Graph-based knowledge representation and extraction from unstructured textual data using machine learning algorithms

Ph.D. Dissertation

Nikos Kanakaris

Department of Mechanical Engineering and Aeronautics
University of Patras

June 2023

Outline

Outline

- Basic concepts and preliminaries
  - What is a graph?
  - Text representations
  - NLP tasks
  - Embeddings
  - Graph mining
  - Why graphs for NLP?
  - Problem of existing approaches
  - Overall approach

Outline

- Basic concepts and preliminaries
  - What is a graph?
  - Text representations
  - NLP tasks
  - Embeddings
  - Graph mining
  - Why graphs for NLP?
  - Problem of existing approaches
  - Overall approach
- Theoretical research 1
- Theoretical research 2

Outline

- Basic concepts and preliminaries
  - What is a graph?
  - Text representations
  - NLP tasks
  - Embeddings
  - Graph mining
  - Why graphs for NLP?
  - Problem of existing approaches
  - Overall approach
- Theoretical research 1
- Theoretical research 2
- Showcase of 4 applications

Outline

- Basic concepts and preliminaries
  - What is a graph?
  - Text representations
  - NLP tasks
  - Embeddings
  - Graph mining
  - Why graphs for NLP?
  - Problem of existing approaches
  - Overall approach
- Theoretical research 1
- Theoretical research 2
- Showcase of 4 applications
- Summary and contribution
- Lessons learned
- Future work directions

What is a graph?

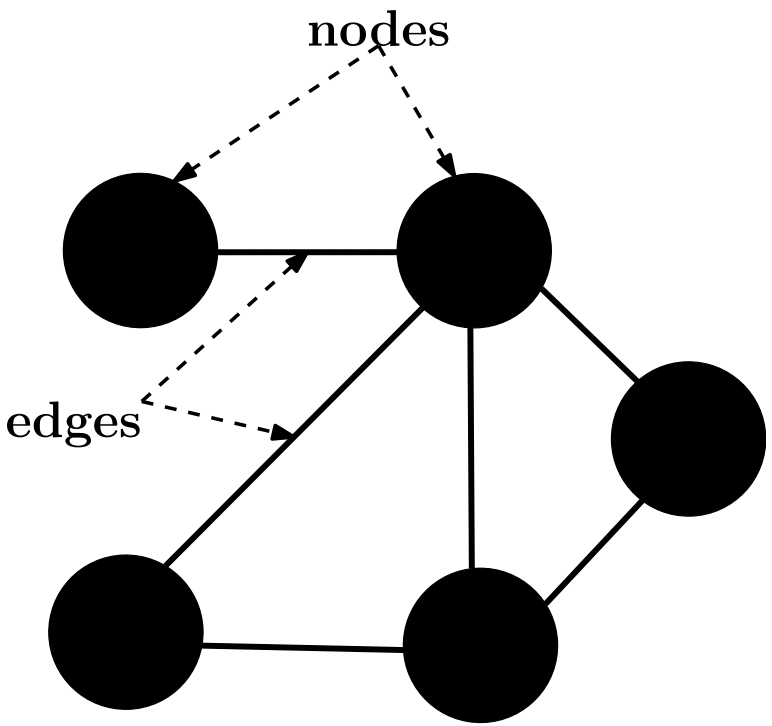
Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.

What is a graph?

Definition

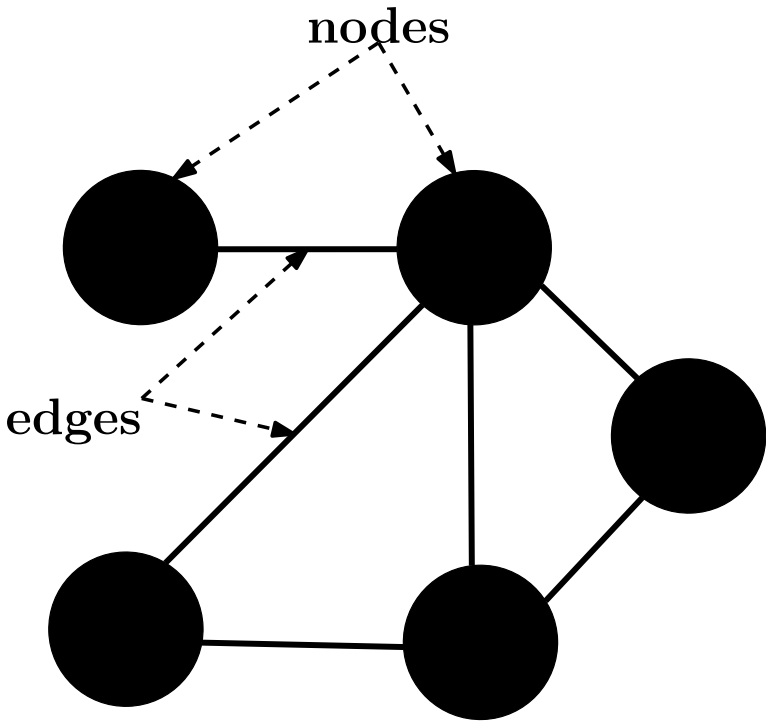
Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.

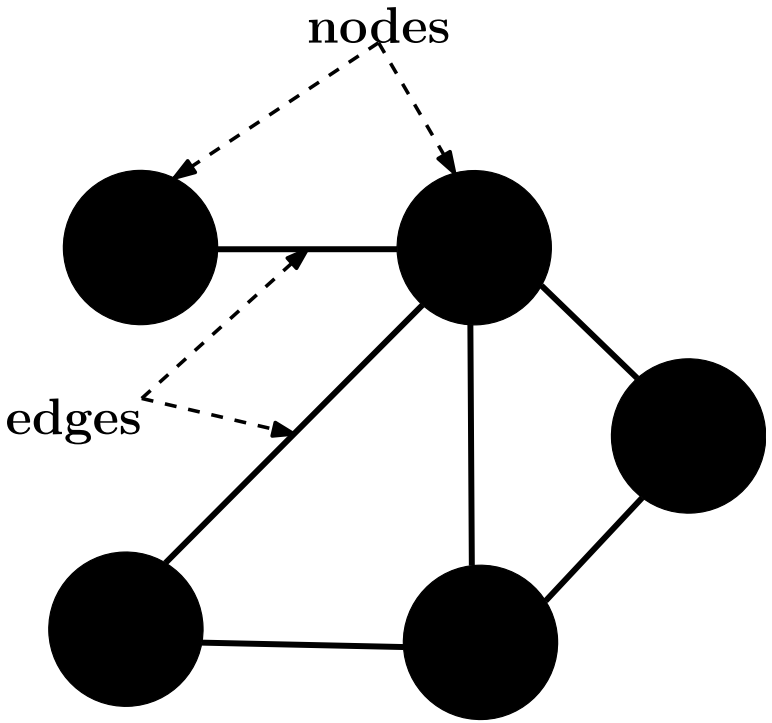


Types of graphs

What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



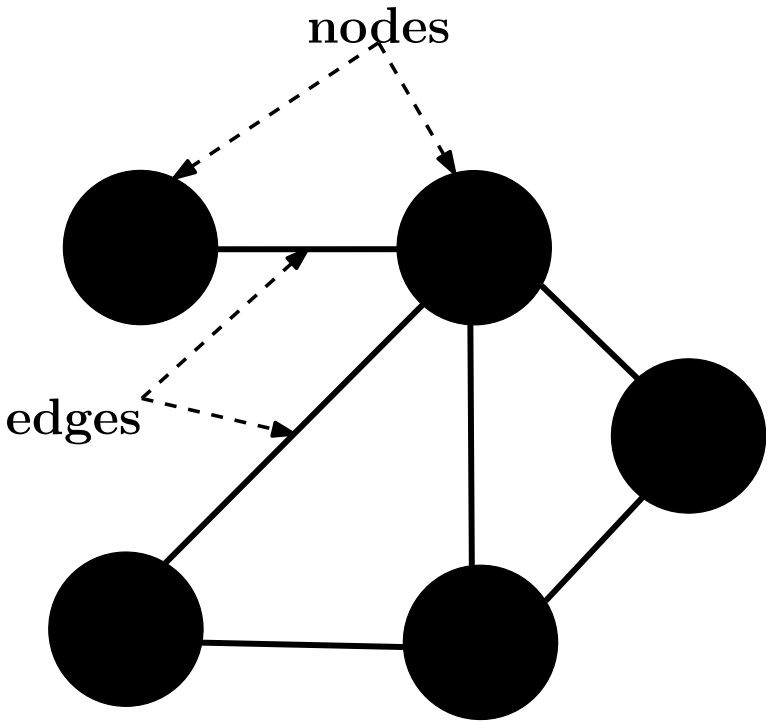
Types of graphs

- Homogeneous graphs

What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



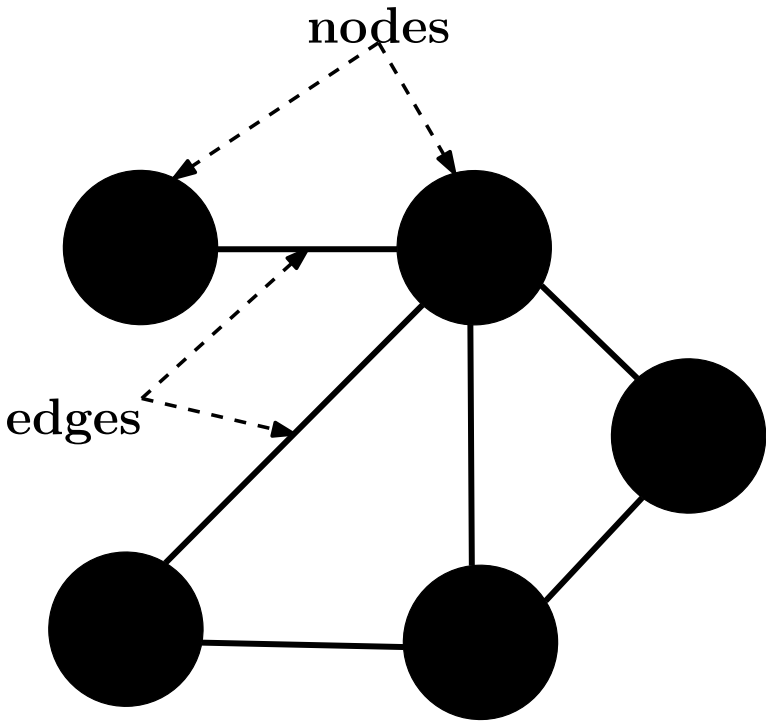
Types of graphs

- Homogeneous graphs
- Heterogeneous graphs
- Labeled graphs

What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



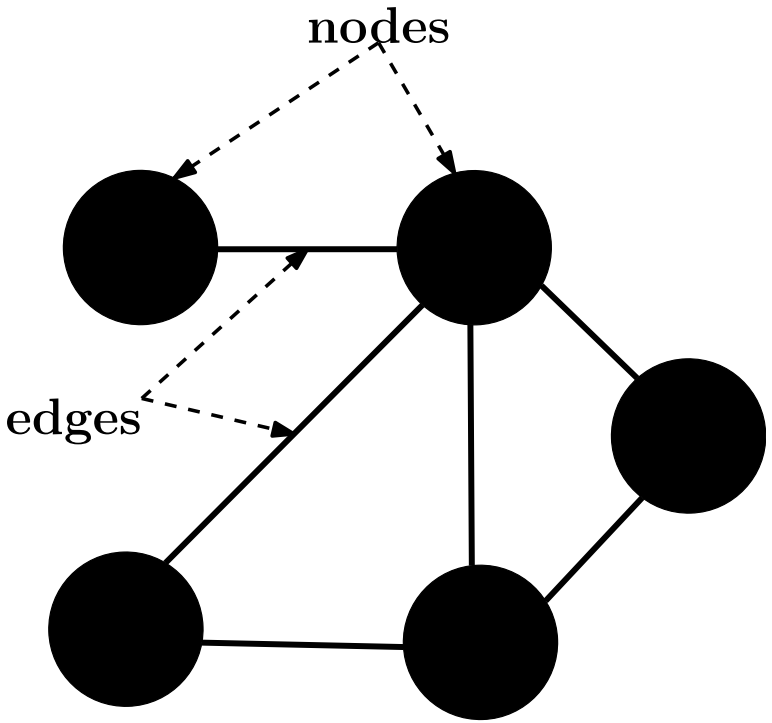
Types of graphs

- Homogeneous graphs
- Heterogeneous graphs
- Labeled graphs
- Directed graphs

What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



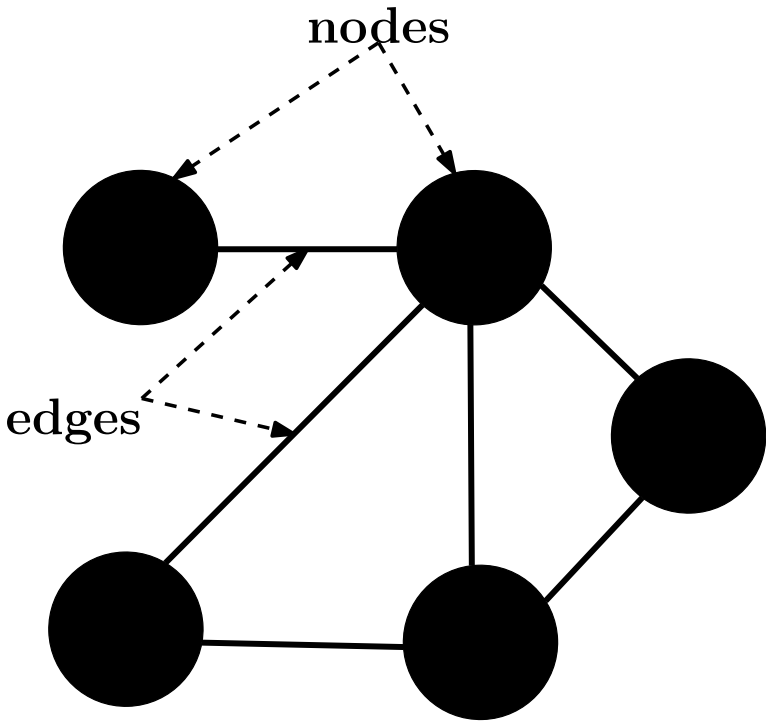
Types of graphs

- Homogeneous graphs
- Heterogeneous graphs
- Labeled graphs
- Directed graphs
- Weighted graphs

What is a graph?

Definition

Let a graph $G = (V, E)$ be defined as a tuple consisting of a set of vertices (or nodes) V and a set of edges $E \subseteq V \times V$.



Types of graphs

- Homogeneous graphs
- Heterogeneous graphs
- Labeled graphs
- Directed graphs
- Weighted graphs
- Attributed graphs

Text representations

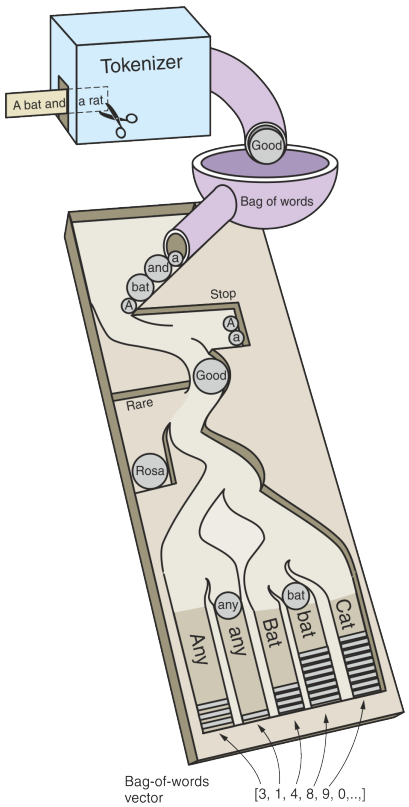
Definition

Numerically representing each unstructured textual document of a corpus to perform various Natural Language Processing (NLP) tasks.

Text representations

Definition

Numerically representing each unstructured textual document of a corpus to perform various Natural Language Processing (NLP) tasks.

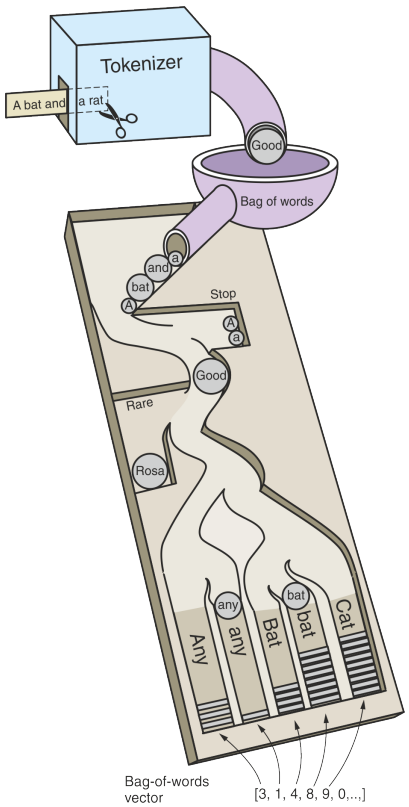


bag-of-words (source)

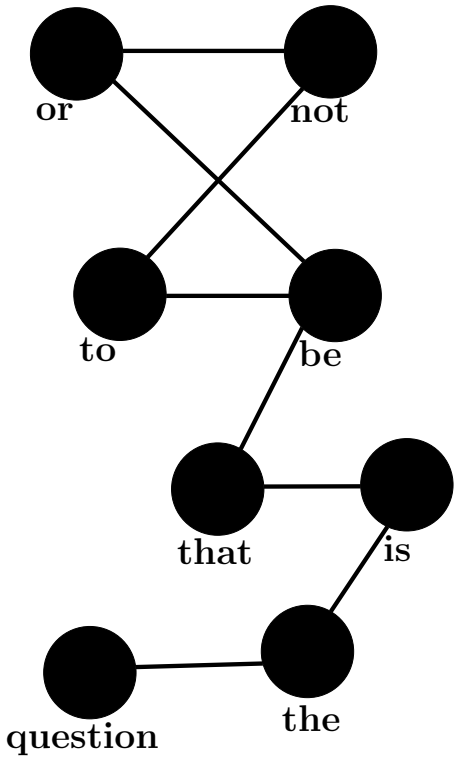
Text representations

Definition

Numerically representing each unstructured textual document of a corpus to perform various Natural Language Processing (NLP) tasks.



bag-of-words (source)

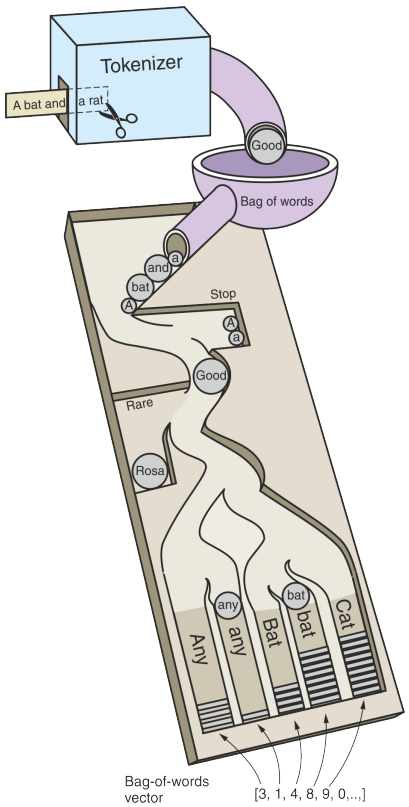


graph-of-words

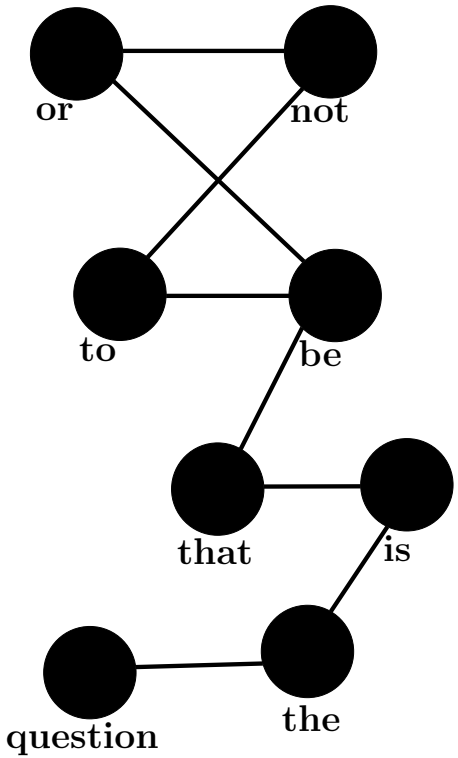
Text representations

Definition

Numerically representing each unstructured textual document of a corpus to perform various Natural Language Processing (NLP) tasks.



