FeatureCloud

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare

Deliverable D4.4
"Experimental results for shape and composition of connection surfaces"

WP4
"Supervised Federated Machine Learning"

Disclaimer

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078. Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

Document information

Grant Agreement Number 826078			**Acronym: FeatureCloud**	
Full title	Privacy preserving federated machine learning and blockchaining for reduced cyber risks in a world of distributed healthcare			
Topic	Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures			
Funding scheme	RIA - Research and Innovation action			
Start Date	1 January 2019	**Duration**		60 months
Project URL	https://featurecloud.eu/			
EU Project Officer	Reza RAZAVI (CNECT/H/03)			
Project Coordinator	Jan Baumbach, TECHNISCHE UNIVERSITAET MUENCHEN (TUM)			
Deliverable	D4.4 “Experimental results for shape and composition of connection surfaces”			
Work Package	WP4 “Supervised Federated Machine Learning”			
Date of Delivery	**Contractual**	M24 (31/12/20)	**Actual**	M24 (31/12/20)
Nature	REPORT	**Dissemination Level**		PUBLIC
Lead Beneficiary	03 MUG			
Responsible Author(s)	Andreas Holzinger			
Keywords	Graph-based machine learning, Graph representation learning, Node & Graph embeddings			

Table of Content

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

Acronyms and definitions

AI	artificial intelligence
API	application programming interface
BA	Barabasi-Albert
BFS	breadth-first search
BOW	Bag of Words
CNN	Convolutional Neural Network
concentris	concentris research management GmbH (Germany)
CPU	Central processing unit
CRF	Conditional Random Field
D	degree matrix
D2V	Doc2Vec
DB	database
DBMS	database management system
DF	Decision Forest
DGL	deep graph library
DW	Deep walk
FB	Facebook
FL	federated learning
GB	gigabyte
GCNN	Graph Convolutional Neural Network
GloVe	Global Vectors for Word Representations
GND	Gnome Design SRL (Romania)
GNN	Graph neural network
GPU	Graphics processing unit
GraphSAGE	Graph Sampling & Aggregation
GRL	Graph representation learning
IID	Independent and Identically Distributed
KB	kilobyte
L	Laplacian matrix
LSTM	Long short-term memory
MB	megabyte
ML	Machine learning
MM	Multi-Modal
MUG	Medizinische Universität Graz (Austria)
NLP	Natural language processing
NLP	natural languages processing

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

NN	Neural Networks
NN-based	Neural Network-based
OGB	Open Graph Benchmark
PB	petabyte
PCA	Principal component analysis
PGN	Pointer Graph Network
ReLU	Rectified Linear Unit
RF	Random Forest
RI	Research Institute AG & Co. KG (Austria)
RW	Random Walk
SBA	SBA Research gemeinnützige gGmbH (Austria)
SDU	Syddansk Universitet (Denmark)
T	tree
TUM	Technische Universität München (Germany)
UM	Universiteit Maastricht (The Netherlands)
UMR	Philipps Universität Marburg (Germany)
UUID	Universally unique identifier
vs.	versus
w.r.t.	with respect to
W2V	Word2Vec
WWW	world wide web
xAI / XAI	Explainable Artificial Intelligence
XGNN	Explanations of Graph Neural Networks

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

1 Objectives of the deliverable based on the Description of Action

Within WP4, *Supervised Federated Machine Learning*, a central focus of our work lies on decentralization and distributed processing, which stems from increasing expectations of privacy and security in Machine Learning applications, legal restrictions to the transmission of sensitive information, as well as the nature of Federated ML itself. Thus, from the project’s inception we were interested in using graph data structures as they are naturally suitable for distributed computation. However, in contrast to our initial assumptions of dealing with stationary graph partitions as well as conventional methods of concurrency / parallelization (albeit, due to the nature of the problem, in a more challenging setting, as we would never see the global graph in its entirety), we quickly realized the necessity to shift our paradigm from a purely graph-theoretical viewpoint to a data-driven approach grounded in medical reality.

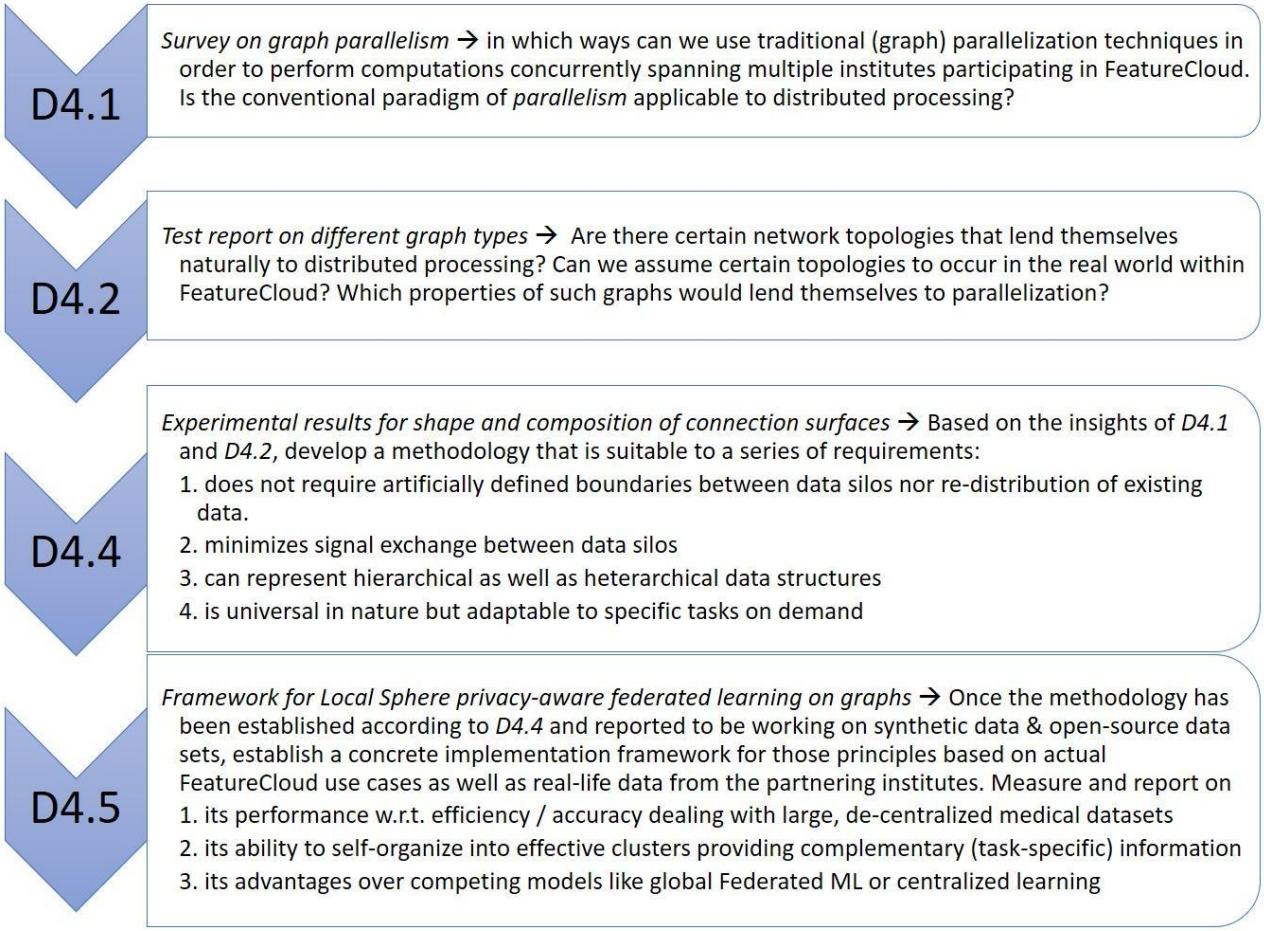


***Figure 1**. Overview of the graph-specific milestones / deliverables within WP4.*

Therefore, our objectives for this deliverable lie in outlining and demonstrating a generally applicable, graph-based embedding approach that is capable of

