	18.982	2.663	2.017	1.955	1.856
	HepTh	93.870	87.125	89.909	86.928	10.935	8.754	9.299	8.829	1.286	0.796	0.886	0.821
	cm05	320.094	316.111	310.849	299.216	34.507	32.485	32.009	29.989	4.871	3.230	3.171	2.973
	cm03	253.117	247.685	242.945	234.606	28.164	25.558	24.903	23.652	2.861	2.537	2.406	2.323
F1-Score	fb-sc	0.744	0.856	0.759	0.251	0.550	0.748	0.528	0.065	0.250	0.300	0.013	0.013
	fb-pp	0.476	0.327	0.315	0.247	0.188	0.092	0.029	0.028	0.045	0.074	0.007	0.002
	email	0.388	0.227	0.190	0.277	0.111	0.017	0.006	0.022	—	—	—	—
	student	0.410	0.226	0.139	0.266	0.333	0.107	0.013	0.04	—	—	—	—
	lesmis	0.571	0.297	0.223	0.331	—	—	—	—	—	—	—	—
	polbooks	0.700	0.288	0.260	0.284	1.000	0.24	0.04	0.12	—	—	—	—
	ppi	0.265	0.075	0.226	0.245	0.079	0.007	0.013	0.018	0	0	0	0
	netscience	0.595	0.517	0.441	0.368	0.333	0.093	0.053	0.013	—	—	—	—
	GrQc	0.398	0.352	0.322	0.251	0.171	0.102	0.041	0.041	0	0.025	0	0
	CondMat	0.395	0.418	0.343	0.280	0.089	0.094	0.036	0.033	0.095	0.010	0.008	0
	HepTh	0.397	0.279	0.329	0.273	0.128	0.054	0.062	0.047	0.125	0.013	0.033	0
	cm05	0.409	0.451	0.366	0.267	0.099	0.131	0.045	0.026	0.111	0.024	0.011	0
	cm03	0.424	0.449	0.369	0.279	0.124	0.127	0.042	0.031	0.037	0.021	0.000	0

References

1. National Library of Medicine. PubMed Dataset. Available online: <https://pubmed.ncbi.nlm.nih.gov/help/> (accessed on 20 October 2021).
2. Xenarios, I.; Rice, D.W.; Salwinski, L.; Baron, M.K.; Marcotte, E.M.; Eisenberg, D. DIP: The database of interacting proteins. *Nucleic Acids Res.* **2000**, *28*, 289–291. [CrossRef]
3. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin, Germany, 2007; pp. 722–735.
4. Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; Yang, S. Community preserving network embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
5. Fortunato, S.; Barthélemy, M. Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **2007**, *104*, 36–41. [CrossRef]
6. Kang, B.; Lijffijt, J.; De Bie, T. Conditional Network Embeddings. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
7. Tang, J.; Fong, A.C.; Wang, B.; Zhang, J. A Unified Probabilistic Framework for Name Disambiguation in Digital Library. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 975–987. [CrossRef]