bag-of-words (source)



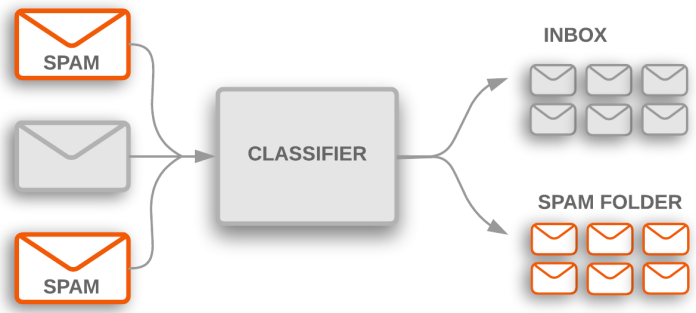
graph-of-words

Embeddings (source)

Popular NLP tasks

Text classification

- Categorizing textual data into predefined labels based on their content.
- Spam filtering
- $CLF : D \rightarrow y \in \{0, 1\}$

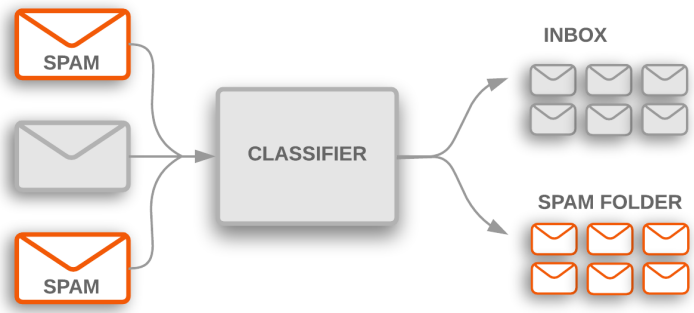


(source)

Popular NLP tasks

Text classification

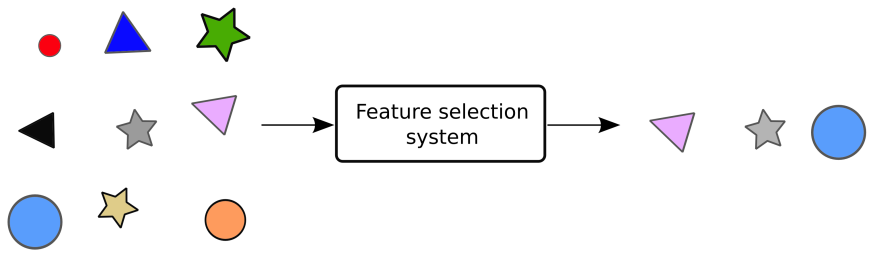
- Categorizing textual data into predefined labels based on their content.
- Spam filtering
- $CLF : D \rightarrow y \in \{0, 1\}$



(source)

Feature selection

- Selecting a subset of the important features.
- Simplification of models
- Shorter training times
- 'Curse of dimensionality'



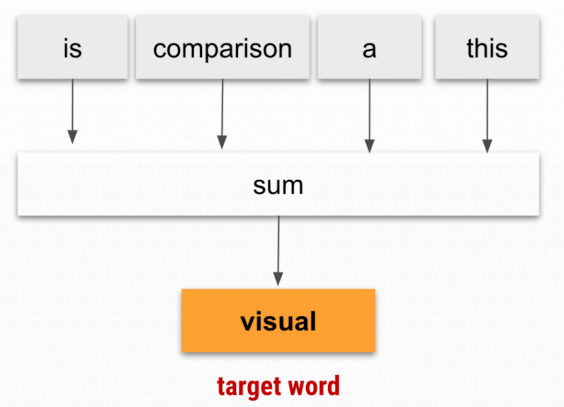
(source)

Word and document embeddings

- Converting documents or words to vectors
- Cosine similarity $\rightarrow cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$
- Useful for machine learning models
- word2vec
- fastText

Word and document embeddings

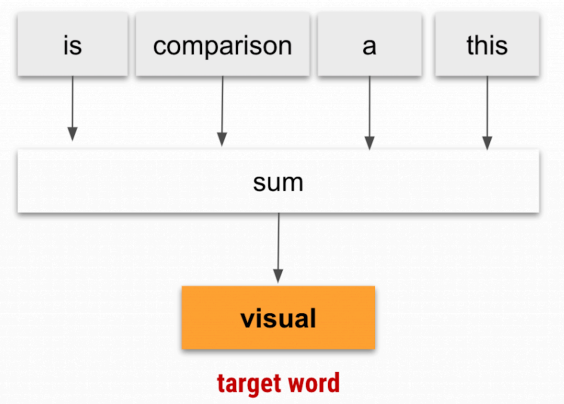
- Converting documents or words to vectors
- Cosine similarity $\rightarrow cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$
- Useful for machine learning models
- word2vec
- fastText



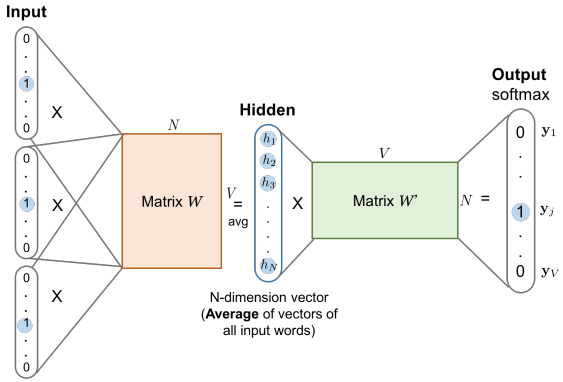
(source)

Word and document embeddings

- Converting documents or words to vectors
- Cosine similarity $\rightarrow cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$
- Useful for machine learning models
- word2vec
- fastText



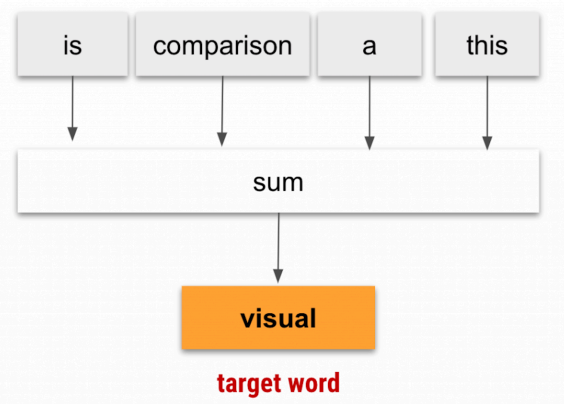
(source)



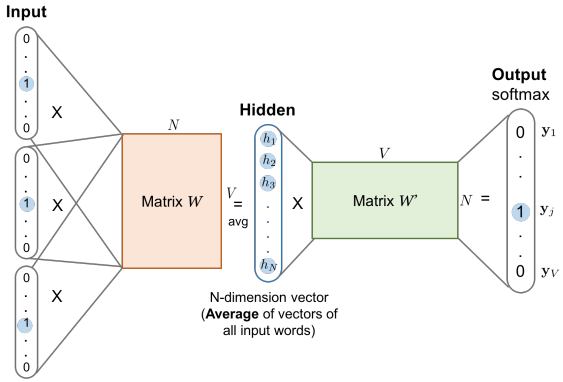
(source)

Word and document embeddings

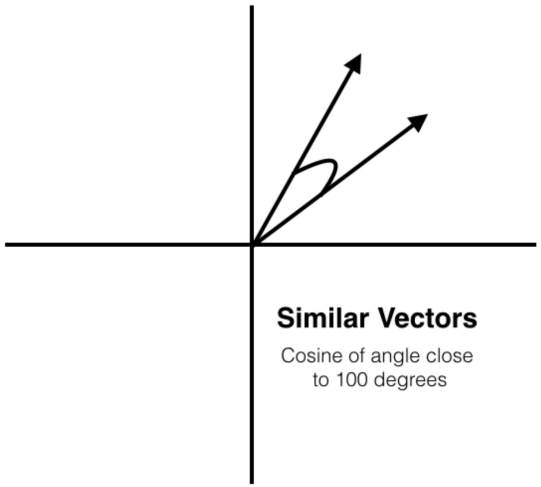
- Converting documents or words to vectors
- Cosine similarity $\rightarrow cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \|\vec{y}\|}$
- Useful for machine learning models
- word2vec
- fastText



(source)



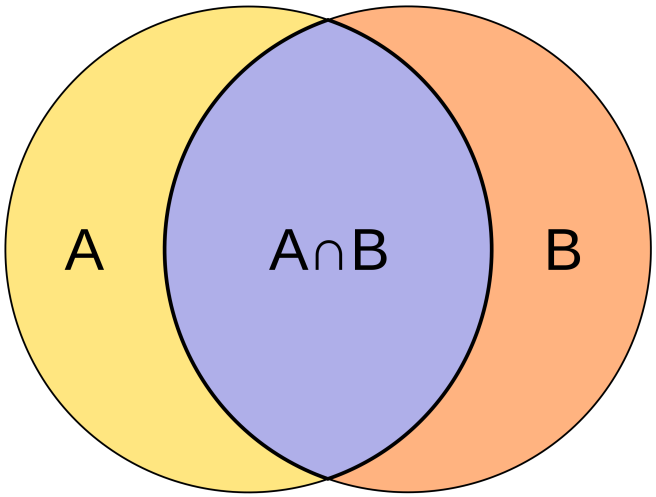
(source)



Classical graph mining

Graph measures and indices

- Centrality measures
- Node similarity
- Structural characteristics
- Jaccard Coefficient
- Common neighbors
- Adamic Adar



(source)

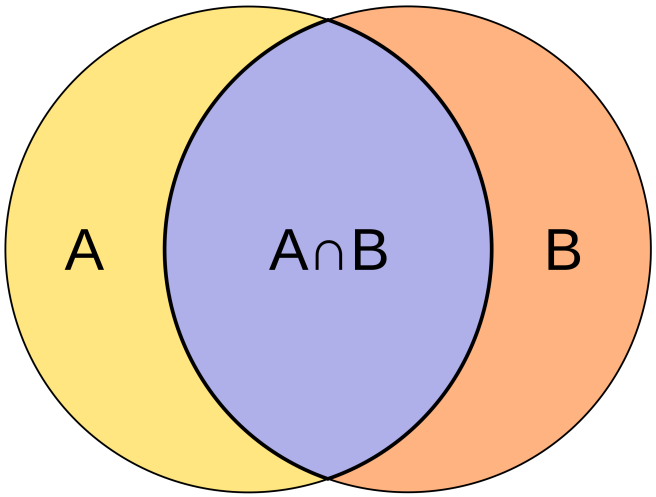
Classical graph mining

Graph measures and indices

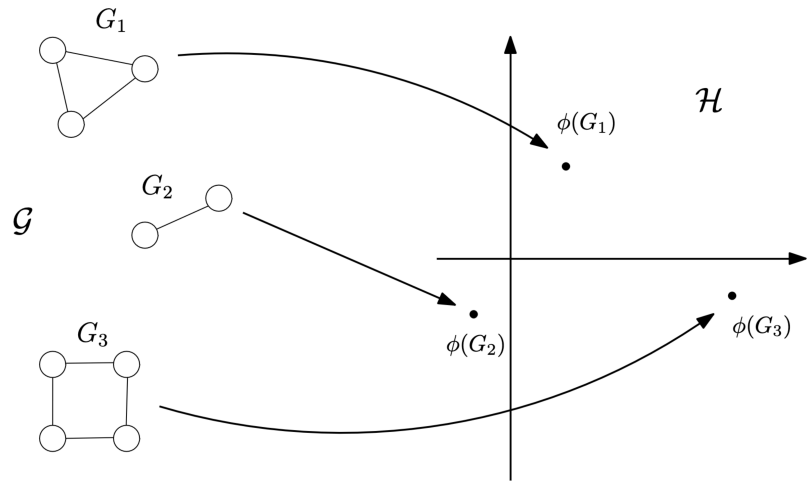
- Centrality measures
- Node similarity
- Structural characteristics
- Jaccard Coefficient
- Common neighbors
- Adamic Adar

Graph kernels

- Similarity between graphs
- Kernel-based algorithms
- Pyramid match
- Propagation kernel
- Weisfeiler Pyramid match
- Quadratic time

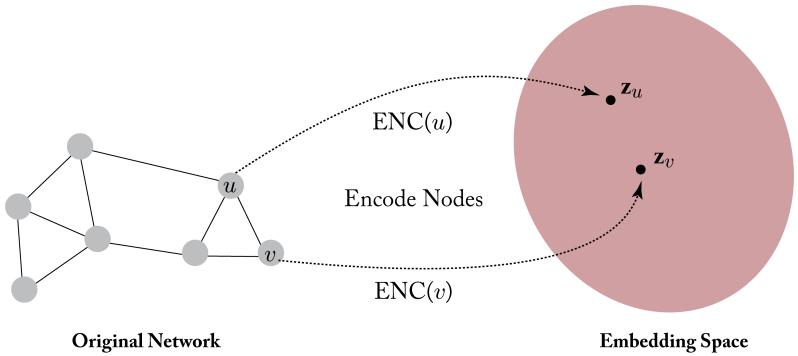


(source)



(source)

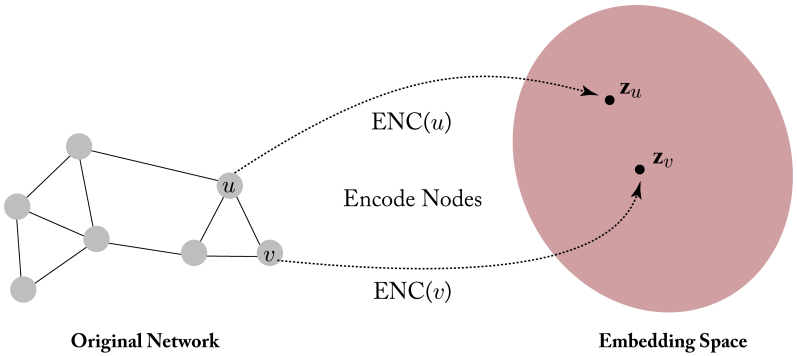
- $ENC : V \rightarrow \mathbb{R}^d$
- $DEC : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$
- Transductive
- Deepwalk



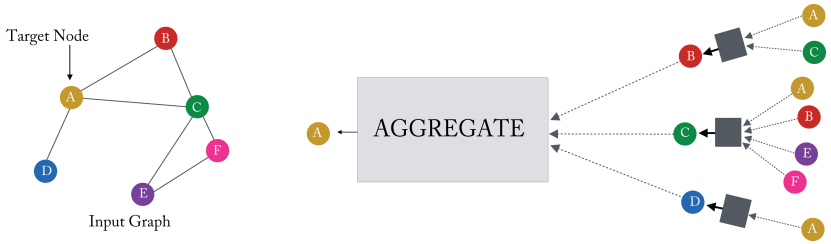
Shallow embeddings (source)

- $ENC : V \rightarrow \mathbb{R}^d$
- $DEC : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$
- Transductive
- Deepwalk

- Graph neural networks
- Features from neighbors
- Inductive
- GraphSAGE



Shallow embeddings (source)



'Deep' embeddings (source)

Why graphs for NLP?

Why graphs for NLP?

- Structural and semantic information

Why graphs for NLP?

- Structural and semantic information
- A variety of algorithms available

Why graphs for NLP?

- Structural and semantic information
- A variety of algorithms available
- Information from external knowledge sources

Why graphs for NLP?

- Structural and semantic information
- A variety of algorithms available
- Information from external knowledge sources
- Dense vector representations through representation learning on graphs

Problem of existing representations

Problem of existing representations

bag-of-words:

- Producing sparse feature vectors
- Suffering from the curse-of-dimensionality phenomenon
- Resulting in overfitted models

Problem of existing representations

bag-of-words:

- Producing sparse feature vectors
- Suffering from the curse-of-dimensionality phenomenon
- Resulting in overfitted models

graph-of-words:

- Incapable of assessing the importance of a word for the whole set of documents
- Does not allow for representing similarities between the documents
- Dealing only with the feature selection and feature extraction part of the whole NLP process