- bridging distributed data sets of same modalities modelling a common ground truth (application domain).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

- *dealing with pre-determined "partitions"*, *minimization of signal exchange*, an *abstraction / conceptualization of concrete input data* as well as a *universal yet flexible representation of entities*.

2 Executive Summary / Abstract

From the inception of the FeatureCloud project, we knew that distributed data silos in combination with harsh data-protection and privacy regulations would make effective, efficient, and secure decentralized data processing a crucial aspect of our research endeavors. Therefore, it quickly became obvious that we would be modeling our data according to structures & models that naturally lend themselves to distributed computation. Apart from simple linear models, Decision Trees or Random Forests, Graphs have been one of the most universal, flexible and most intuitively separable data structures in existence, with a wealth of mathematical approaches and algorithmic research at our disposal.

While we originally intended to approach the problem from the angle of traditional graph partitioning and conventional concurrency / parallelism, we soon realized the need for a new paradigm in the age of edge-computing, privacy & (local) Explainability. Embeddings, which are low-dimensional, dense representations of concepts, hold the advantage of compositionality, reflecting one of the key properties of reality, enabling the concise representation of various interesting structures including clusters, hierarchies, heterarchies or any arbitrary combinations of those. The recent rise of graph neural networks (GNN) will thus allow us to accommodate the requirements of **i.** dealing with pre-determined "partitions", **ii.** minimization of signal exchange, **iii.** abstraction / conceptualization of concrete input data as well as **iv.** a universal yet flexible representation of entities.

This report will detail the train of thought we already outlined in earlier documents, explaining the need for **i.** distributed embeddings, therefore **ii.** embeddings from different sources and **iii.** different modalities; **iv.** we will sketch out a series of experiments that we are currently conducting to approach this complex undertaking by *simulating* distributed computations using two different graph data sets modeling the same underlying domain. We are also outlining in detail future work and challenges the research community will have to face, before ushering in a new era of patient-controlled, personalized & decentralized Machine Learning services.

The title of this report, *“Shape and composition of connection surfaces”*, is our way to describe the compositional nature of networks, meaning that only very points of connections are needed to construct a whole from distributed parts. This can be thought of traditionally as edges spanning subgraphs (as in graph partitioning), as communication channels along which messages pass (as in Belief Propagation), but also as representatives of one network temporarily infused into another (see section *virtual nodes* below).

This document is therefore a logical successor to our earlier surveys of graph types & parallelism as well as a precursor to our ultimate goal of *Local Sphere privacy-aware Federated Learning* on graphs (see Fig. 1).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

3 Introduction and Motivation

Learning on fused data from different sources and modalities can substantially outperform traditional methods learning on just one type of data structure; jointly learning on input data of different modalities is therefore a standard routine (Wang et al., 2014), (Liu et al., 2014), with a constant stream of innovation in recent years (Tong et al., 2017), (Vivar et al., 2019).

The fundamental challenge in fusing disparate modalities lies in bridging the semantic gap between them and handling potential disagreements thereof (Wei et al., 2019). In our application domain, probable sources of information are (histopathological) images, medical databases, personal case files as well as time-series data (see Fig. 2).

This problem is identified within the literature as *aligning local geometries* of subjects across feature spaces (Liu et al., 2014), but in earlier works multi-modality was understood to be limited to different image taking techniques (e.g. CT, MRI, PET, etc.) or different resolutions. Simple methods just concatenate feature vectors to fuse different domains, not considering varying distance or neighborhood metrics across domain boundaries. We believe it is important to initially capture each source domain's intrinsic *ontology* - e.g., relations, hierarchies, partitions in a graph, analogies, co-occurrence, and other forms of semantics within texts, or pathways on an *omics level; concurrently, we need to define *links* - interactions and correspondences - between entities of different domains. For instance, a super-pixel in an image could correspond to an entry in a controlled vocabulary, or a mutation within a gene causes a different behavior on the protein level. For a lack of pre-defined *cross-domain semantics* most of this work will have to be done either manually or by deriving connection rules from knowledge provided by human experts; this will require intensive interdisciplinary effort. Sampling this ontology-enriched cross-domain graph produces positive / negative instances, which are subsequently fed to an embedding algorithm.

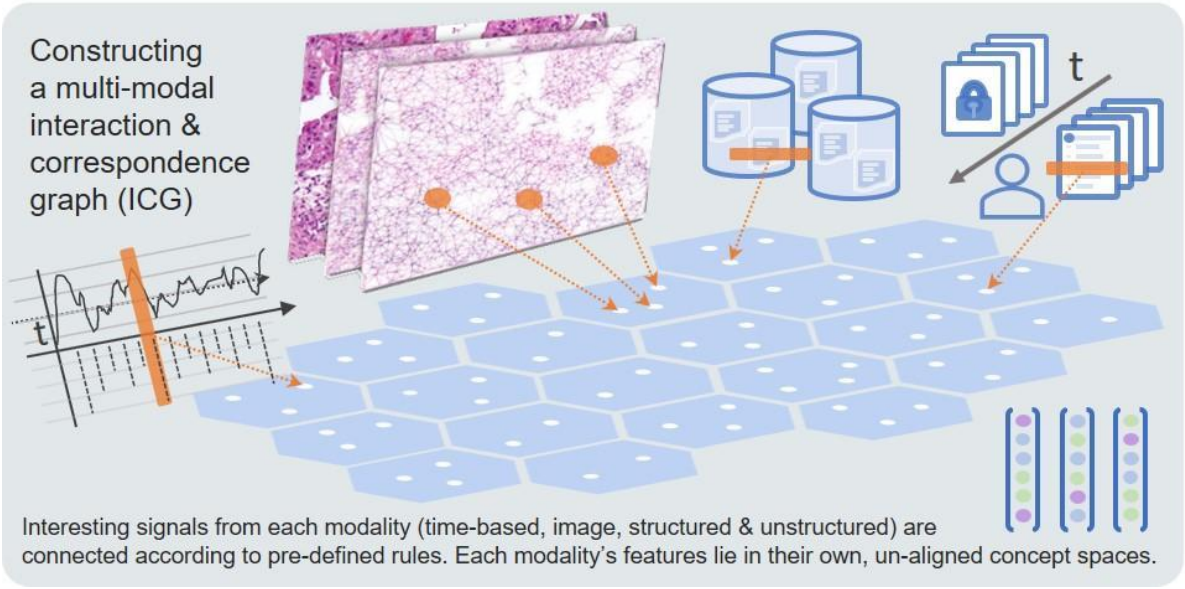


Figure 2: *A multi-modal, distributed embedding space constructed from isolated data sources whose internal representations are connected via a knowledge graph. Source: Bernd Malle*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078.