8. Haak, L.L.; Fenner, M.; Paglione, L.; Pentz, E.; Ratner, H. ORCID: A system to uniquely identify researchers. *Learn. Publ.* **2012**, *25*, 259–264. [\[CrossRef\]](#)
9. Zhang, B.; Al Hasan, M. Name Disambiguation in Anonymized Graphs Using Network Embedding. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Singapore, 6–10 November, 2017; ACM: New York, NY, USA, 2017; pp. 1239–1248.
10. Parravicini, A.; Patra, R.; Bartolini, D.B.; Santambrogio, M.D. Fast and Accurate Entity Linking via Graph Embedding. In Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA), GRADES-NDA'19, New York, NY, USA, 30 June–5 July 2019; ACM: New York, NY, USA, 2019; pp. 10:1–10:9.
11. Shen, W.; Wang, J.; Han, J. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 443–460. [\[CrossRef\]](#)
12. Zhang, Y.; Zhang, F.; Yao, P.; Tang, J. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1002–1011.
13. Ma, X.; Wang, R.; Zhang, Y. Author name disambiguation in heterogeneous academic networks. In *International Conference on Web Information Systems and Applications*; Springer: Berlin, Germany, 2019; pp. 126–137.
14. Lerchenmueller, M.J.; Sorenson, O. Author disambiguation in PubMed: Evidence on the precision and recall of author-ity among NIH-funded scientists. *PLoS ONE* **2016**, *11*, e0158731. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Ma, Y.; Wu, Y.; Lu, C. A Graph-Based Author Name Disambiguation Method and Analysis via Information Theory. *Entropy* **2020**, *22*, 416. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Saha, T.K.; Zhang, B.; Al Hasan, M. Name disambiguation from link data in a collaboration graph using temporal and topological features. *Soc. Netw. Anal. Min.* **2015**, *5*, 11. [\[CrossRef\]](#)
17. Hermansson, L.; Kerola, T.; Johansson, F.; Jethava, V.; Dubhashi, D. Entity disambiguation in anonymized graphs using graph kernels. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 1037–1046.
18. Xu, J.; Shen, S.; Li, D.; Fu, Y. A network-embedding based method for author disambiguation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018; pp. 1735–1738.
19. Chen, T.; Sun, Y. Task-guided and path-augmented heterogeneous network embedding for author identification. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; pp. 295–304.
20. Cavallari, S.; Zheng, V.W.; Cai, H.; Chang, K.C.C.; Cambria, E. Learning Community Embedding with Community Detection and Node Embedding on Graphs. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, Singapore, 6–10 November 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 377–386.
21. Weichselbraun, A.; Kuntzsch, P.; Brasoveanu, A.M. Name variants for improving entity discovery and linking. In Proceedings of the 2nd Conference on Language, Data and Knowledge (LDK 2019), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Leipzig, Germany, 20–23 May 2019.
22. Alhelbawy, A.; Gaizauskas, R. Graph ranking for collective named entity disambiguation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Baltimore, MD, USA, 2014; Volume 2, pp. 75–80.
23. Guo, Y.; Che, W.; Liu, T.; Li, S. A graph-based method for entity linking. In Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 7–13 November 2011; pp. 1010–1018.
24. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
25. Nesterov, Y. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.* **1998**, *9*, 141–160. [\[CrossRef\]](#)
26. Luo, Z.Q.; Ma, W.K.; So, A.M.C.; Ye, Y.; Zhang, S. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Process. Mag.* **2010**, *27*, 20–34. [\[CrossRef\]](#)
27. Leone, F.; Nelson, L.; Nottingham, R. The folded normal distribution. *Technometrics* **1961**, *3*, 543–550. [\[CrossRef\]](#)
28. Van Leeuwen, M.; De Bie, T.; Spyropoulou, E.; Mesnage, C. Subjective interestingness of subgraph patterns. *Mach. Learn.* **2016**, *105*, 41–75. [\[CrossRef\]](#)
29. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. 2014. Available online: <http://snap.stanford.edu/data> (accessed on 20 October 2021).
30. Goethals, B.; Le Page, W.; Mampaey, M. Mining interesting sets and rules in relational databases. In Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010; pp. 997–1001.
31. Breitkreutz, B.J.; Stark, C.; Regulj, T.; Boucher, L.; Breitkreutz, A.; Livstone, M.; Oughtred, R.; Lackner, D.H.; Bähler, J.; Wood, V.; et al. The BioGRID interaction database: 2008 update. *Nucleic Acids Res.* **2007**, *36*, D637–D640. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Knuth, D.E. *The Stanford GraphBase: A Platform for Combinatorial Computing*; AcM Press: New York, NY, USA, 1993.
33. Newman, M.E. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 404–409. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Järvelin, K.; Kekäläinen, J. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* **2002**, *20*, 422–446. [\[CrossRef\]](#)