Overall approach

- Introduction of the graph-of-docs text representation

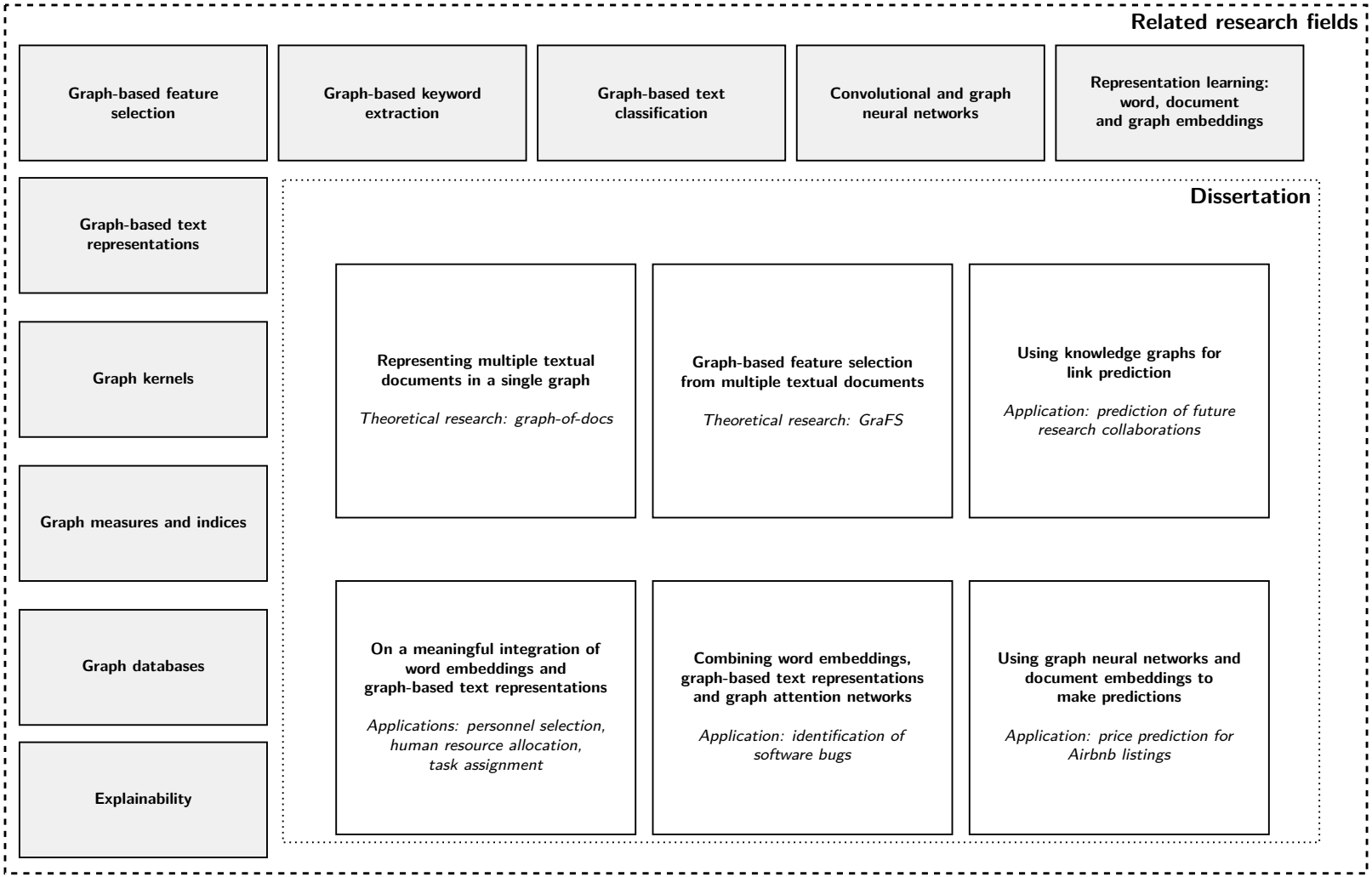
Overall approach

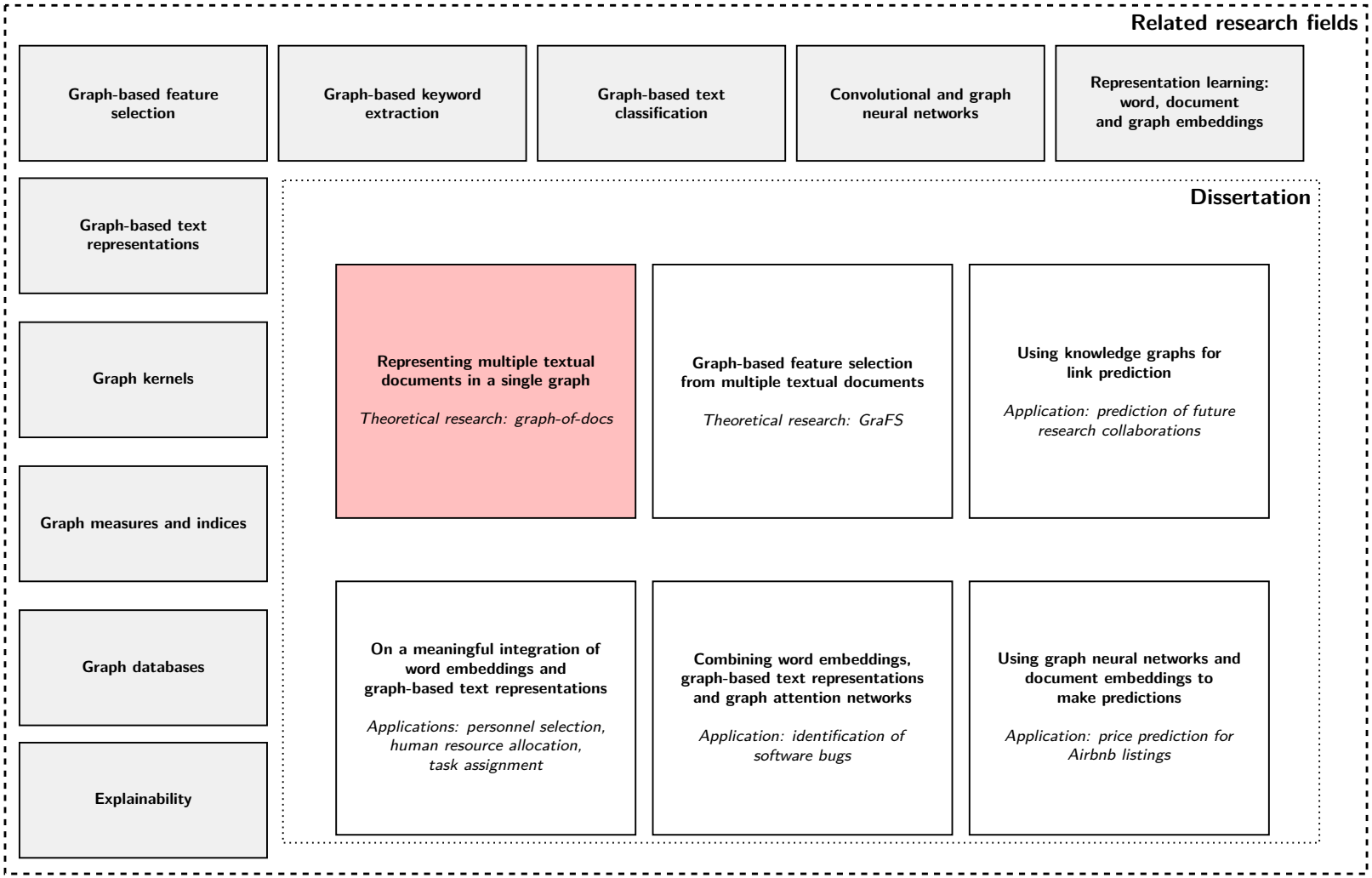
- Introduction of the graph-of-docs text representation
- Going beyond classical ML techniques
- Adopting a hybrid approach
  - word embeddings
  - graph-based text representations
  - graph neural networks

Overall approach

- Introduction of the graph-of-docs text representation
- Going beyond classical ML techniques
- Adopting a hybrid approach
  - word embeddings
  - graph-based text representations
  - graph neural networks
- Advancing classical ML tasks
  - Text classification
  - Feature engineering
  - Feature selection

Overview and contributions of the dissertation




Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Graph of docs text representation

We propose:

Graph of docs text representation

We propose:

- representing multiple textual documents as a **single graph**

Graph of docs text representation

We propose:

- representing multiple textual documents as a **single graph**
- enabling the investigation of the **importance** of a term into a **whole corpus** of documents

Graph of docs text representation

We propose:

- representing multiple textual documents as a **single graph**
- enabling the investigation of the **importance** of a term into a **whole corpus** of documents
- masking the overall complexity by **reducing** each graph-of-words to a 'document' **node**

Graph of docs text representation

We propose:

- representing multiple textual documents as a **single graph**
- enabling the investigation of the **importance** of a term into a **whole corpus** of documents
- masking the overall complexity by **reducing** each graph-of-words to a 'document' **node**
- enabling the calculation of important **metrics** concerning the **documents**

Our Approach: Graph of Docs

Schema (1/2)

Graph details

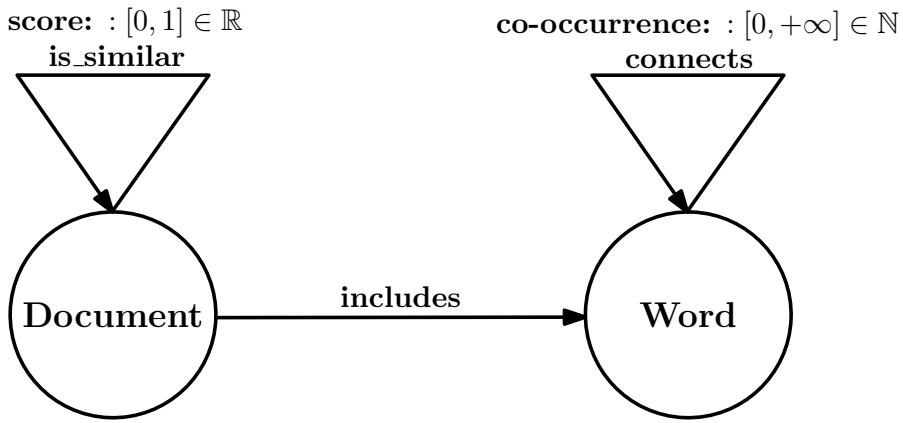
- Expands the graph-of-words model
- Directed dense graph
- Connections between words and documents

Allowed types of nodes:

- Document
- Word

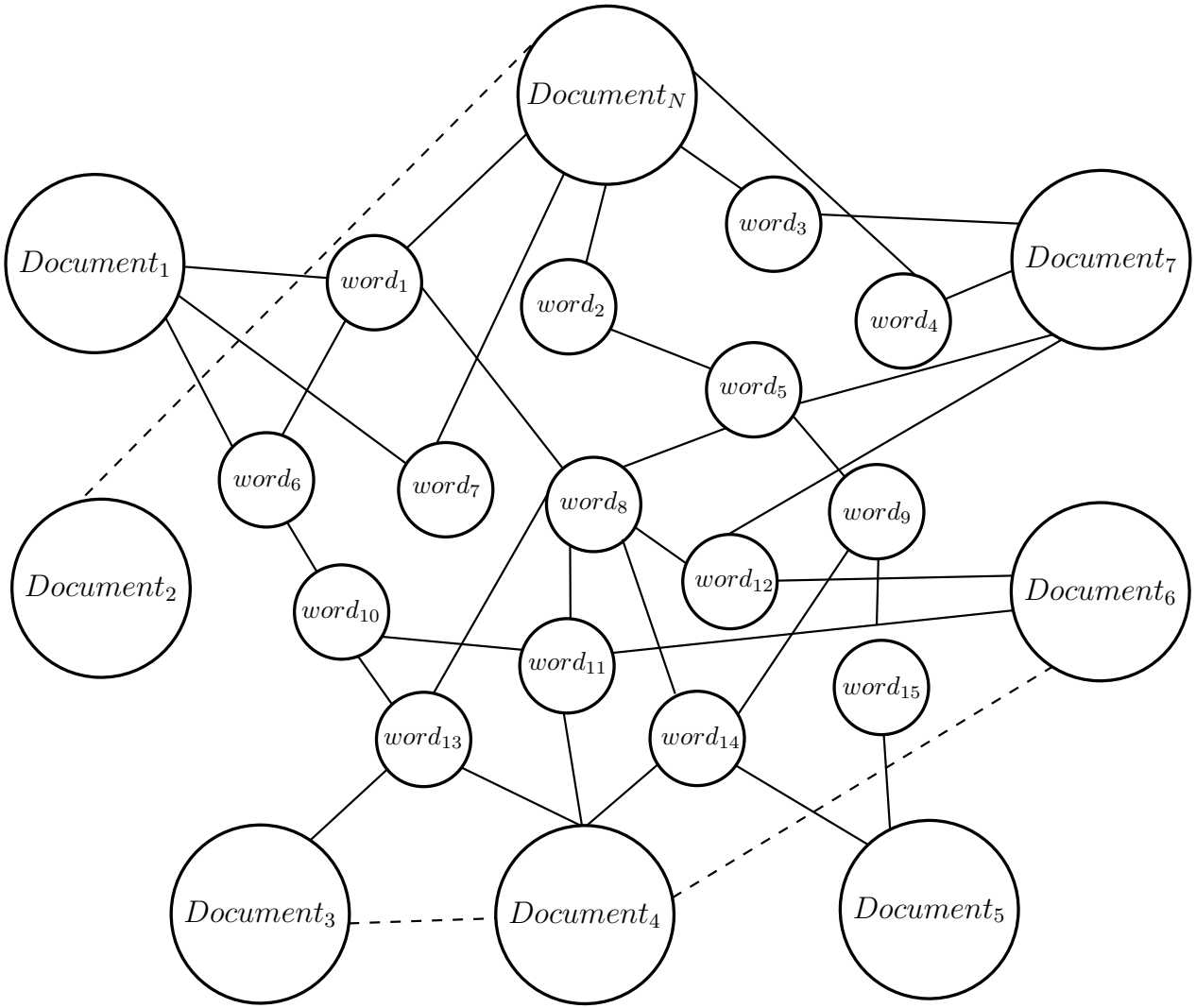
Allowed types of edges:

- includes
- connects (co-occurrence property)
- is_similar (score property)



Our Approach: Graph of Docs

Schema (2/2)



Steps

Data preprocessing:

- Stopwords removal
- List of terms production

Experiments: Graph of Docs

Implementation

Steps

Data preprocessing:

- Stopwords removal
- List of terms production

Nodes generation:

- Terms to nodes conversion
- Co-occurrence score calculation

Experiments: Graph of Docs

Implementation

Steps

Data preprocessing:

- Stopwords removal
- List of terms production

Nodes generation:

- Terms to nodes conversion
- Co-occurrence score calculation

Influential nodes identification:

- PageRank algorithm

Steps

Data preprocessing:

- Stopwords removal
- List of terms production

Nodes generation:

- Terms to nodes conversion
- Co-occurrence score calculation

Influential nodes identification:

- PageRank algorithm

Document similarity subgraph generation:

- Jaccard similarity
- Communities of similar nodes (Louvain algorithm)

Experiments

Dataset

- A collection of approximately 20,000 newsgroup documents
- 20 different newsgroups (document classes)
- A popular dataset for experiments in text classification and text clustering

Experiments

Evaluation Results

Text classifier	Accuracy
5-NN	54.8%
2-NN	61.0%
1-NN	76.0%
naive Bayes	93.7%
logistic regression	93.9%
neural network (100x50)	95.5%
neural network (1000x500)	95.9%
graph-of-docs classifier	97.5%

Experiments

Evaluation Results

Text classifier	Accuracy
5-NN	54.8%
2-NN	61.0%
1-NN	76.0%
naive Bayes	93.7%
logistic regression	93.9%
neural network (100x50)	95.5%
neural network (1000x500)	95.9%
graph-of-docs classifier	97.5%

Pros

- **Multiple** textual documents in **single** graph
- NLP tasks $\rightarrow$ well-studied **graph theory** problems
- Generation of document similarity **subgraphs**
- Importance of a word within a **collection** of documents

Experiments

Evaluation Results

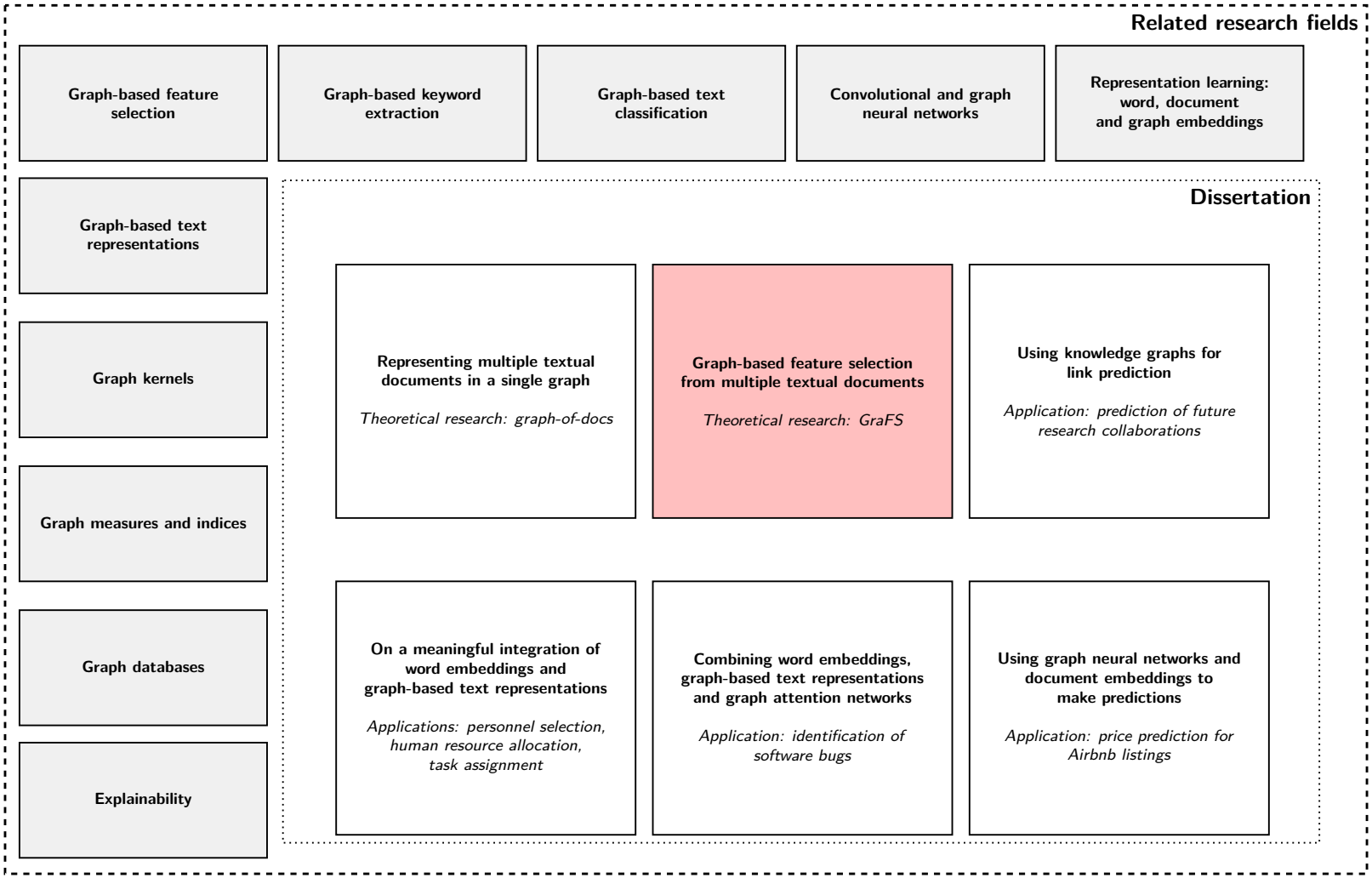
Text classifier	Accuracy
5-NN	54.8%
2-NN	61.0%
1-NN	76.0%
naive Bayes	93.7%
logistic regression	93.9%
neural network (100x50)	95.5%
neural network (1000x500)	95.9%
graph-of-docs classifier	97.5%

Pros

- **Multiple** textual documents in **single** graph
- NLP tasks $\rightarrow$ well-studied **graph theory** problems
- Generation of document similarity **subgraphs**
- Importance of a word within a **collection** of documents

Cons

- Poorly performing with **outlier** documents
- Performance issues


Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Feature selection

Existing feature selection techniques

- **traditional statistical methods**: Rely heavily on statistical tests to identify important features.
- **graph-based methods**: Combine statistical tests and graph algorithms to uncover hidden correlations. They also take into account the co-occurrences between terms.

Problem of existing feature selection techniques

traditional statistical methods:

- Unable to capture the structural characteristics of a document
- Suffer from the curse-of-dimensionality phenomenon
- Result in overfitted models

graph-based feature selection techniques:

- Represent each document of a corpus a single graph
- Incapable of assessing the importance of a word for the whole set of documents
- Allow only one node type to exist (i.e. word nodes)

Feature selection

Existing feature selection techniques

- **traditional statistical methods**: Rely heavily on statistical tests to identify important features.
- **graph-based methods**: Combine statistical tests and graph algorithms to uncover hidden correlations. They also take into account the co-occurrences between terms.