Two possible approaches to this end are joint embeddings (Wu et al., 2018) and Graph Representation Learning (GRL) (Hamilton et al., 2017), (Kipf & Welling, 2017).

Generally, embeddings are *low-dimensional, dense vector representations* of entities which are usually learned from large corpora in an unsupervised fashion, i.e. by forcing an algorithm to approximate pairwise distance in the embedding space to a given similarity function in the original, higher-dimensional input space. There is a wealth of literature on so-called *neural embeddings*, with the most prominent work presented by (Bengio et al., 2001).

In order to connect information from images, text and *omics data, we need to compute and learn representations for nodes / sub-graphs in this low-dimensional space, considering node-level features as well as their structural surroundings (*graph neighborhoods*). Within this embedding space we can subsequently compare items in diverse ways (e.g. hierarchies, analogies, clustering, etc.) across different input modalities whose intrinsic geometries have been aligned during the joint embedding procedure.

The most important aspect in this phase is generating positive / negative samples within and across modalities as input features - as an example, *Word2Vec* (Mikolov, Sutskever, et al., 2013) assumes word co-occurrence within sentences as positive samples - this is highly domain specific and dependent on human assumptions, which constitutes a *bias trap* that will have to be carefully avoided through repeated experimentation with diverse sets of assumptions and automated as well as human control (objective performance measures and inter-expert validation combined).

Recent research also successfully demonstrated a combination of these different methods in a two-step process: **i)** pre-processing intra-modal data using traditional assumptions about their respective domain, and **ii)** feeding the resulting, normalized feature vectors into a graph representation learner (neighborhood aggregation, GAN, etc.); we will extend this approach to multi-modal data sets. Others utilized traditional machine learning techniques (e.g. Random Forest) to build intra-modal similarity matrices which they subsequently fuse (Tong et al., 2017), but this requires the full matrix to exist at computation time, which is unrealistic for larger or distributed graphs. Depending on the approach, we might need to utilize a pre-instantiated target dictionary of features curated by humans (e.g. cell-types visible in an image).

The long-term goal is to produce a corpus of multi-modal feature representations originating in e.g. histopathological images, patient case files (text), *omics data as well as medical knowledge bases, where we can utilize a *knowledge graph* as an *initial connector*. Such pre-trained concept embeddings would constitute an advanced pendant to word & document embeddings like the well-known *GloVe* (Pennington et al., 2014) or *fasttext* (Joulin et al., 2017) corpora and could help institutions worldwide to elevate the utility of their already existing, yet unstructured and unconnected data bases.

Beyond establishing broadly applicable embeddings, GRL can also be used in a supervised fashion by integrating a label-based term into the loss function. This would enable end-to-end learning on specific tasks, where one network architecture comprises all processing steps from handling raw inputs up to the final prediction. However, this comes with the downside of decreased generality in the learned representations (if those are task-specific, they are not reusable across objectives),

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

unless informed data augmentation (Bloice et al., 2017) techniques are used. Moreover, embeddings learned as an intermediate step within a neural network will generally not correspond to concepts of human understanding and will therefore lack causability by nature.

4 Experiments

In order to prepare ourselves for the training of large-scale multi-modal networks on real-world data sets located in the data silos of the respective partner institutions within FeatureCloud we first need to work on data that are:

- available in their entirety (as we need to compare our results to a "traditional" Machine Learning setting)
- small enough to run repeated experiments within a short time horizon
- easily interpretable - in exploring algorithmic possibilities (assumptions, hyper-parameters, constraints, etc.) researchers are often required to make multiple decisions per experimental series. It is therefore necessary to work on data that allow easy interpretation by non-domain experts (such as ML practitioners) without the need for consulting medical professionals whose time is usually severely limited.
- uncritical from a data security and privacy standpoint. Although eventually FeatureCloud's objective is to integrate large, medical data sets, we want to keep our training data unobjectionable for the longest duration possible.
- Multi-source but *not* multi-modal at first, in order to tackle one challenge after the other.

For all these reasons, we decided to follow the established approach of training on publicly available, relatively small as well as easily interpretable data sets, albeit not originating in the medical domain.

4.1 Challenge

In order to match nodes between different graphs modeling the same underlying domain, we employ graph representation learning in different settings and / or differently modified input graphs.

Matching structures in different graphs, as well as subgraph matching within the same graph, are still open problems due to the exponential number of possible structural combinations (= *combinatorial explosion*). One possible approach to solving this matter efficiently might lie in computing embeddings, i.e., low-dimensional, sparse vector representations, for nodes and whole subgraphs. The basic idea is to reduce any arbitrarily complex neighbourhood structure surrounding a node into a compact representation by "aggregating" the feature vectors of its adjacent vertices (for each node of the graph). The resulting numerical representations are assumed to be representative of ***i)*** the features of nodes surrounding a node, but also **ii)** the local neighborhood

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

structure. This should theoretically result in high cosine similarity for nodes of similar "meaning" in the graph.

Our overarching goal for this challenge is to experiment on two data sets - in principle already provided as curated graphs - and compute node embeddings in order to perform:

- node matching: nodes in different graphs exhibiting close cosine similarity should hold similar meaning in their original spaces (see Fig. 3). A challenge here is to check for similar meaning in the first place (and the original graphs) since textual descriptions in node feature vectors might only be semantically "close" according to human understanding.

- link prediction: here we can use the original graphs as *ground truth*, which means links (edges) existing between nodes in those graphs should also crystallize in the computed embedding space (via proximity between nodes). An additional challenge in this setting is the sparsity problem, since in any tractable graph there are usually just a tiny fraction of possible edges present in the actual network. Thus, it is much easier to detect positive occurrences than negative ones.

- Representative virtual node computation: Given a task we wish to solve in a distributed fashion while minimizing signal exchange across boundaries, we can first train each local network on the task, identifying clusters of nodes and sub-graphs of highest significance / contribution to the result (Ying et al., 2019). Connecting these clusters to a virtual node, thereby learning their combined representation, will offer a natural point of connection between distributed *graphs*.