Problem of existing feature selection techniques

traditional statistical methods:

- Unable to capture the structural characteristics of a document
- Suffer from the curse-of-dimensionality phenomenon
- Result in overfitted models

graph-based feature selection techniques:

- Represent each document of a corpus a single graph
- Incapable of assessing the importance of a word for the whole set of documents
- Allow only one node type to exist (i.e. word nodes)

Our Solution: GraFS

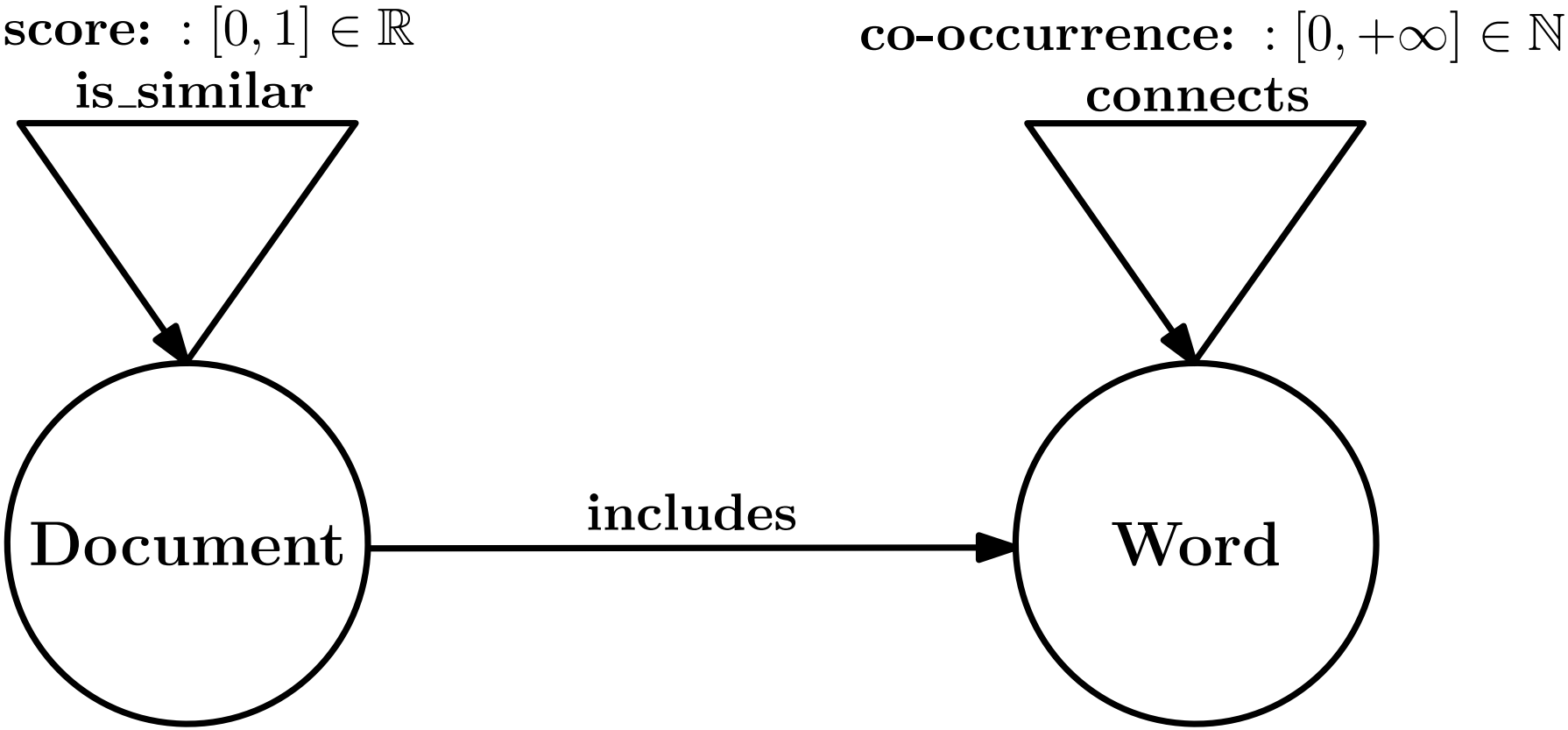
- Represents multiple textual documents as a single graph, based on graph-of-docs model
- Enables the investigation of the importance of a term into a whole corpus of documents
- Allows multiple node types to co-exist

Our Approach: Graph-based Feature Selection (GraFS)

- Builds on graph-of-docs
- Expands the graph-of-docs schema by adding the 'feature' type of edge
- Consists of four steps:
  - Creation of a document similarity subgraph
  - Detection of document communities
  - Feature selection for **each community**
  - Feature selection for the **whole corpus** of documents

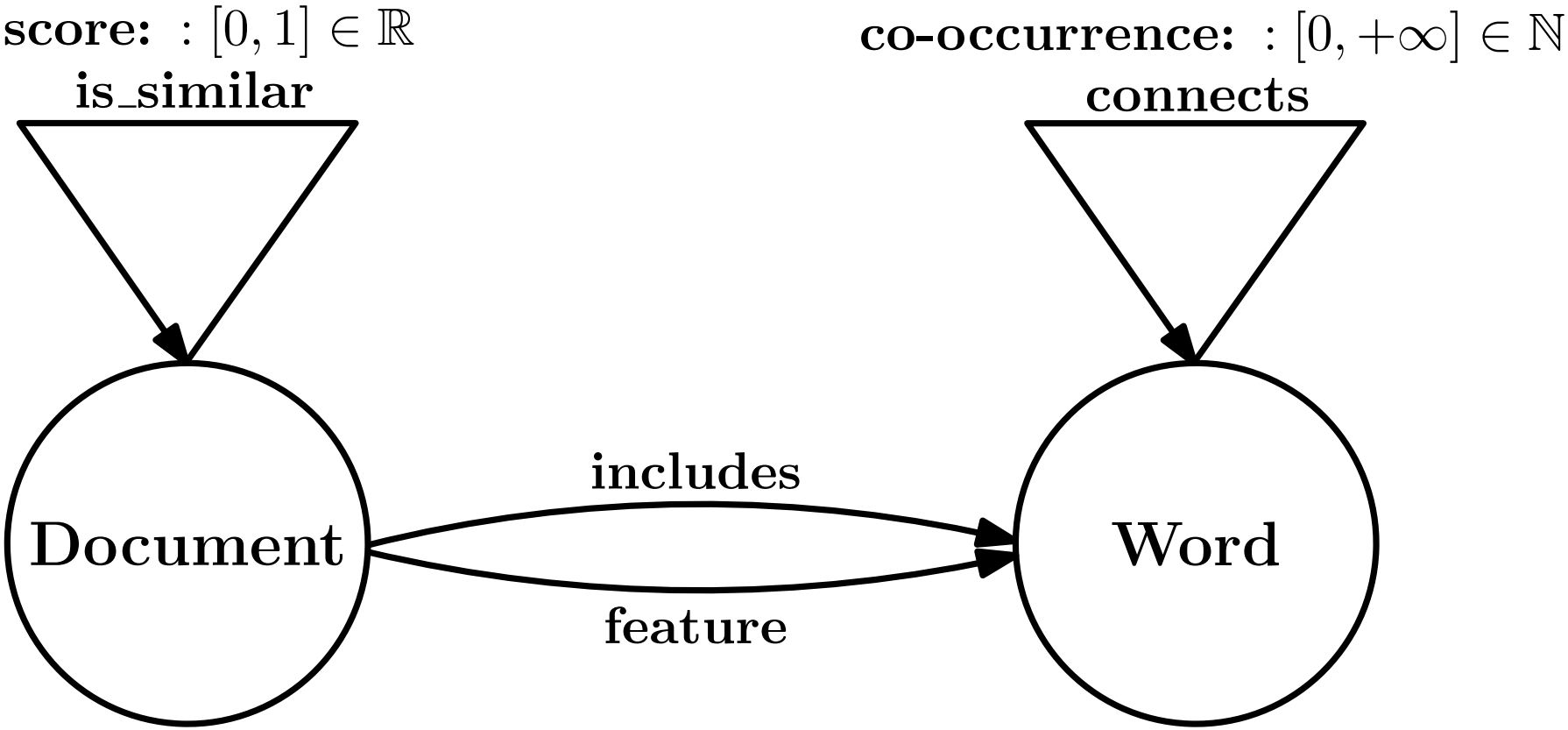
Our Approach: Graph-based Feature Selection (GraFS)

Graph database schema (1/2)



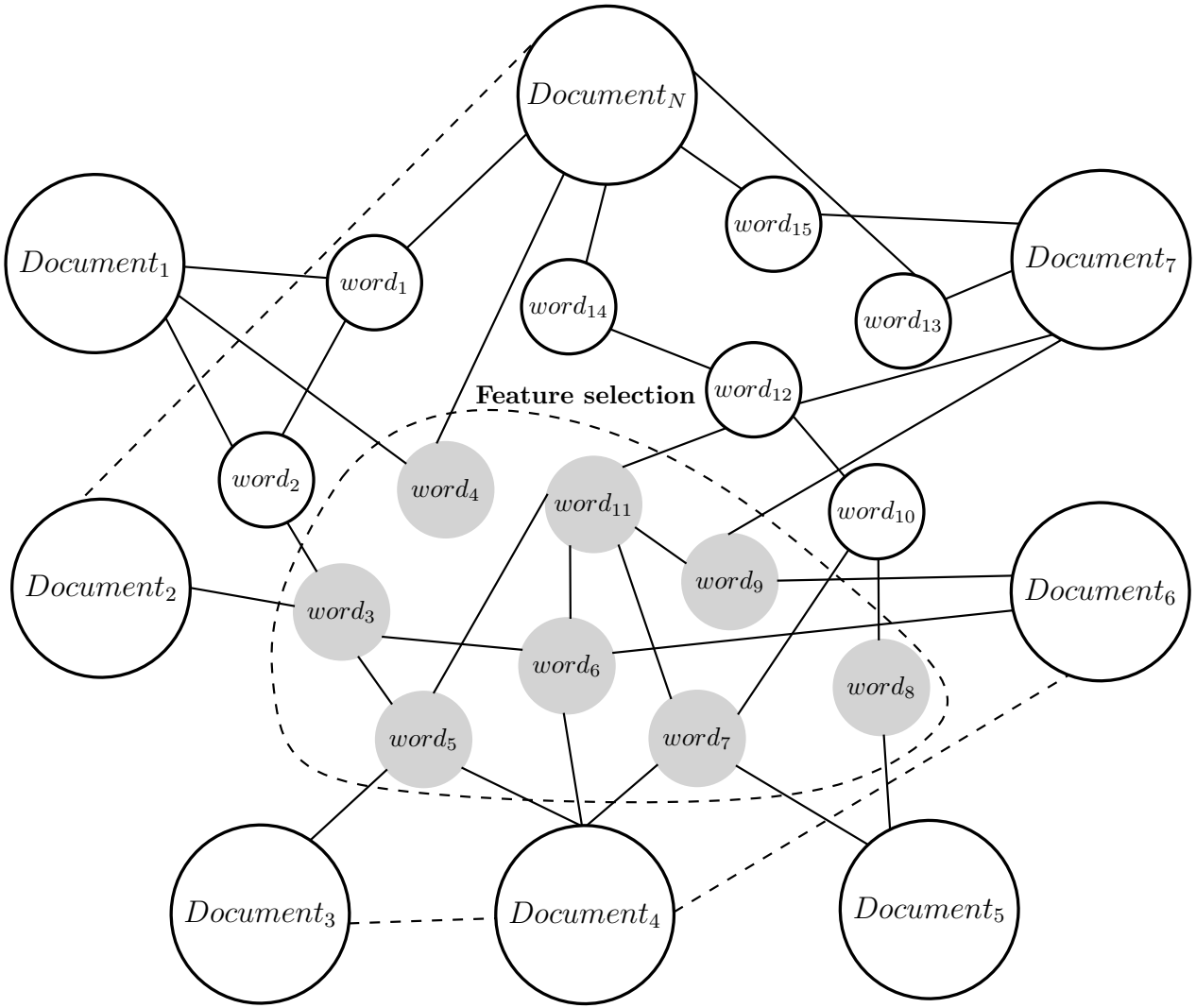
Our Approach: Graph-based Feature Selection (GraFS)

Graph database schema (1/2)



Our Approach: Graph-based Feature Selection (GraFS)

Graph database schema (2/2)



Our Approach: Graph-based Feature Selection (GraFS)

GraFS steps (1/2)

Creation of a document similarity subgraph

- **Similar** documents share common **words** and **structural** characteristics
- Calculates document similarity:
  - Similarity measures
  - Frequent subgraph mining techniques
  - Graph kernels
- Consists of document **nodes** and **edges** of the 'is_similar' type

Our Approach: Graph-based Feature Selection (GraFS)

GraFS steps (1/2)

Creation of a document similarity subgraph

- **Similar** documents share common **words** and **structural** characteristics
- Calculates document similarity:
  - Similarity measures
  - Frequent subgraph mining techniques
  - Graph kernels
- Consists of document **nodes** and **edges** of the 'is_similar' type

Detection of document communities

- Detects **communities** of contextually **similar** documents
- Utilizes the **'score'** property of the **'is_similar'** edges as a distance value
- **Community detection** algorithms (e.g. Louvain, Label Propagation and Weakly Connected Components)

Our Approach: Graph-based Feature Selection (GraFS)

GraFS steps (2/2)

Feature selection for each community

- Documents in the **same** community $\rightarrow$ contextually **similar**
- Documents in the **same** community $\rightarrow$ share common **features**
- GraFS ranks the **terms** of each community by:
  - document frequency
  - PageRank score
- Selects the **top-N terms** of each community

Our Approach: Graph-based Feature Selection (GraFS)

GraFS steps (2/2)

Feature selection for each community

- Documents in the **same** community $\rightarrow$ contextually **similar**
- Documents in the **same** community $\rightarrow$ share common **features**
- GraFS ranks the **terms** of each community by:
  - document frequency
  - PageRank score
- Selects the **top-N terms** of each community

Feature selection for the whole corpus of documents

- Defines the **feature space**
- **Merges** the top-N features of each community
- Reduces the number of the candidate features
- **Accelerates** the feature selection process
- Mitigates the effects of the 'curse-of-dimensionality' phenomenon
- Enables the training of more **reliable** ML models

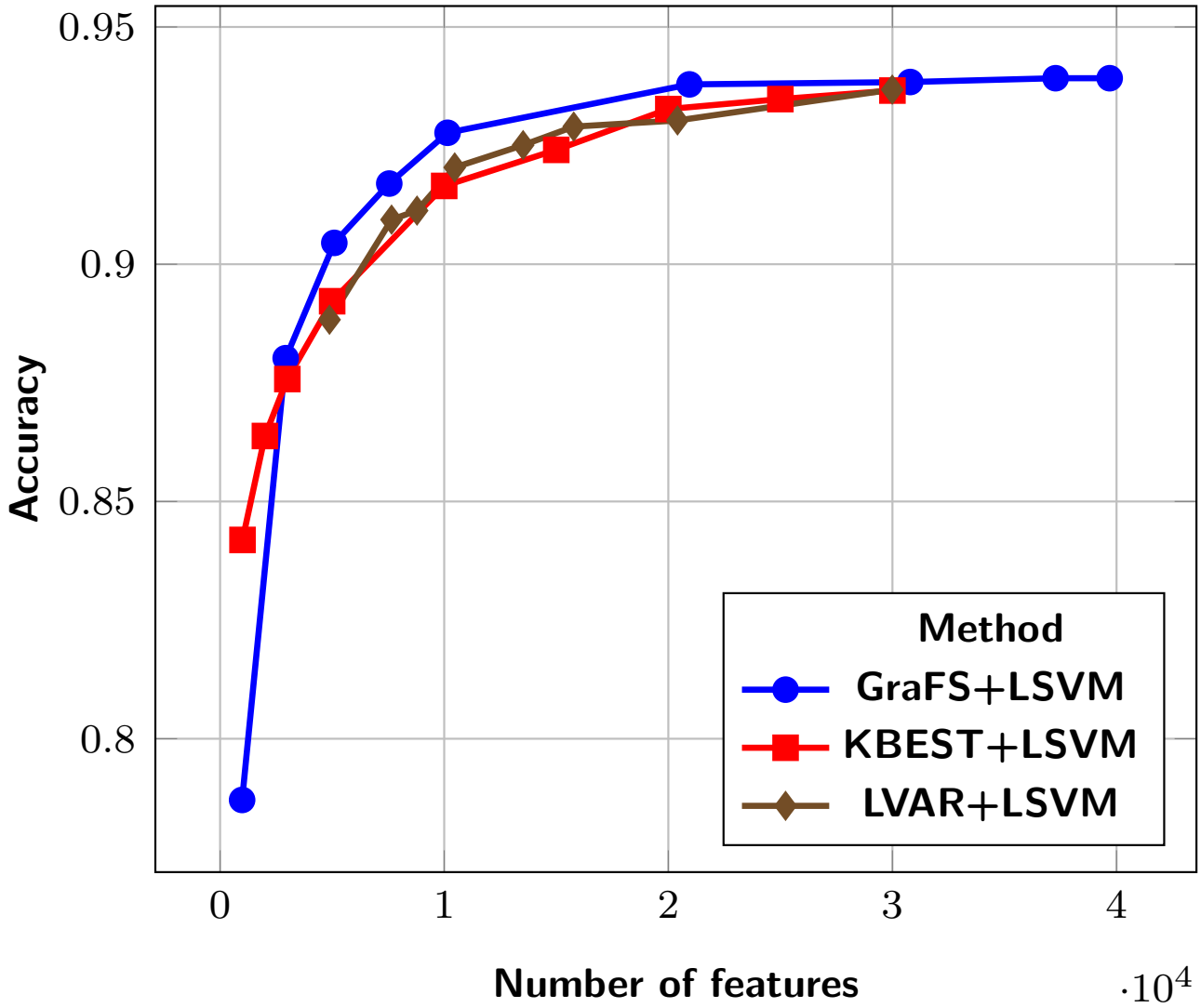
Experiments

Dataset

- Five datasets:
  - **20 Newsgroups** (20,000 newsgroup documents; multi-class text classification)
  - **Reuters** (21,578 news stories; multi-class text classification)
  - **Amazon Reviews** (2000 reviews; opinion mining)
  - **LingSpam** (2,893 email messages; spam detection)
  - **JiraIssues** (228,969 descriptions of Jira issues; multi-class text classification)