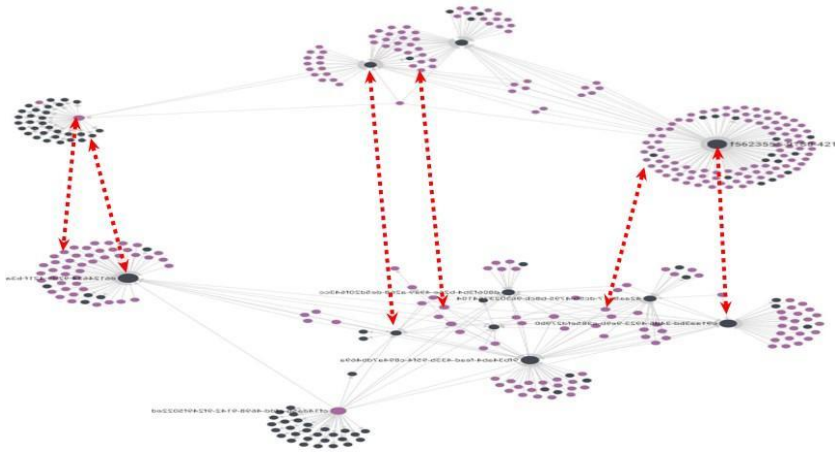


Figure 3. *Two different graphs (north=N and south=S) modeling the same underlying domain can exhibit very different graph structure, depending on bias, measuring or modeling approaches. Given such different views onto the same ground truth, can we use node embeddings / GRL to find node pairs (u, v) with u ∈ N, v ∈ S and low proximity cos(u, v) ≈ 0 so that they represent the same concept in ground truth? In our example setup, we would ask whether two nodes modeled in ESCO and O*NET, respectively, can be mapped to close points in embedding space if they encode e.g. the same occupation. In our experiments, we considered a) only network structure, and b) network structure as well as pre-processed, initial node embeddings. Source: Bernd Malle, generated with ArangoDB graph viewer.*

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

Although learning embeddings from 2 graphs separately sounds like standard procedure, their coordinate systems will not align naturally - meaning the *k^th* dimension of a vector in space *A* will not automatically bear the same meaning as the *k^th* dimension of a vector in space *B*. It is therefore necessary to train the embeddings via a joint procedure. However, without natural connections between the two graphs, message passing will fail to propagate information (without mechanisms like *“Teleportation”* (as in Pagerank) or intrinsic, ad-hoc link prediction as in *Pointer Networks* (Veličković et al., 2020). Likewise, the generation of meaningful positive / negative samples will be impossible utilizing the network structure alone.

4.2 Data set preparation

For this challenge, we prepared two graph-based data sets modeling the same underlying domain covering skills, occupations, knowledge as well as their relations. One of these data sets is *ESCO* (European classification of Skills/Competencies & Occupations), the other being its U.S. counterpart *O*NET* (Occupational Information Network). Both contain nodes representing *Occupations* (which we have renamed and shall henceforth refer to as *“Jobs”* simply for the sake of (code) brevity, *Skills & Knowledge* (although *O*NET* covers a variety of other entities – e.g. *Tasks* – in addition), but there are striking differences regarding their makeup: While *ESCO* contains richer textual information per node and a larger collection of skills, *O*NET* offers a much more complex (and therefore presumably more expressive) network structure. It will thus be interesting to see whether (and by what initial assumptions in graph construction) we can arrive at representations capable of matching similar nodes from different structural embeddings.

One interesting aspect of both data sets is their diverse nature concerning relationship types, as they encompass *hierarchical* relations (a *job* can belong to a *job category*, the same goes for skills), *heterarchical* relations (*skills* can be related, just like friends in a social network), and *bipartite* relations (in *ESCO*, *jobs* and *skills* form a bipartite subgraph via their *required-by* relation, which means that a skill can be related to another skill by sharing a connected job for which they are essential or optional, even without a direct link established between them).

Both corpora were provided as *.csv/.tsv* files, but whereas *ESCO* follows a clear separation into files containing nodes and edges, *O*NET* – probably due to its much longer history and therefore organic growth – presented us with a multitude of files that often contained nodes as implicit ends in an edge-list format, sometimes even encoding hyper-edges, requiring significantly more manual pre-processing time, especially when having to resolve hyper-edges into a format expressible as simple adjacency or edge lists. For node / hyper-node / edge statistics, please refer to Appendix A.

In order to ensure a consistent graph structure before starting our experiments, we chose to instantiate both graphs in a graph DBMS (ArangoDB); this has the additional benefit of providing a “ground truth” from which we can draw different graph *views* (= subgraphs filtered by queries) for diverse sets of experiments in the future. We subsequently utilized the DBMSs built-in export feature to obtain sanitized / normalized *.csv* files amenable to import into a ML pipeline without much pre-processing; the only step we still had to perform was normalization of node IDs, since we are using the *deep graph library* (DGL) which assumes node IDs as consecutive integers starting at zero, while our DB supported UUIDs.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

4.3 Feature initialization

Although using network structure alone is often sufficient to achieve respectable results, modern Graph convolutional architectures also build upon feature propagation / aggregation to improve results. It was therefore important to make use of the abundance of textual description included in the *ESCO* / *O*NET* data sets in order to determine the significance of real-world node features in relatively small & simple graph structures. We therefore pre-processed 3 types of node features: ***i.*** textual information (like *description*), which was converted directly to embeddings using the fast spaCy library (Choi et al., 2015), ***ii.*** numerical features which were first converted to one-hot encodings, then compressed via PCA to match the dimensionality of the pre-processed string embeddings, and ***iii.*** categorical data (e.g. *skill type*) which was merely one-hot encoded and stored as a training target but not added to the initial node feature vectors.

5 Methodology

We first wanted to establish a baseline understanding of how well graph-based DL methods work on the chosen data sets. For this first phase of *supervised* learning, we used a simple *RGCN* (Relational Graph Convolutional Network) (Schlichtkrull et al., 2018) with a two-layer architecture, either using pre-computed node features as described above or, for reasons of comparison, randomly initialized feature vectors. All experiments were fixed at input / hidden dimensions of *d=300*, while we varied nr. of epochs, learning rates, aggregator function & train/test splits. We used *leaky ReLU*, *cross-entropy* loss and a standard *Adam* optimization.

As far as the network structure itself was concerned, we limited ourselves to one fixed graph structure for *ESCO* while we alternated between a graph structure with & without *hyper*-edges for *O*NET.*

The 6 learning targets were all multi-class classification problems: predicting the *top category* for jobs in each graph, predicting *skill type* and *skill reuse-level* in *ESCO*, and predicting *task type* and *task (rating) source* for *O*NET*.

For unsupervised graph representation learning, we are using GraphSAGE with a loss function of

$$J_G(z_u) = -\,log\,log\left(\sigma(z_u^T z_v)\right) - Q \cdot E_{v_n \sim P_n(v)}\, log\,log\left(\sigma\left(-z_u^T z_{v_n}\right)\right)$$

where

- *u* and *v* are two neighbours
- the loss is computed for *u*
- $log\,log\left(\sigma(z_u^T z_v)\right)$ promotes similarity maximization
- *Q* is the number of negative samples
- v_n is a negative sample drawn from the negative sample distribution $P_{n(v)}$

Since the networks are relatively small in the number of nodes as well as diameter (*d=1* for *ESCO* and *d=4* for *O*NET*, although most of the nodes in the latter are centered around *jobs* at a distance

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

of 1), we set the parameter *k* (the sampling distance determining the neighborhood for the sampling of individual *compute graphs* to *k=1* and *k=2*, respectively).