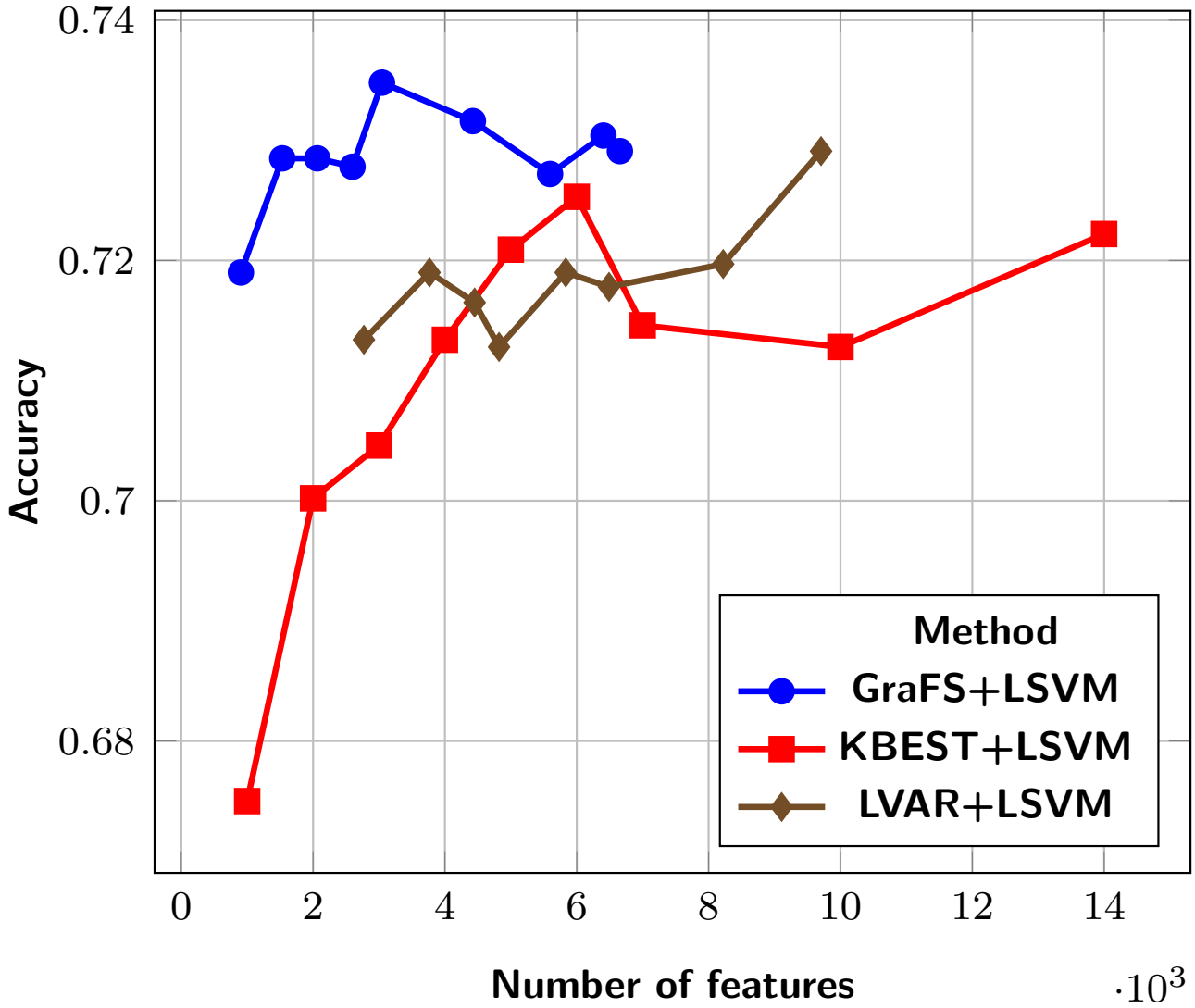
Evalutation results

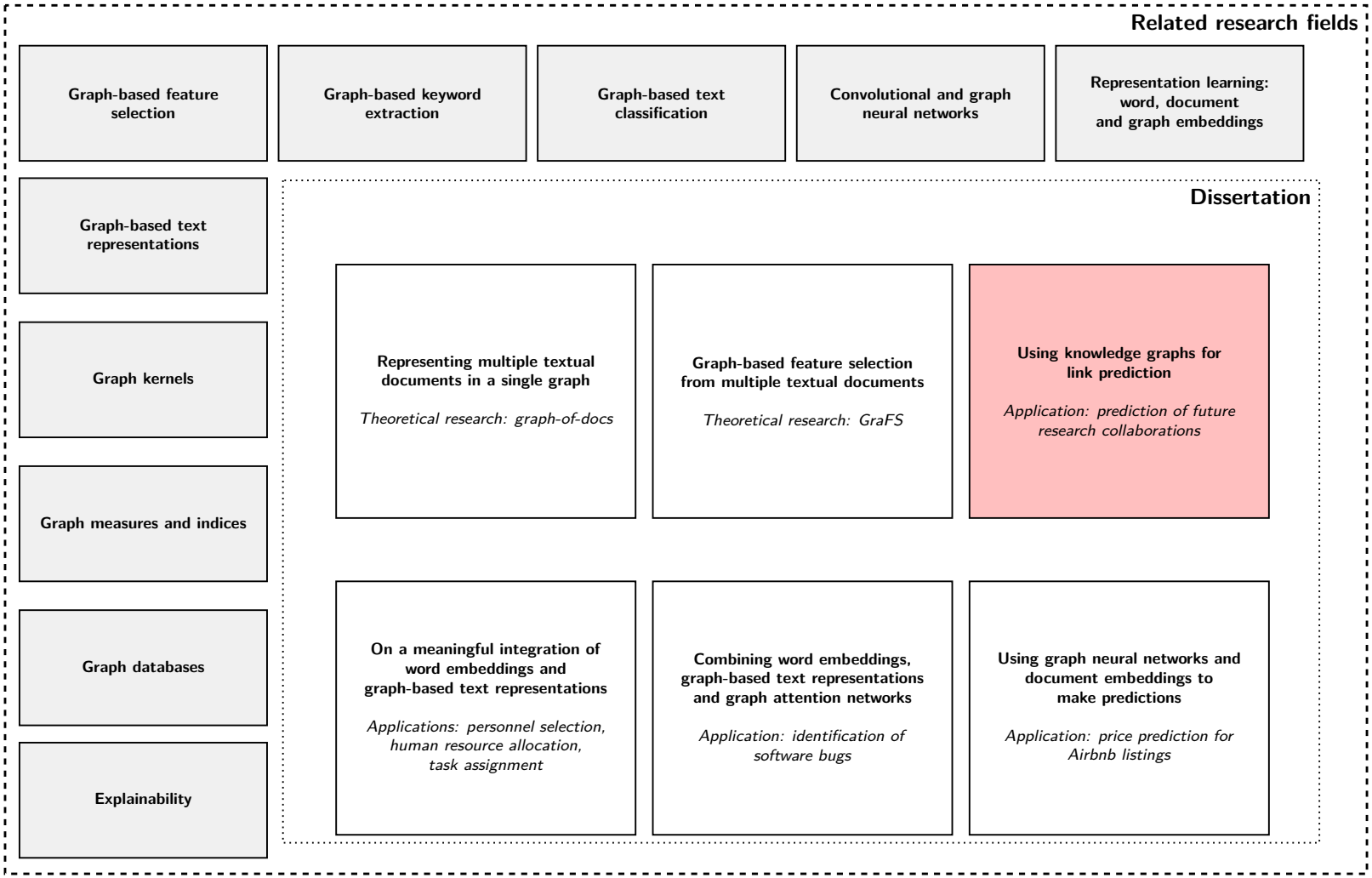
Accuracy of the LSVM classifier on the 20 Newsgroups dataset



Evalutation results

Accuracy of the LSVM classifier on the JiraIssues dataset




Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Discovering future research collaborations

Definition

The task of predicting future research collaborations using scholar and scientific data.

Existing future research collaborations discovery techniques

- Graph-based approaches
- Construct a research graph
- Build on concepts and methods from graph theory
- Utilize the structural chracteristics of the research graph

Problems of existing future research collaborations discovery techniques

- Utilize only the structural characteristics of a research graph
- Incapable of exploiting both the structural and the textual information of the graph

Discovering future research collaborations

Definition

The task of predicting future research collaborations using scholar and scientific data.

Existing future research collaborations discovery techniques

- Graph-based approaches
- Construct a research graph
- Build on concepts and methods from graph theory
- Utilize the structural chracteristics of the research graph

Problems of existing future research collaborations discovery techniques

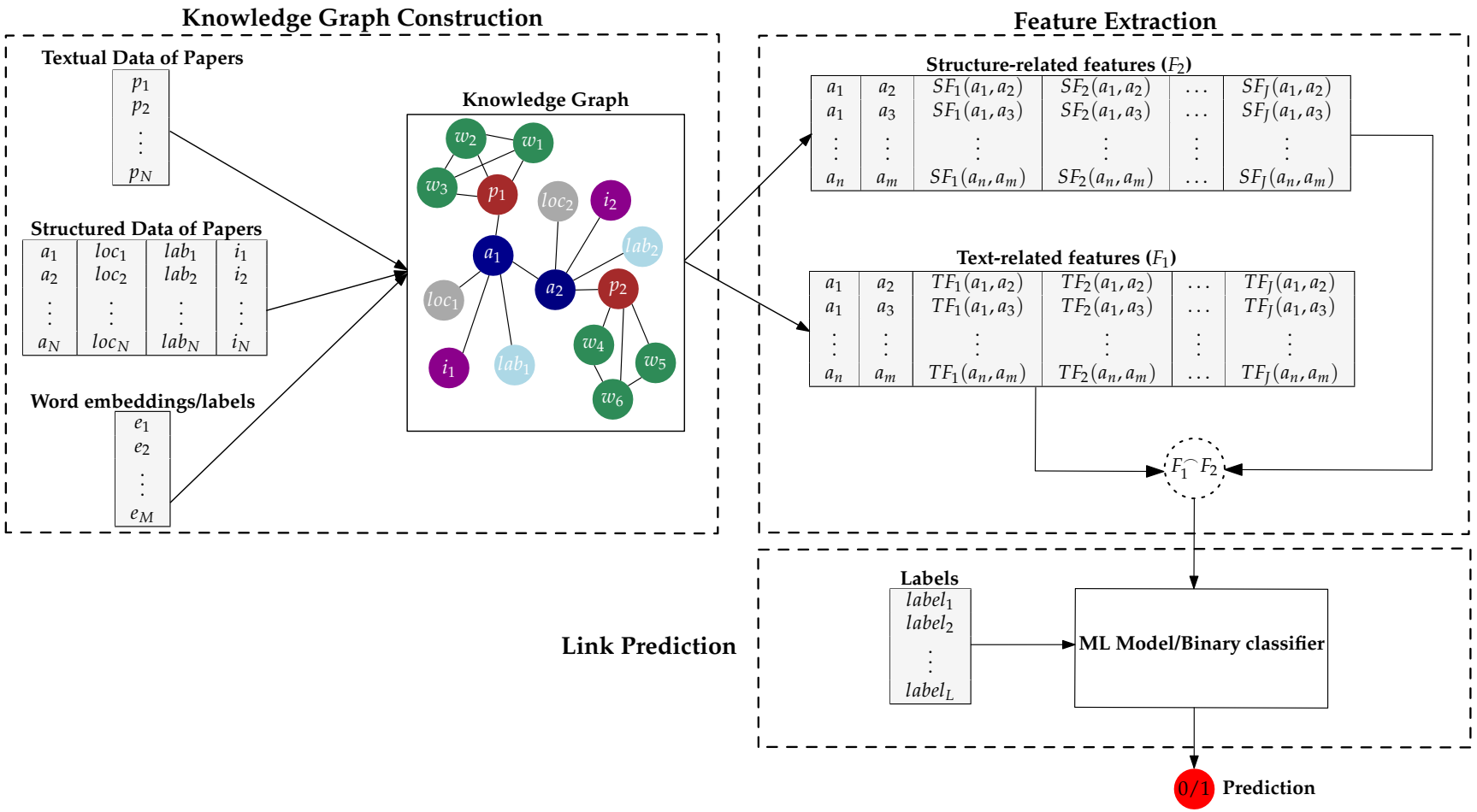
- Utilize only the structural characteristics of a research graph
- Incapable of exploiting both the structural and the textual information of the graph

Our approach

- Proposes the construction of a scientific knowledge graph
- Allows structured and unstrctured data to co-exist (e.g. document, author and word nodes)
- Enables the utilization of both structural and textual characteristics of the graph

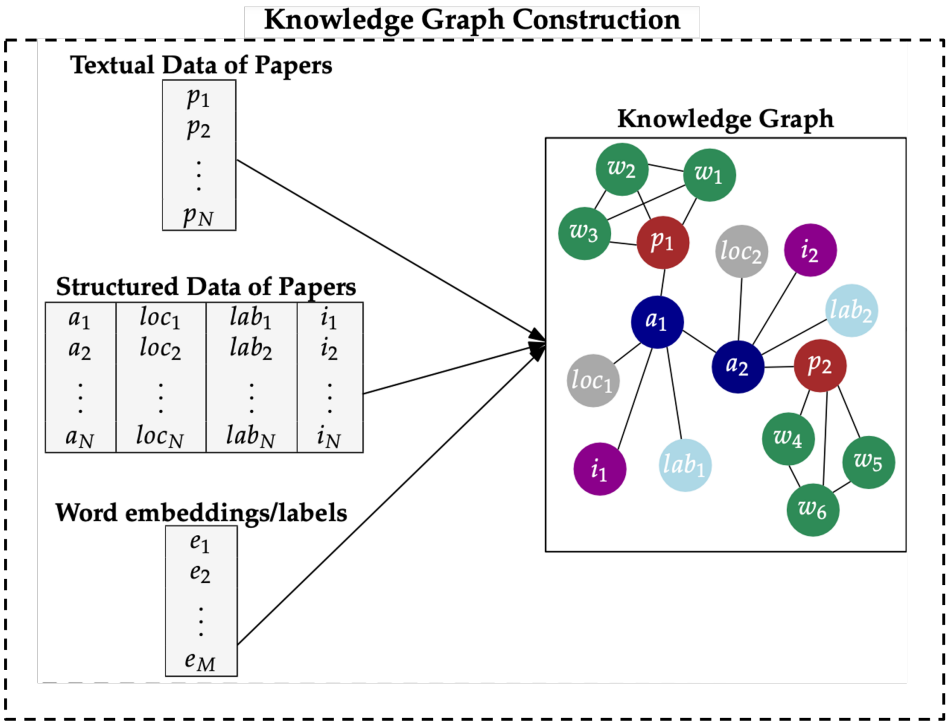
Our Approach

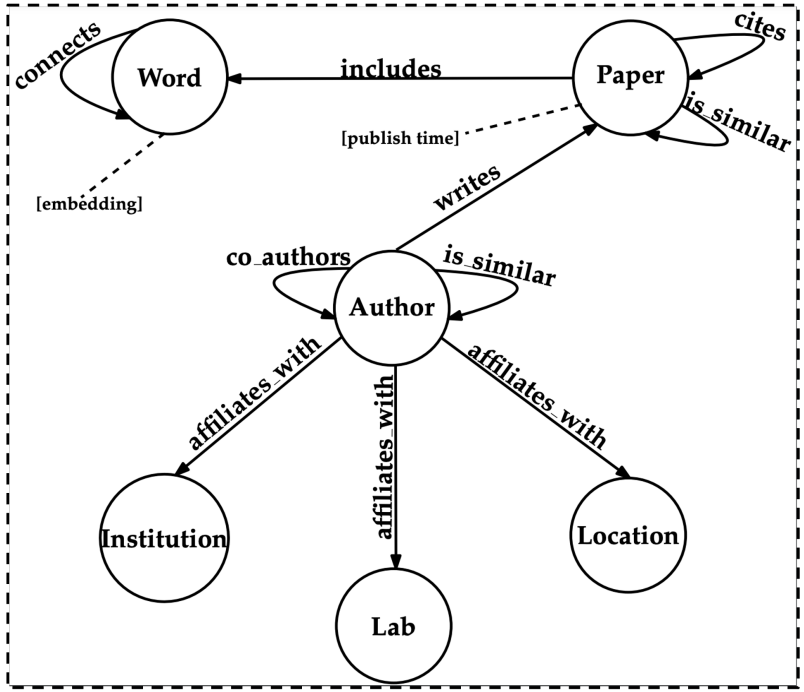
The architecture of the proposed approach



Our Approach

Knowledge graph schema





Our Approach

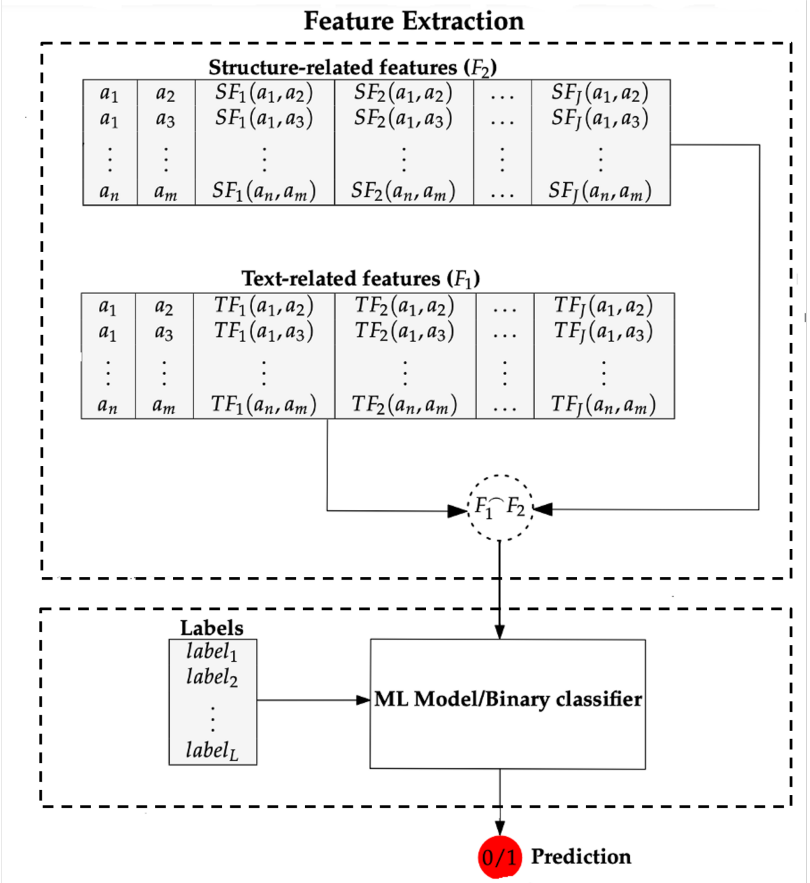
Feature extraction and link prediction

Feature extraction

- Structural characteristics:
  - Adamic Adar
  - Common Neighbors
  - Preferential attachment
  - Total neighbors
- Textual similarity: graph-of-docs similarity

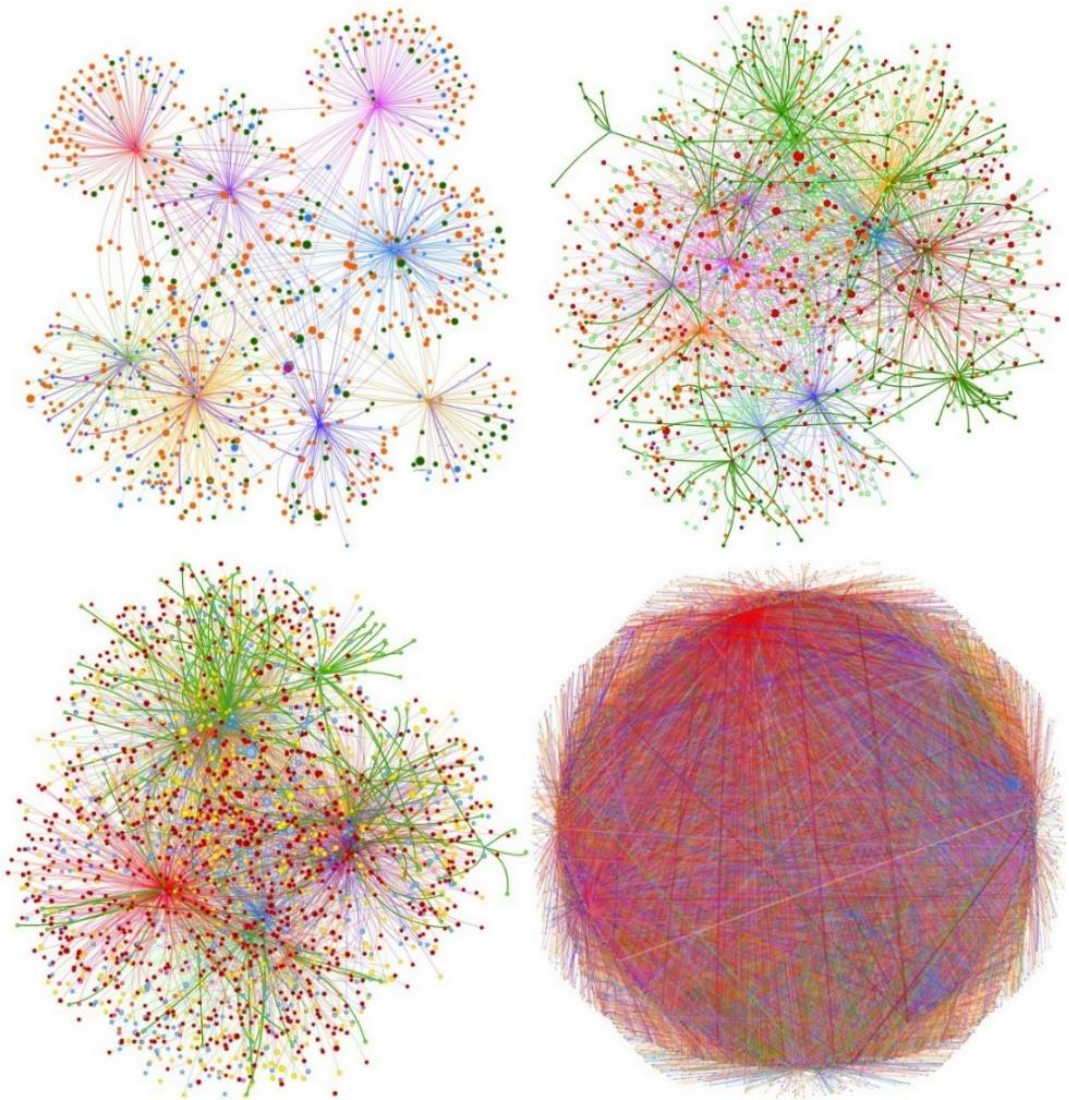
Link prediction

- Binary classification
- Presence or absence of a '*co_authors*' edge



Our Approach

Snapshots of the produced knowledge graph with different numbers of nodes



Experiments

Dataset

- Ten datasets:
  - Subsets of the CORD-19 dataset
  - Different volumes of papers (from 1.536 to 63.023)
  - Balanced datasets

Experiments

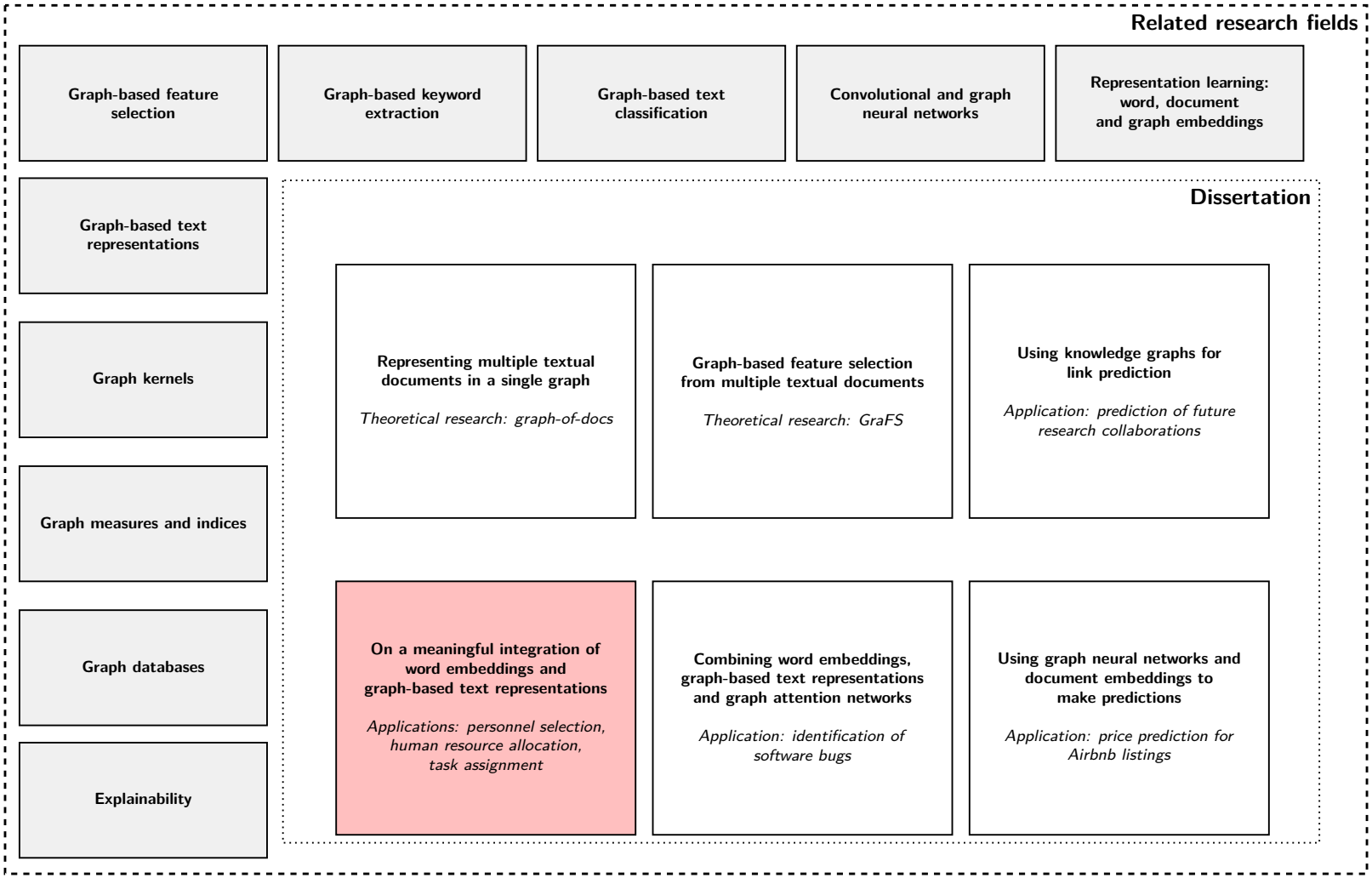
Combinations of features and classification models

Feature combination name	Features included
ALL	Adamic Adar, Common Neighbors, Preferential attachment, Total Neighbors, Pyramid match, Weisfeiler Pyramid match, Jaccard, Propagation
PM	Pyramid Match
WPM	Weisfeiler Pyramid match
AA_J (baseline)	Adamic Adar, Jaccard
AA (baseline)	Adamic Adar
p	Propagation
J (baseline)	Jaccard
AA_WPM	Adamic Adar, Weisfeiler Pyramid match
AA_P	Adamic Adar, Propagation
AA_PM	Adamic Adar, Pyramid match

Evaluation Results

Performance of the neural network classifier per feature combination

Feature Combination	Accuracy	Recall	Precision	Train Los	Test Loss	Abs Loss Difference
ALL	0.9908	**0.9931***	0.9886	0.1020	0.0499	0.0521
J	0.5093	0.0233	**1.0**	0.6647	0.6858	**0.0211**
AA	0.9922	0.9850	0.9995	0.1303	0.0497	0.0806
AA_J	0.9925	0.9856	0.9995	0.1097	0.0413	0.0684
p	0.6954	0.5045	0.8624	0.6289	0.6057	0.0232
PM	0.8452	0.7085	0.9816	0.3219	0.3990	0.0771
WPM	0.9248	0.8590	0.9905	0.2612	0.2390	0.0222
AA_P	0.9923	0.9851	0.9995	0.1311	0.0464	0.0847
AA_PM	**0.9940***	0.9886*	0.9995	0.1281	0.0395	0.0886
AA_WPM	0.9923	0.9870	0.9995	0.1108	0.0372	0.0736


Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Task assignment process

Definition

Deciding which employee will work on each task of an organization.

Problems of existing solutions

- Performed through simple **keyword search** and **ad-hoc** techniques
- **Ad-hoc** techniques:
  - Rely on the managers' subject opinion
  - Tacit knowledge
  - Biased in favor or against certain employees (Sullivan et al. 1988)
- **AI-based** approaches:
  - Structured data
  - Maximize diverse Key Performance Indicators (KPIs)
- **Real-world** problems:
  - Unstructured textual data
  - Jira issues
  - Task descriptions

Task assignment process

Definition

Deciding which employee will work on each task of an organization.

Problems of existing solutions

- Performed through simple **keyword search** and **ad-hoc** techniques
- **Ad-hoc** techniques:
  - Rely on the managers' subject opinion
  - Tacit knowledge
  - Biased in favor or against certain employees (Sullivan et al. 1988)
- **AI-based** approaches:
  - Structured data
  - Maximize diverse Key Performance Indicators (KPIs)
- **Real-world** problems:
  - Unstructured textual data
  - Jira issues
  - Task descriptions

Our Solution

- Unstructured textual data describing past completed tasks
- Natural Language Processing (NLP)
- Linear Assignment Problem

Our Approach (1/2)

Assumption

A company has more **chances to succeed** if its **task assignment** process considers the **skills** of each employee and the nature of the **job** to be accomplished in each task.

Our Approach (1/2)

Assumption

A company has more **chances to succeed** if its **task assignment** process considers the **skills** of each employee and the nature of the **job** to be accomplished in each task.

Our approach

- Aims to increase the **chance of success**
- Properly assigning **employees** to **tasks**
- Reveals hidden knowledge that exists in **unstructured** textual data
- NLP and Operations Research techniques

Our Approach (2/2)

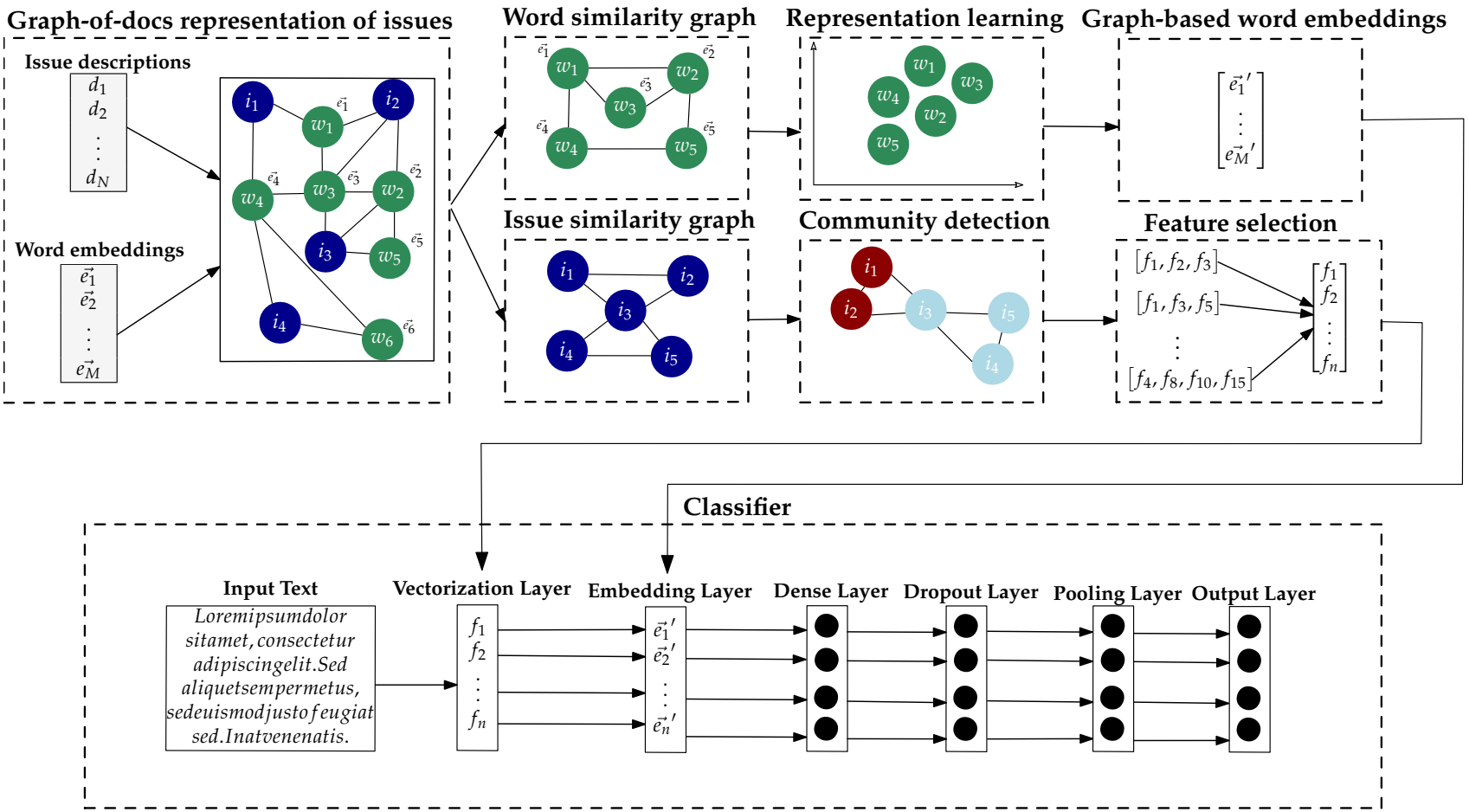
- **Personnel Selection**:
  - Estimating how relevant each employee is to undertake each task
  - How? By analyzing the textual information of past completed tasks
  - For a given task X and an employee Y, we consider a ***relevance metric*** as the probability that the employee Y possesses the skills required by task X

Our Approach (2/2)

- **Personnel Selection**:
  - Estimating how relevant each employee is to undertake each task
  - How? By analyzing the textual information of past completed tasks
  - For a given task X and an employee Y, we consider a ***relevance metric*** as the probability that the employee Y possesses the skills required by task X
- **Human Resource Allocation**:
  - Assigning the employees to tasks in a way that the total relevance is ***maximized***
  - Linear assignment problem

Our Approach

Personnel selection



OR Component

- Linear assignment problem algorithm
- Google OR-Tools
- The total relevance metric is **maximized**
- Bipartite graph

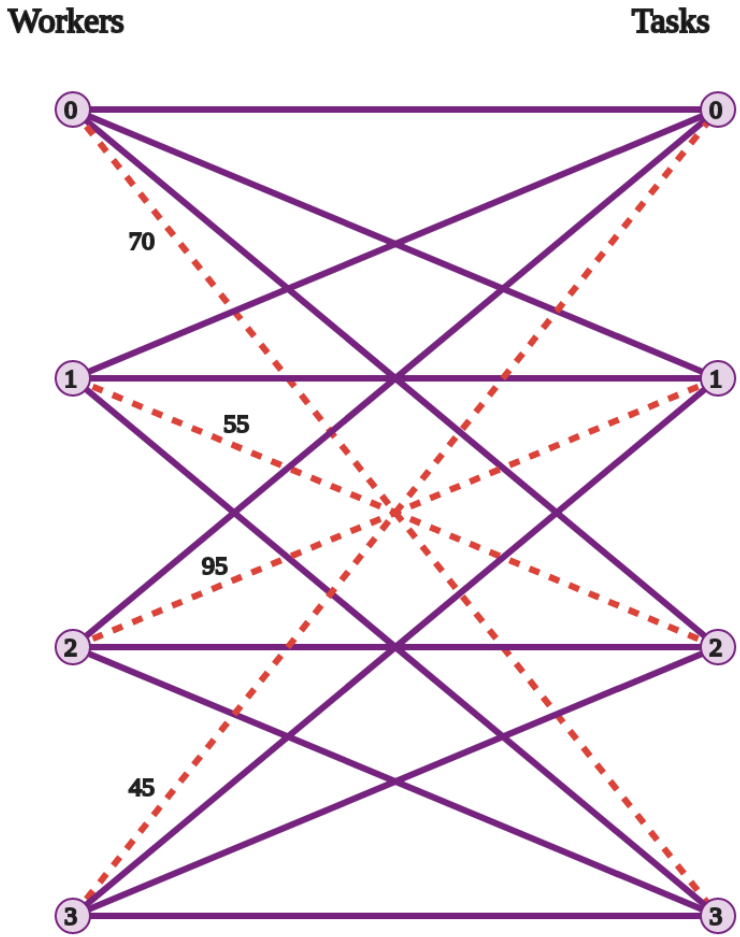


Figure: Source: Google OR-Tools

Experiments

Dataset

- Real-world dataset containing Jira issues of Apache Software Foundation
- 168 projects (including Hadoop, Spark, Airflow)
- Each **label** of the dataset corresponds to an **employee name**

Artifacts

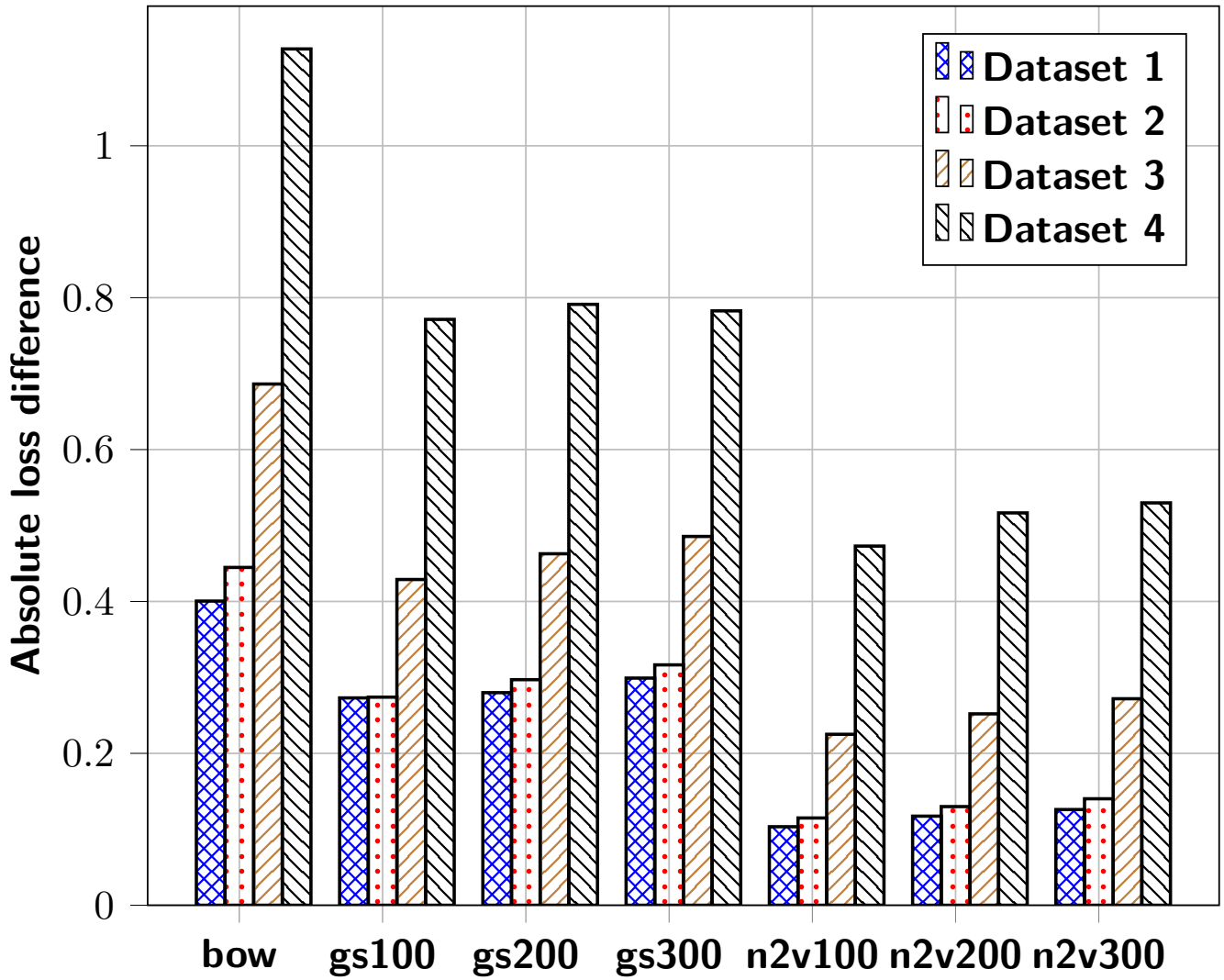
Dataset	Number of unique classes	Number of samples
Dataset 1	3 (most frequent)	37,695
Dataset 2	4 (most frequent)	39,259
Dataset 3	21 (most frequent)	57,798
Dataset 4	300 (all)	228,969

Evaluation results

All the available (300) assignees (Dataset 4)