The output dimension is set to *d=300* in order to make it directly comparable to the supervised methodology given our pre-computed node features. Representations are stored as node-wise embeddings and can be loaded into *gensim KeyedVectors* for downstream computations such as *node- or link-prediction*. These tasks are computed relatively straightforward by ***i.*** comparing the *cosine* similarity of two nodes in embedding space with a *graph-based similarity* score (RW sets + Jaccard similarity) for *node classification*, and ***ii.*** predicting edges by pairwise *cosine* similarity of nodes in the embedding space and comparing against the *existence* of an edge in the actual graph.

Furthermore, in order to train GraphSAGE in a supervised manner, one can either compute the loss directly from a node label (for node classification) or input pairs of nodes, sampling and training pairs of compute graphs which are connected to the final output (for link prediction). A possible loss function (cross-entropy):

$$L = \Sigma_{v \in V} y_v \, log \, log \left(\sigma(z_v^T \theta)\right) + (1 - y_v) \, log \, log \left(1 - \sigma(z_v^T \theta)\right)$$

where

- σ stands for the sigmoid function
- θ is a vector of parameter weights.

5.1 Connecting graphs spatially and / or via embedding similarity

In order to facilitate learning on two separated graphs without adding an unrealistic number of edges between them (i.e., keeping with the assumption of distributed data silos and practically *“stationary”* patient data), we need to find ways of propagating information with a minimum of connection points. In our line of experiments, we realize this by one of 3 methods:

1 Simply relying on pre-trained, initial node embeddings (e.g. w2v) to sufficiently *pre-align* the respective concept spaces so that even after separate trainings, similarities in the embedding space are still indicative of real-world *relatedness*. Besides a potentially unrealistic assumption this method also lacks the potential for inter-network propagation of information and is thus merely suitable for *retaining comparability* between records in different data silos.

2 Connect the graphs via addition of a limited set of edges. This can potentially be done via manual inspection of ground similarities or automated approaches like hierarchical clustering, then connecting the super-nodes obtained. The most fascinating question with this method is how sensitive node embeddings will react to even the smallest amount of edges produced (ideally one should suffice).

3 Compute representative virtual nodes in one graph, then inject those nodes into the other graph via simple similarity measures. The difference to the 2nd approach lies in the fact that we do not assume functional correlation between the injected nodes and their neighbours in

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

the target network; edges are established purely by embedding similarity. We assume that the effectiveness of this approach will largely be determined by how much the concept spaces of the respective (initially unconnected) graphs drift apart during the pre-training phase.

6 Results and discussion

Accuracy for the targets *top-job* (*ESCO*: 10 classes, *O*NET*: 23 classes) were astonishingly high with >99% and >98%, respectively. The reason may lie in the fact that this problem is relatively trivial: the information is contained in the nodes themselves and there is no complex surrounding graph structure that would influence or perturb the algorithm’s convergence. We achieved similarly good results for the targets *ESCO skill-type* (3 classes) with >99% accuracy and *O*NET task source* (3 classes) with equally >99%. We see a different picture with target *ESCO skill reuse-level*, which consistently and across different settings achieved ~65% accuracy. Lastly, target *O*NET task type* reached between ~74% and ~76% accuracy; it is interesting to note that randomly initializing node embeddings & reducing feature dimensions to *d=50* was not able to improve these results significantly.

All multi-class classification problems showed highly skewed distributions, with some classes containing orders of magnitude more members than others. This is true for both the targets that achieved almost perfect scores as well as targets that did not; for example, attributes *Domain Source* and *Type* of *O*NET tasks* are equally skewed with a difference between smallest and largest class of more than 2 orders of magnitude (yet ~75% vs. >99% accuracy).

We varied the number of epochs between 50 and 200 but observed that test accuracy rarely peaked decisively after epoch #50. However, it was interesting to see that test scores tended to *“oscillate”* in some cases as training went on, first indicating *overfitting* after a peak, but then recovering & sometimes even surpassing their previously reached maximum. We speculate that this might lie in the fact that graph convolutions work via message passing in the network and depending on aggregator as well as activation functions, this might lead to cyclical behaviour.

6.1 Effects of network complexity

Although *O*NET* is represented by a much more complex graph than *ESCO,* this complexity had hardly any effect on the supervised training part of our experiments; this is even more interesting since the training targets based on the *skill* nodes in *ESCO* as well as the *task* nodes in *O*NET* have different structural roles in their respective networks: Whereas *skills* in ESCO form a *bipartite* graph together with *jobs* and can therefore be thought of as being equal in importance, *tasks* in *O*NET* are on the periphery of the star-like topology around the centrally located *jobs*.

As mentioned earlier, the *O*NET* graph contains 5 different types of hyper-edges, which we chose to dissolve into a pattern of one central *“hyper-node”* connecting 3 adjacent (real) nodes via simple directed edges. This construct solves the problem of redundant edges but introduces even more complex – and from a message-passing view, probably meaningless – structure into the graph. Throughout our experiments, we observed diminished accuracies across the selected *O*NET* targets by about 2-3%. A possible reason for this could lie in the fact that these additional edges

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

increase the diameter of the network while introducing additional paths for information flow. However, this standpoint would have to be carefully examined w.r.t. aggregation functions, activation functions as well as the original network topology (in *O*NET* they do not change the original star-shaped topology, for instance).

6.2 Feature- vs. featureless initialization

Modern GCNs work on the assumption that a combination of structural and descriptive features tend to achieve much better results than either of these components alone. In our experiments, however, we could observe that replacing pre-computed node embeddings with randomly initiated ones resulted in practically no difference of outcomes. There are several possible reasons for this, including misleading or contradictory node features, a poor choice of embedding methods or pre-trained embedding models, as well as a counter-productive approach to node feature aggregation (especially w.r.t. mixing string- & numeric-based properties). However, we surmise that in our case the simple star-shaped topology of the *O*NET* graph with Tasks, Skills, Tools etc. being directly positioned around Jobs and their edges being grouped via Job categories, makes the network’s structural component alone so descriptive that additional node feature information cannot further improve its results. We will examine this aspect with less symmetric / centralized graphs in the future and report on our findings.

6.3 Unsupervised aggregation-based experiments

Since our *node matching* as well as *link prediction* experiments are still being conducted and intermediate results might be subject to any kind of sources of errors, we have created a github repository that can be found under the following URL:

https://github.com/cassinius/fc-graph-embeddings

This repository will be updated in waves, meaning that new results will arrive as soon as they are published or at least officially submitted.

7 Open issues & future work

7.1 A self-improving embedding – graph – embedding loop?

While graphs are universally applicable tools suitable for almost any undertaking, the task of similarity-based computations (e.g. recommendations) can often be approached computing embeddings, since multiple factors contributing to a desired association between nodes would result in complex, therefore computationally expensive graph traversal rules. However, this similarity can also be measured via cosine distance in the embedding space, with the added benefit that great (or infinite) distances between nodes in the graph are no inevitable hindrance. The training of embeddings from graphs has been successfully demonstrated in recent years; however, in reality one can often not start with a graph. Given a purely text-based corpus, for instance, we would need many assumptions translating into edge formation rules before arriving at a (biased) graph structure.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

Here, we could go the reverse way: starting from an embedding space computed on e.g. text, we use its intrinsic similarity metric to construct a k-nearest neighbor graph. The question thus arises: can we combine both approaches in order to refine embeddings *"out of themselves"*, e.g. because transitive effects in the similarity space can only be captured via structures in the resulting graph (which in turn influences how GRL aggregates features)? An intuition of different ways to arrive at graph embeddings can be seen in Fig. 4.

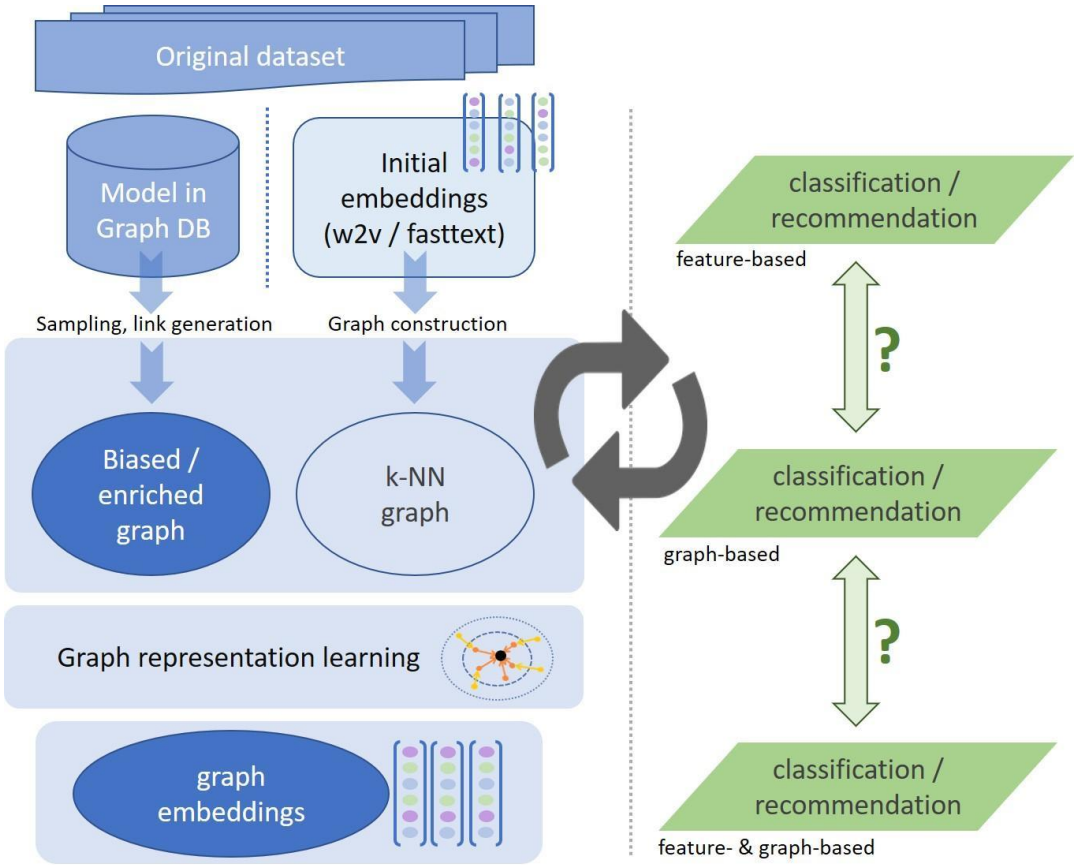


Figure 4. *Starting with an original data set, one can either strive to build a graph immediately or collect & preprocess its features computing initial embeddings; this can often be useful when no intuitive graph structure is present. Since embeddings encode proximity, it is possible to use this embedding space to construct a k-nearest neighbor graph, i.e., for each entry, find the k-nearest neighbors and connect them via an edge. Challenges in this kNN-graph construction typically lie in efficiency of search as well as pruning. Once a graph is established, GRL is used to combine features as well as structure, arriving at either unsupervised or task-specific representations. Each layer of this pipeline yields applicable results (classification / recommendation). Source: Bernd Malle*

7.2 Encoding diverse relation types into embeddings

Embeddings are usually seen as shortcuts to measuring similarity, implying that some form of similarity metric in the *real world* is translated into the more direct version of *cosine similarity* within

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

an embedding space. However, there is no necessity for this internal *"proximity space"* to correspond to actual similarity. As an example, similarity in *Word2Vec* (Mikolov, Sutskever, et al., 2013), (Mikolov, Chen, et al., 2013) does not stem from any measurable string distance, but rather from implicit *semantic* similarity induced via co-occurrence of words within sentences. Therefore, one can easily imagine a mapping function encoding *relatedness* w.r.t. any arbitrary metric to cosine similarity in the embedding space, e.g. Jaccard distance, word co-occurrence, Random Walk distance of two nodes in a graph, distance between counterfactual nodes, or any other kind of concurrence (see Fig. 5). Moreover, two entities mapped to proximity in the embedding space do not necessarily have to stem from the same input source or modality, so a region within an image depicting a tumor could be mapped closely to a genetic mutation causing it (or correlating with it).

However, embeddings spanning different *modalities / sources* are almost non-existent in literature and therefore may require new approaches, especially as cost functions spanning the original input spaces are concerned (e.g. how do we measure the *relatedness between pixels and genes*).