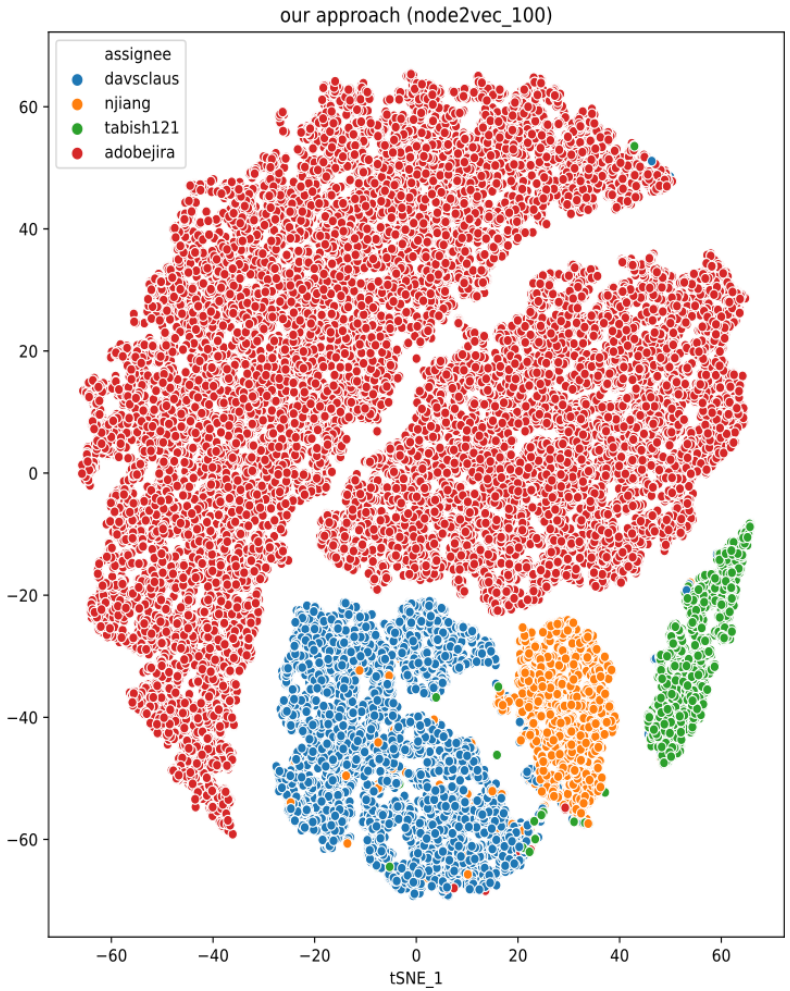
Method	Accuracy	F_1	Train Loss	Validation Loss	Abs Loss Difference
bow_100x50	$50.59(\pm 0.38)$	$50.59(\pm 0.38)$	1.4043	2.5318	1.1275
graphsage_100	$51.00(\pm 0.53)$	$51.00(\pm 0.53)$	1.3063	2.0776	0.7714
graphsage_200	$50.91(\pm 0.30)*$	$50.91(\pm 0.30)$	**1.2925**	2.0836	0.7912
graphsage_300	$50.91(\pm 0.36)*$	$50.91(\pm 0.36)$	1.3114	2.0940	0.7826
node2vec_100	$\mathbf{51.64(\pm 0.27)}*$	$\mathbf{51.64(\pm 0.27)}$	1.5557	**2.0287**	**0.4729**
node2vec_200	$51.32(\pm 0.29)*$	$51.32(\pm 0.29)$	1.5184	2.0350	0.5166
node2vec_300	$51.25(\pm 0.60)*$	$51.25(\pm 0.60)$	1.5131	2.0429	0.5298

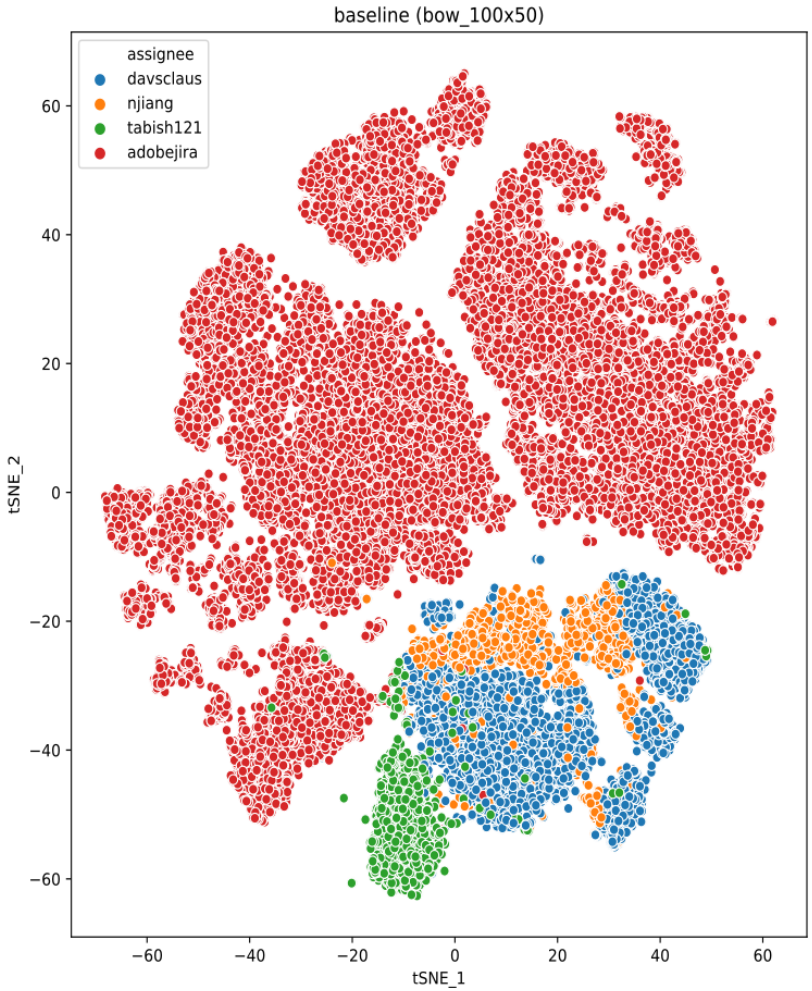
Absolute loss difference of the classification models

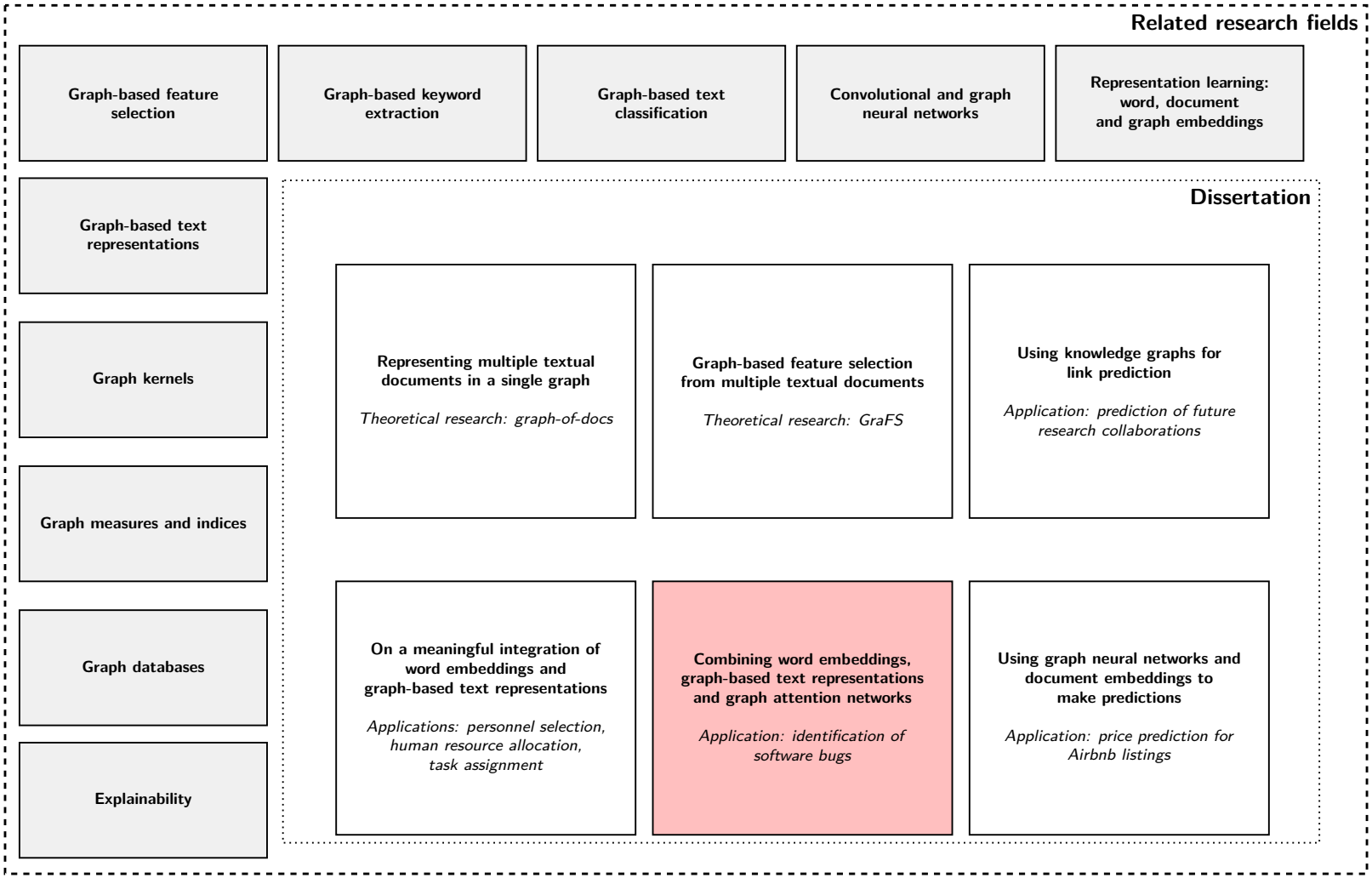


Document embeddings visualization

t-SNE visualizations






Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Identification of software bugs

Definition

Recognize or even predict an error or fault in the design or operation of a computer software.

Existing techniques for the identification of software bugs

- Follow a classical ML model approach
- Utilize word embeddings

Problems of the existing techniques

- Advance each of the above components individually
- Ignore information that is related to the structure of a text or a word of the text

Identification of software bugs

Definition

Recognize or even predict an error or fault in the design or operation of a computer software.

Existing techniques for the identification of software bugs

- Follow a classical ML model approach
- Utilize word embeddings

Problems of the existing techniques

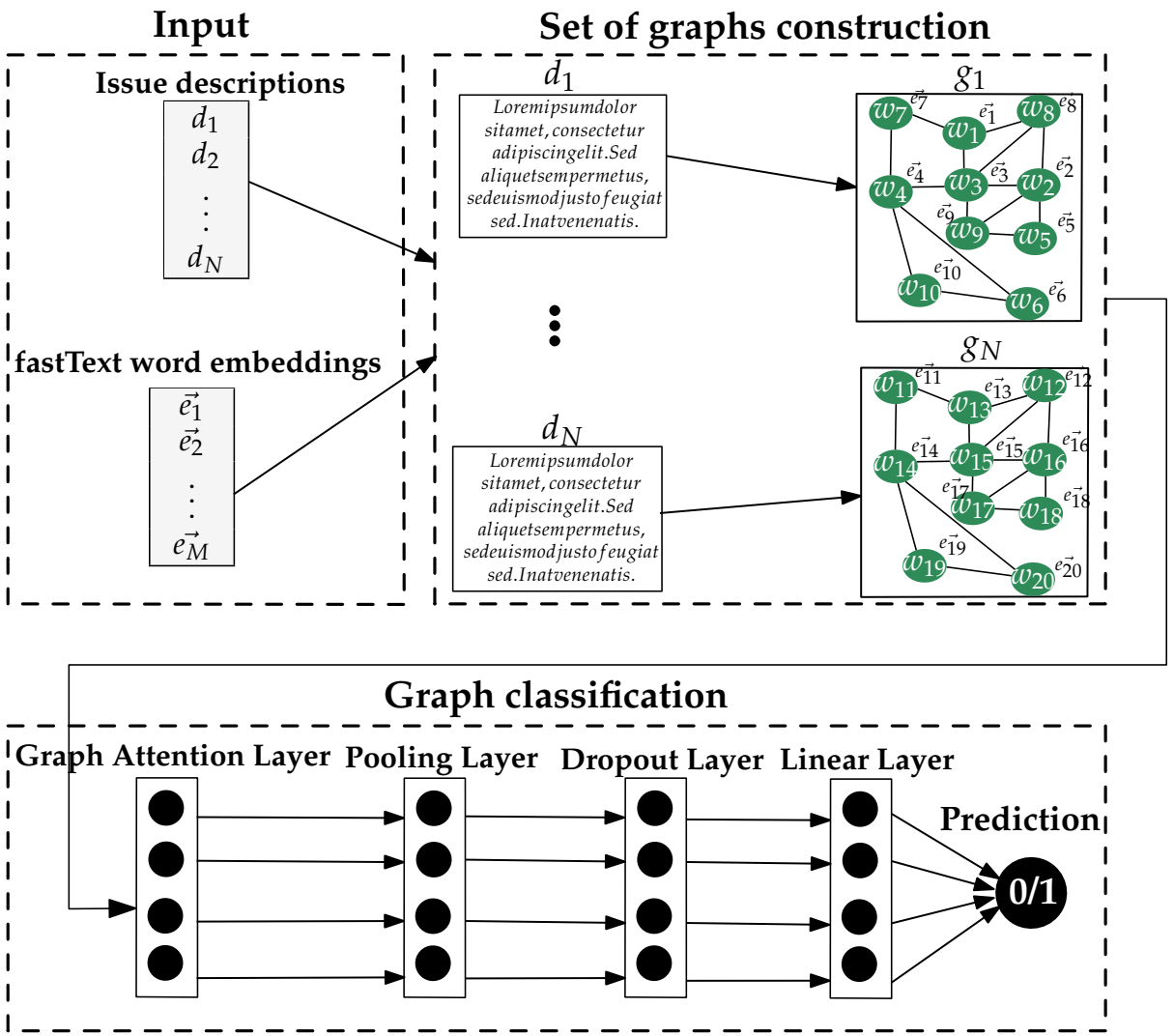
- Advance each of the above components individually
- Ignore information that is related to the structure of a text or a word of the text

Our approach

- Combines word embeddings, graph-based text representations and graph attention networks
- Represents each task as a graph-of-words
- Utilizes both structural and textual characteristics of a document

Our Approach

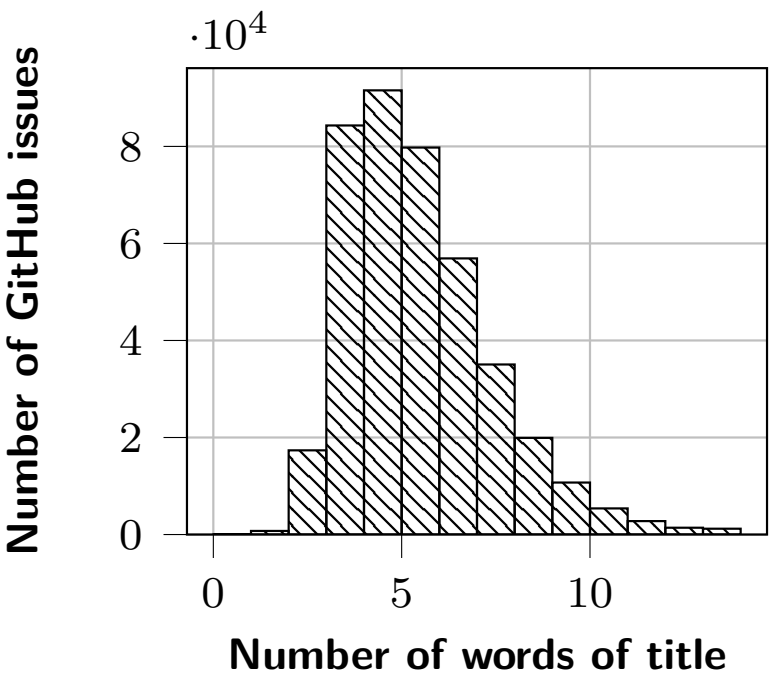
The architecture of the proposed approach

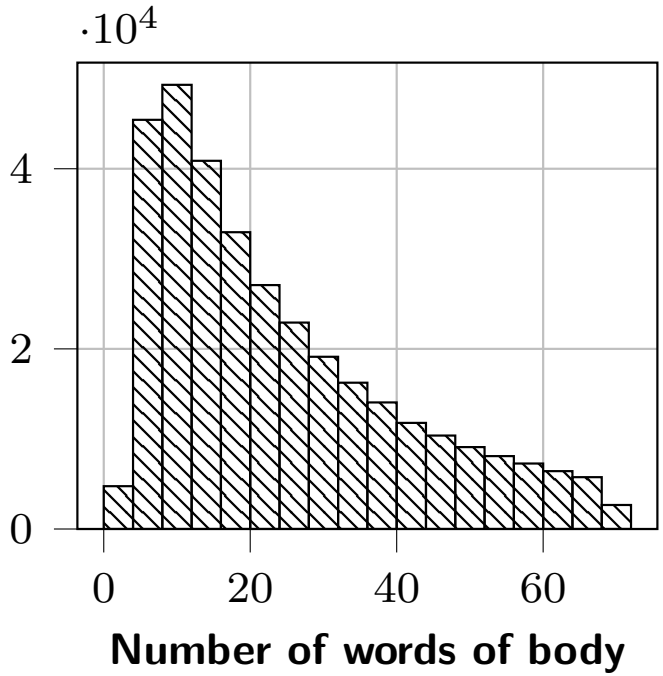


Experiments

Dataset

- A dataset describing GitHub issues
- 450,000 software issues
- 2 classes (bug, feature)





Experiments

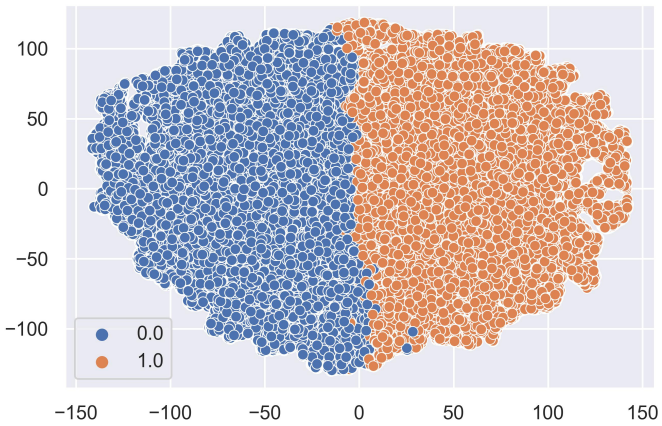
Evaluation Results

Method \ Metric	Accuracy	Precision	Recall
GloVe + LR	0.7227 ± 0.0000	0.7408 ± 0.0000	0.7014 ± 0.0000
GloVe + k-NN	0.6918 ± 0.0000	0.7341 ± 0.0000	0.6199 ± 0.0000
fastText	0.7968 ± 0.0002	0.8107 ± 0.0009	0.7845 ± 0.0017
GloVe + MLP	0.7401 ± 0.0002	0.7544 ± 0.0006	0.7268 ± 0.0013
GloVe + GATConv	0.7651 ± 0.0006	0.7776 ± 0.0022	0.7537 ± 0.0034
GloVe + GCNConv	0.7295 ± 0.0001	0.7475 ± 0.0023	0.7068 ± 0.0043
GloVe + GraphConv	0.7493 ± 0.0006	0.7642 ± 0.0025	0.7331 ± 0.0049
GloVe + SAGEConv	0.7493 ± 0.0006	0.7642 ± 0.0025	0.7331 ± 0.0049
FastGATConv[a]	0.8022 ± 0.0002	0.8125 ± 0.0024	0.7943 ± 0.0036