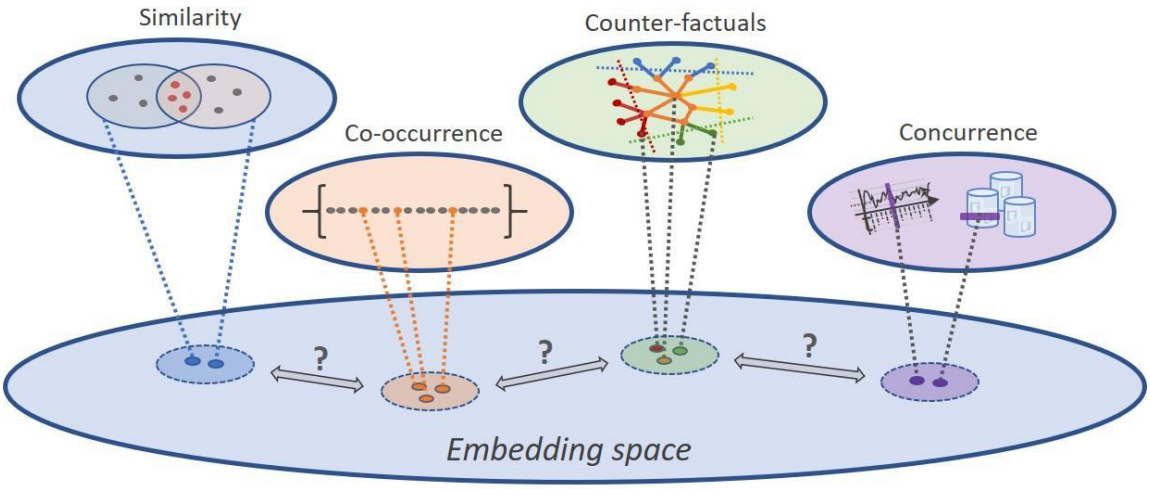


Figure 5: *Embeddings are generated via mappings of any relatedness for which a scoring function is defined into the embedding (“proximity”) space in which pairs of objects that are “close” in their input spaces (according to the scoring function) are close in the proximity space, usually measured via Cosine or Euclidean distance. It is important to note that relatedness in the input space does not necessarily require closeness in a geometric sense, but could be a) overlap of sets, b) co-occurrence of words in a sentence, c) counterfactuals or d) concurrence of signals from different sources / modalities. Source: Bernd Malle*

Establishing a successful, generic approach to this problem might open a plethora of new applications for embedding-based algorithms, including the establishment of a pre-trained, multi-modal embedding space (akin to *Glove* or *fasttext* corpora) for the medical domain.

7.3 Distributed GRL

Distributed learning has been a trending topic for decades, either for reasons of limited central memory & processing power, legal restrictions on data transmission, non-identically distributed data over local sub-populations, or for the potential of hyper-scalable systems operating at lower costs.

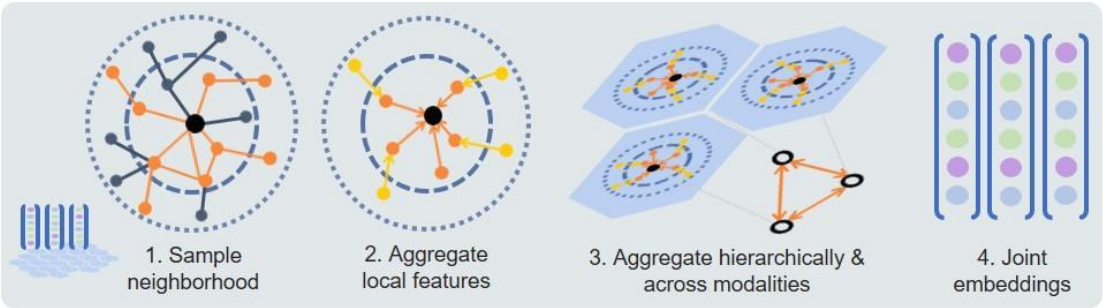
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

This is even more true for the medical domain, where locally generated data is sensitive, practically stationary (is not allowed to be transferred between institutions) and of huge volume (on the order of Terabytes per day).

There are three main versions of distributed learning: **i)** purely decentralized, where local models do not automatically contribute to each other above manual sampling of the models and update of hyper-parameters; **ii)** Federated learning schemes (Malle et al., 2017), (Sattler et al., 2020), where a global model is constructed considering update signals of all local clients which are merged into a global model and distributed back to the clients; **iii)** collaborative learning in various forms, where the goal is to exchange information about internal model formation among the parties involved in a peer-to-peer fashion, yet keep the local training data confidential (a variant could also train on decentralized features supposedly modeling the same underlying instances (Hu et al., 2018). A great challenge for distributed learning of embeddings is to keep the representation space aligned across boundaries – akin to the feature alignment problem in the Multi-modal (MM) setting – and use as few connection points as possible to do so (keep the connection surfaces small).

Considering local pockets of data as natural clusters whose representative *super nodes* can be used as inputs to a more abstract graph embedding step, literature contains several approaches to achieve such coarsened embeddings: while **i)** simple aggregation of node features can already suffice (Duvenaud et al., 2015), **ii)** introducing *virtual nodes* and learning their embeddings together with the rest of the subgraph (Pham et al., 2017) promises a very flexible clustering strategy, e.g. one could run a series of local predictions and record which nodes in the graph had the greatest influence on a desired outcome, this set could then be connected to a virtual node whose vector representation can be expected to be particularly helpful in distributed predictions of the same kind; **iii)** sampling strategies such as Random Walks (RW) or Anonymous RWs (Ivanov & Burnaev, 2018), where the local graph structure itself is translated to fixed-size vectors which are subsequently used as inputs to an embedder.

Lastly, Ying et. al (2018) (Ying et al., 2018) have proposed an extension of their earlier GraphSAGE approach to take hierarchical feature levels into account. They invented graph coarsening “*DiffPool”* layers akin to pooling layers in traditional CNNs which learn a node assignment matrix, thus performing an embedding-level clustering step. Although this approach is innovative, it is unclear whether and to what extent it can handle distributed data. Therefore, we will extend it to suit our future needs (see Fig. 6).



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

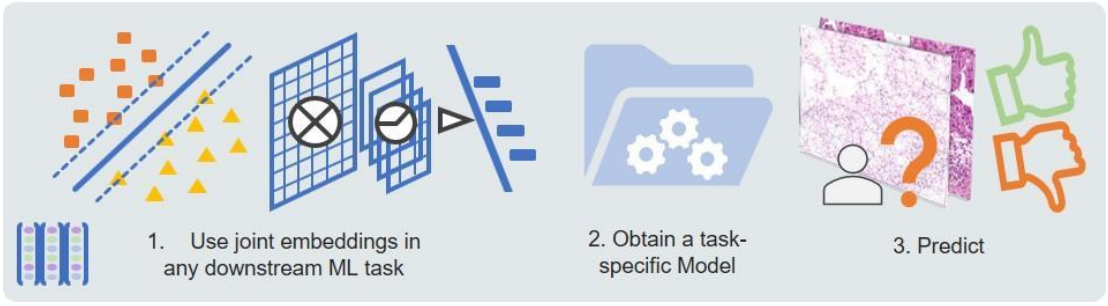


Figure 6: *Learning hierarchies & logical clusterings of nodes in a graph. This approach will extend the original GraphSAGE as well as* (Ying et al., 2018) *to a decentralized setting where node clusters are generated locally & their aggregated feature vectors propagated across the network connecting them to higher-order clusters based on some arbitrary fitness metric. It is important to note that the desired level of abstraction is not fixed, since embeddings could serve a purely representative purpose as in unsupervised learning but could also be task-specific (and computed ad-hoc from an unsupervised "base", if feasible) to solve supervised problems as they emerge. Source: Bernd Malle*

Consequently, our proposed pipeline should broadly consist of the following steps:

- Once a repertoire of suitable GRL / embedding methods has been established, we will simulate a decentralized scenario. We understand *distributed* with the additional challenge that we have no throughput / latency guarantees concerning the connections between local subgraphs; in the extreme case we even need to consider common internet & mobile connections. Thus, the principle challenge lies in propagating aggregated information per subgraph in such a way that GRL is still feasible from a performance & accuracy standpoint.

- We will therefore extend existing *neighborhood aggregation* techniques to a decentralized setting, selecting & experimenting with different *aggregation schemes*, some of which are rather trivial (mean, max, sum) but perform well according to literature; others can be complete neural networks in themselves, such as Long Short-Term Memory (LSTM) (Schmidhuber, n.d.), which were successfully used in *GraphSAGE* (Hamilton et al., 2017).

- Depending on its *size, domain or shape*, we might employ different aggregators per local subgraph in order to obtain fitting representations. In case this proves a successful strategy, we will explore the possibility of establishing heuristics of when to use which aggregator, enabling the network to optimize its aggregation strategy autonomously. These heuristics could incorporate domain knowledge or be derived by interpretations of reinforcement learning algorithms, which will be incorporated to maximize the overall reward of achieving this goal. The need for an explicit definition of predefined and rigid logic rules is thereby avoided, whereas the emergence of new strategies is enabled.

- Regarding LSTMs, a complementary challenge lies in explaining their good performance on what intuitively look like *permutation invariant* data, like the ordering of node neighbors in a graph. We conjecture that many graph-related learning problems could be modeled under the aspect of *information flow*, where the sequence of nodes propagating signals is decisive. However, this assumption would need to be tested in various real-world scenarios.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

8 Conclusion

In this report, we motivated the need for a novel, holistic and flexible approach towards distributed learning of graph-based data sets based on state-of-the-art Machine Learning research. We emphasized the need for learning embeddings – low dimensional, dense representations of concepts – from different sources and different modalities. Towards this end, we described a series of experiments designed to establish understanding and gaining experience as to how separated graphs modelling the same underlying domain behave in realistic learning tasks. We discussed the setup & initial phases of these experiments including results and their potential explanations, followed by an outline of future work which will seamlessly fuse into our overall research endeavours. Ultimately, the transition from centralized, cloud-based, untransparent services to a decentralized, secure & end-user-controlled future still confronts us with a multitude of challenges.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

9 References

Bengio, Y., Ducharme, R., & Vincent, P. (2001). A neural probabilistic language model. *Advances in Neural Information Processing Systems*, *3*, 1137–1155.

Bloice, M. D., Stocker, C., & Holzinger, A. (2017). Augmentor: An Image Augmentation Library for Machine Learning. *ArXiv*, *August 2017*. https://doi.org/10.21105/joss.00432

Choi, J. D., Tetreault, J., & Stent, A. (2015). It depends: Dependency parser comparison using a web-based evaluation tool. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 387–396. https://doi.org/10.3115/v1/p15-1038

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, *2015-Janua*, 2224–2232.

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 1025–1035.

Hu, Y., Niu, D., Yang, J., & Zhou, S. (2018). *Stochastic Distributed Optimization for Machine Learning from Decentralized Features*. http://arxiv.org/abs/1812.06415

Ivanov, S., & Burnaev, E. (2018). Anonymous walk embeddings. *35th International Conference on Machine Learning, ICML 2018*, *5*, 3448–3457.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, *2*, 427–431. https://doi.org/10.18653/v1/e17-2068

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 1–14.

Liu, S., Liu, S., Pujol, S., Kikinis, R., Feng, D., & Cai, W. (2014). Propagation graph fusion for multi-modal medical content-based retrieval. *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, *2014*(December), 849–854. https://doi.org/10.1109/ICARCV.2014.7064415

Malle, B., Giuliani, N., Kieseberg, P., & Holzinger, A. (2017). *The More the Merrier - Federated Learning from Local Sphere Recommendations* (Vol. 1). https://doi.org/10.1007/978-3-319-66808-6

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 1–9.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, *January*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Pham, T., Tran, T., Dam, H., & Venkatesh, S. (2017). Graph classification via deep learning with virtual nodes. *ArXiv*.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

Sattler, F., Wiedemann, S., Muller, K. R., & Samek, W. (2020). Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(9), 3400–3413. https://doi.org/10.1109/TNNLS.2019.2944481

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). Modeling Relational Data with Graph Convolutional Networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10843 LNCS*(1), 593–607. https://doi.org/10.1007/978-3-319-93417-4_38

Schmidhuber, J. (n.d.). *Lstm Can Solve Hard*.

Tong, T., Gray, K., Gao, Q., Chen, L., & Rueckert, D. (2017). Multi-modal classification of Alzheimer’s disease using nonlinear graph fusion. *Pattern Recognition*, *63*(October 2016), 171–181. https://doi.org/10.1016/j.patcog.2016.10.009

Veličković, P., Buesing, L., Overlan, M. C., Pascanu, R., Vinyals, O., & Blundell, C. (2020). Pointer Graph Networks. *ArXiv*.

Vivar, G., Burwinkel, H., Kazi, A., Zwergal, A., Navab, N., & Ahmadi, S.-A. (2019). *Multi-modal Graph Fusion for Inductive Disease Classification in Incomplete Datasets*. 1–9. http://arxiv.org/abs/1905.03053

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., & Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, *11*(3), 333–337. https://doi.org/10.1038/nmeth.2810

Wei, Y., He, X., Wang, X., Hong, R., Nie, L., & Chua, T. S. (2019). MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, 1437–1445. https://doi.org/10.1145/3343031.3351034

Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J. (2018). StarSpace: Embed all the things! *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 5569–5577.

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. *ArXiv*, *iii*.

Ying, R., Morris, C., Hamilton, W. L., You, J., Ren, X., & Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. *Advances in Neural Information Processing Systems*, *2018-Decem*, 4800–4810.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.

10 Appendix ESCO / ONET statistics

ESCO Statistics

Label	Node	HyperNode	Edge	Count	Sum (resp.)
Jobs	X			3561	
Skills	X			14158	**17719**
Broader Jobs			X	3551	
Broader Skills			X	21486	
Job -> Skills			X	114403	
Related Skills			X	5971	**145411**

ONET Statistics

Legend: ("WA" = work activity)

Label	Node	HyperNode	Edge	Count	Sum (resp.)
Intermediate WA	X			332	
Detailed WA	X			2067	
Jobs	X			1133	
Skills	X			35	
Tech Skills	X			127	
Abilities	X			52	
Activities	X			41	
Alternative Titles	X			45734	
Reported Titles	X			7502	
Knowledge items	X			33	
Tasks	X			20043	
Tools	X			4180	
Scales	X			29	**81308**
JobAbilities		X		50336	
JobActivities		X		39688	
JobSkills		X		33880	
JobKnowledge		X		31944	
JobTaskRating		X		19498	**175346**
Job -> Alternative Titles			X	58354	
Job -> Reported Titles			X	9213	
Job -> Tech Skills			X	29370	
Job -> Tasks			X	20047	
Job -> Tools			X	42278	
DWA -> IWA			X	2067	
Abilities -> Activities			X	381	
Broader Jobs			X	1110	
Skill -> Activities			X	232	
Task -> DWA			X	23091	
Scale -> Job Task Ratings			X	175482	**361625**

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 826078.