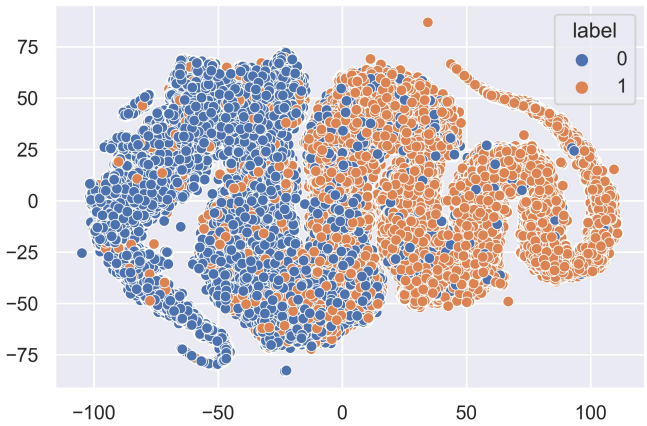
[a] proposed method

Document embeddings visualization

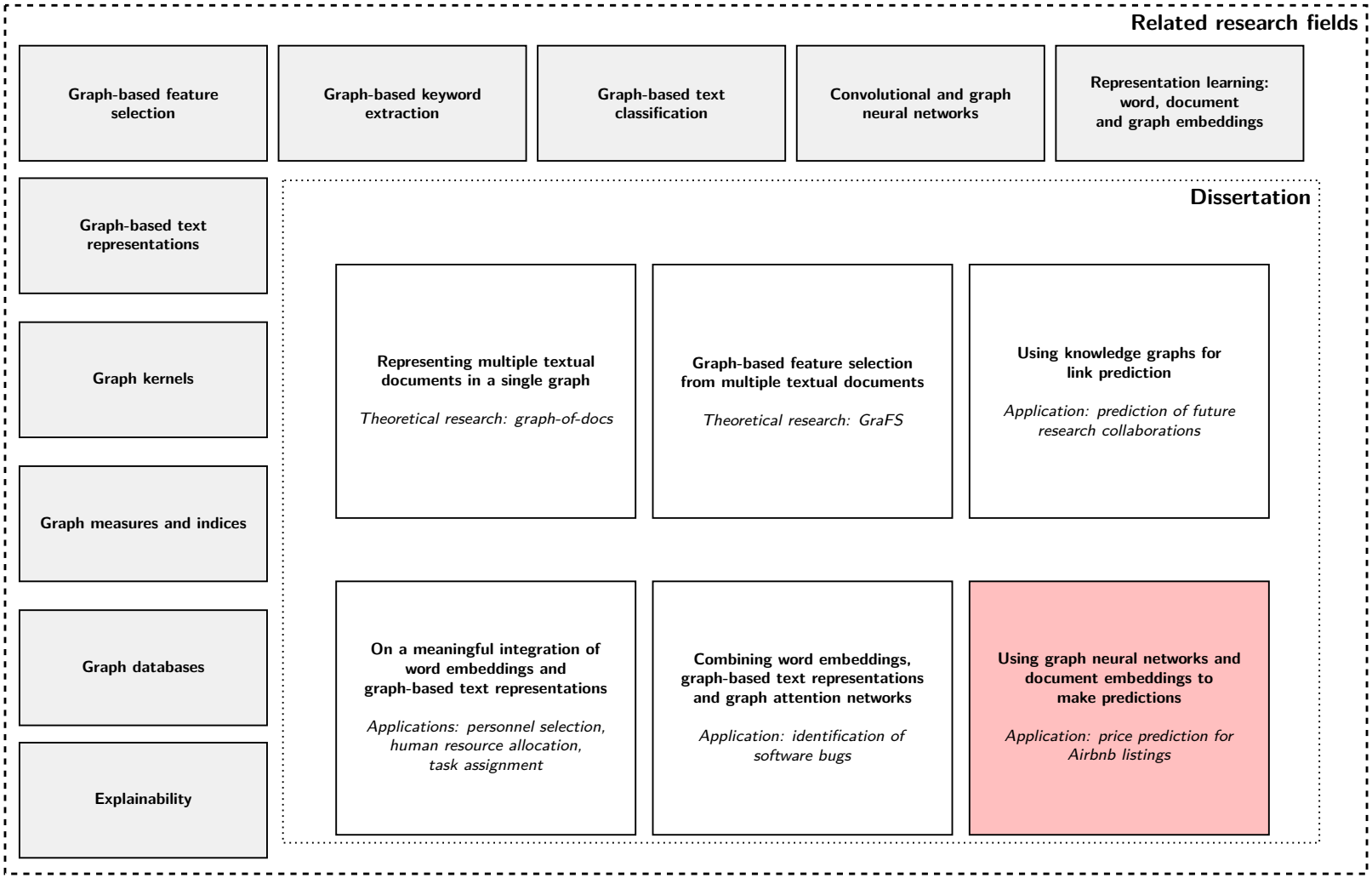
t-SNE visualizations



(a) FastGATConv



(b) fastText


Related research fields
Graph-based feature selection
Graph-based keyword extraction
Graph-based text classification
Convolutional and graph neural networks
Representation learning: word, document and graph embeddings
Graph-based text representations
Graph kernels
Graph measures and indices
Graph databases
Explainability
Dissertation
Representing multiple textual documents in a single graph
Theoretical research: graph-of-docs
Graph-based feature selection from multiple textual documents
Theoretical research: GraFS
Using knowledge graphs for link prediction
Application: prediction of future research collaborations
On a meaningful integration of word embeddings and graph-based text representations
Applications: personnel selection, human resource allocation, task assignment
Combining word embeddings, graph-based text representations and graph attention networks
Application: identification of software bugs
Using graph neural networks and document embeddings to make predictions
Application: price prediction for Airbnb listings

Predicting prices of Airbnb listings

Task

Predict the price for a given Airbnb listing for a specific day, taking also into consideration the prices of the Airbnb listings nearby.

Problems of the existing techniques for price prediction of Airbnb listings

- Rely only on the features of each individual listing
- Ignoring any topological or neighborhood properties

Predicting prices of Airbnb listings

Task

Predict the price for a given Airbnb listing for a specific day, taking also into consideration the prices of the Airbnb listings nearby.

Problems of the existing techniques for price prediction of Airbnb listings

- Rely only on the features of each individual listing
- Ignoring any topological or neighborhood properties

Our approach

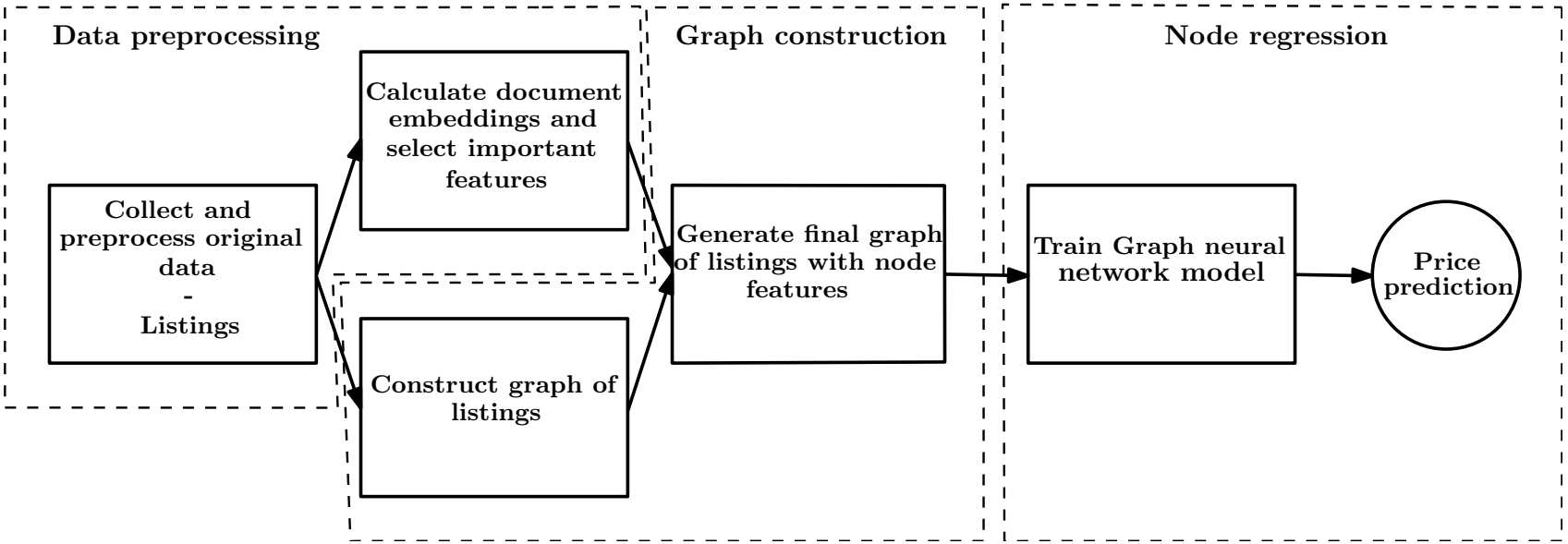
- Represents the listings of a given area as a graph
- Exploits the features of each individual listing
- Takes into consideration information related to the neighborhood of a listing
- Based on graph neural networks
- Document embeddings

Our Approach

- Combines techniques from:
  - graph neural networks
  - word and document embeddings
- Recommends an appealing and profitable price for a listing
- Regression problem

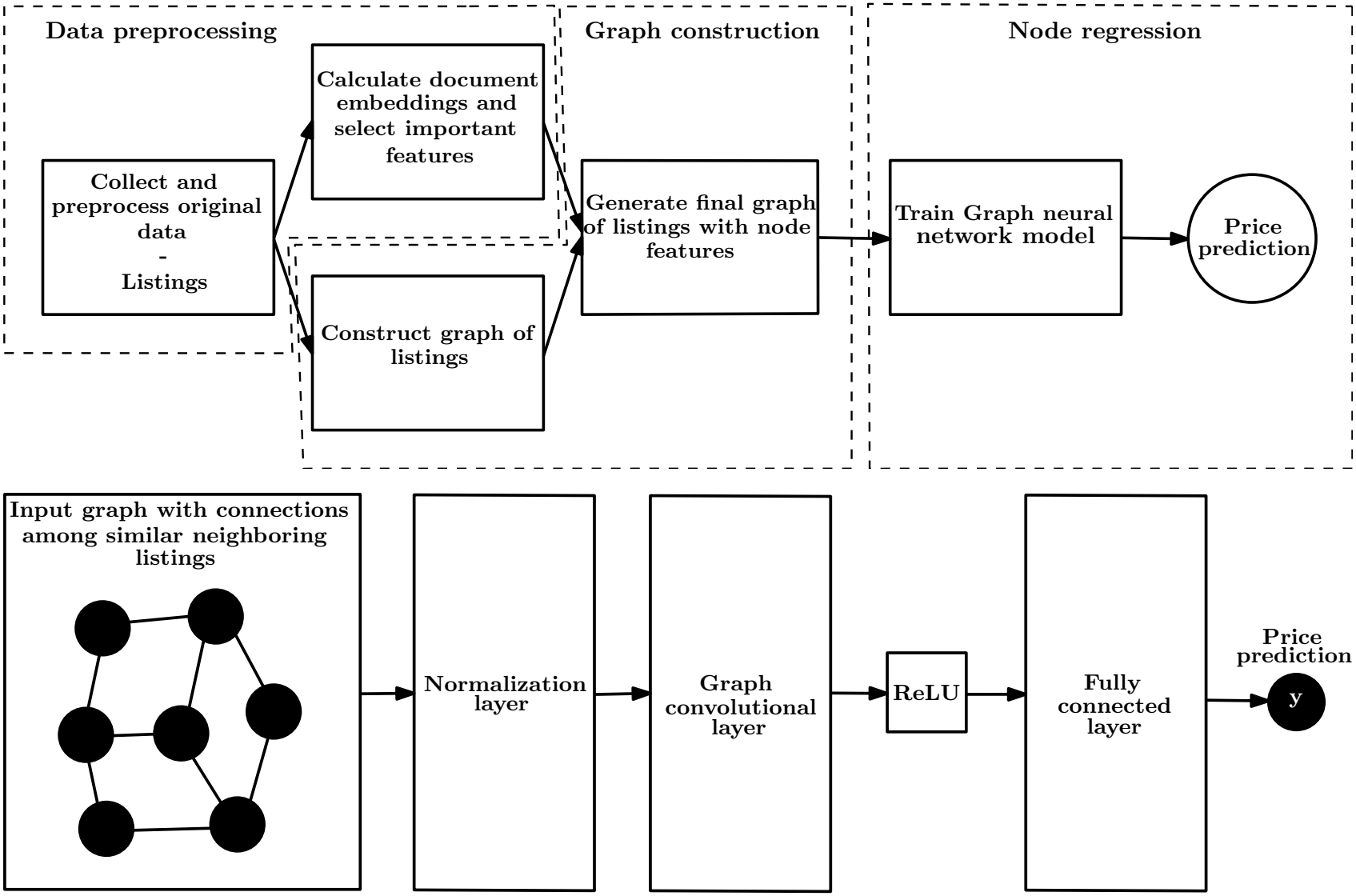
Our Approach

The architecture of the proposed approach



Our Approach

The architecture of the proposed approach



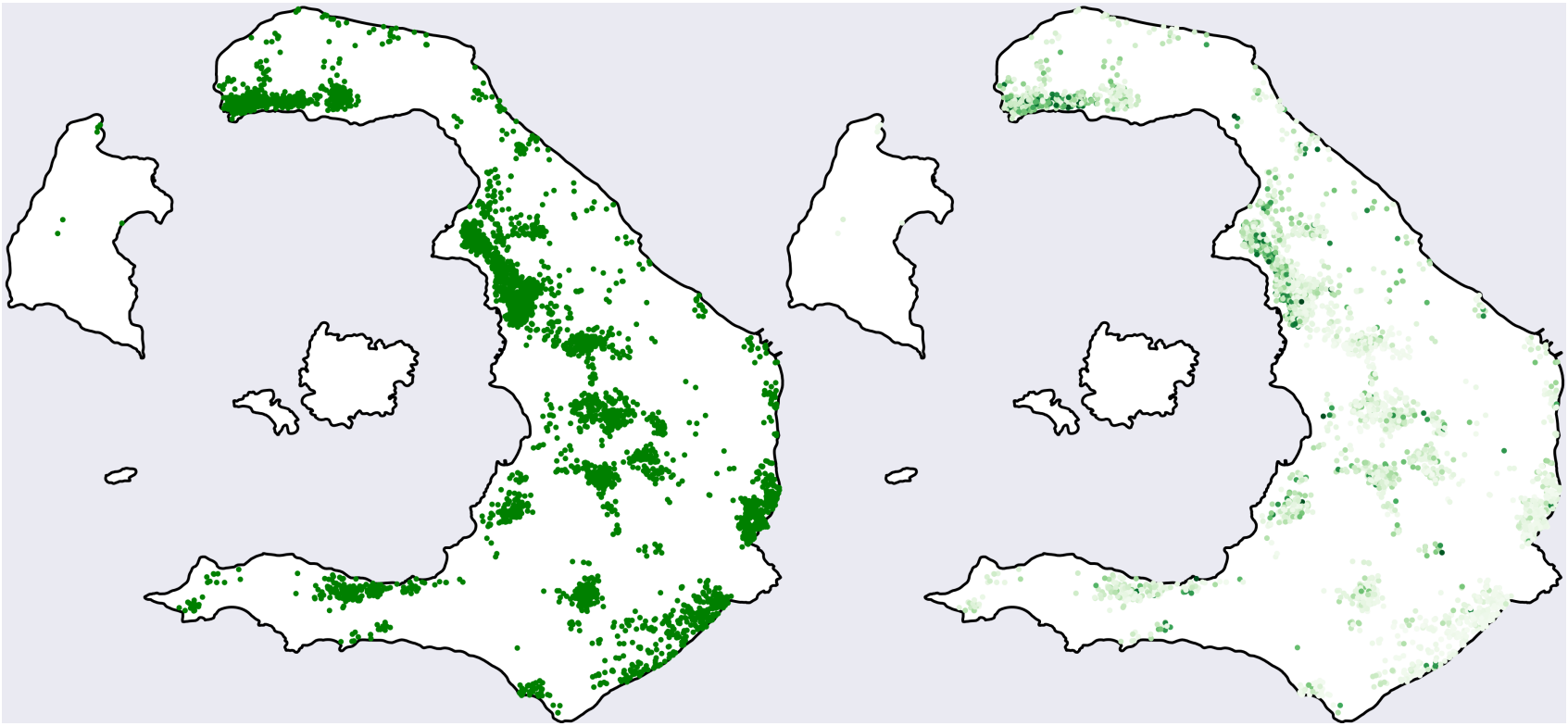
Experiments

Dataset

- A dataset describing Airbnb listings
- 4,540 Airbnb listings
- Each sample has:
  - Number of accommodates
  - latitude
  - longitude
  - number of reviews
  - number of beds
  - description
  - price
- Node regression

Feature \ Dataset	G@0	G@0.75	G@0.9	G@0.95
Number of nodes	4540	4540	4540	4540
Number of edges	159283	16370	7914	5426
Average node degree	70.2	7.211	3.4863	2.39030
Number of attributes of a node	114	114	114	114
Distance threshold	200	200	200	200
Document similarity threshold	0	0.75	0.9	0.95

Distribution of the Airbnb listings across the island



Evaluation results

Method	R^2@0	MSE@0	R^2@0.75	MSE@0.75	R^2@0.9	MSE@0.9	R^2@0.95	MSE@0.95
Our approach (GCNBnB)	**0.208**	0.704	**0.281**	0.640	**0.359**	0.570	**0.4630**	0.478
Neural network (NN)	-0.171	1.043	-0.171	1.043	-0.171	1.043	-0.171	1.043
Linear regression (LR)	-0.034	0.921	-0.034	0.921	-0.034	0.921	-0.034	0.921
Random forest (RF)	0.001	0.889	0.001	0.889	0.001	0.889	0.001	0.889
Decision tree (DT)	-0.039	0.925	-0.039	0.925	-0.039	0.925	-0.039	0.925

Evaluation results

Method	R^2@0	MSE@0	R^2@0.75	MSE@0.75	R^2@0.9	MSE@0.9	R^2@0.95	MSE@0.95
Our approach (GCNBnB)	**0.208**	0.704	**0.281**	0.640	**0.359**	0.570	**0.4630**	0.478
Neural network (NN)	-0.171	1.043	-0.171	1.043	-0.171	1.043	-0.171	1.043
Linear regression (LR)	-0.034	0.921	-0.034	0.921	-0.034	0.921	-0.034	0.921
Random forest (RF)	0.001	0.889	0.001	0.889	0.001	0.889	0.001	0.889
Decision tree (DT)	-0.039	0.925	-0.039	0.925	-0.039	0.925	-0.039	0.925

- Graph-based models perform **better** than traditional approaches
- **Location** along with the **nearby listings** affect the price of a listing
- Graph-based representations **enable** the utilization of topological characteristics of an area
- The addition of the **postal code** to the feature space of a listing is not **sufficient**

Conclusions

Contribution

Contribution

- Introduction of graph-of-docs

Contribution

- Introduction of graph-of-docs
- Introduction of a graph-based feature selection method

Contribution

- Introduction of graph-of-docs
- Introduction of a graph-based feature selection method
- Empirical evaluation of the combination:
  - embeddings
  - graph-based text representations
  - graph neural networks

Lessons learned

Lessons learned

Feature engineering

- graph-based feature selection methods $\Rightarrow$ *accuracy* $\uparrow$
- textual + structural characteristics $\Rightarrow$ *accuracy* $\uparrow$
- structural information + GNN models $\Rightarrow$ *accuracy* $\uparrow$

Lessons learned

Feature engineering

- graph-based feature selection methods $\Rightarrow$ *accuracy* $\uparrow$
- textual + structural characteristics $\Rightarrow$ *accuracy* $\uparrow$
- structural information + GNN models $\Rightarrow$ *accuracy* $\uparrow$

Representation learning

- fast execution + efficiency $\Rightarrow$ node2vec algorithm
- **dimension**$\uparrow$ of a classical word embedding $\not\Rightarrow$ **performance**$\uparrow$
- The **combination** of word embeddings, neural networks and graph representation learning is generally effective
- GNNs with **local-structure propagation** rules $\Rightarrow$ good for **short** textual data

Lessons learned

Feature engineering

- graph-based feature selection methods $\Rightarrow$ *accuracy* $\uparrow$
- textual + structural characteristics $\Rightarrow$ *accuracy* $\uparrow$
- structural information + GNN models $\Rightarrow$ *accuracy* $\uparrow$

Representation learning

- fast execution + efficiency $\Rightarrow$ node2vec algorithm
- **dimension**$\uparrow$ of a classical word embedding $\not\Rightarrow$ **performance**$\uparrow$
- The **combination** of word embeddings, neural networks and graph representation learning is generally effective
- GNNs with **local-structure propagation** rules $\Rightarrow$ good for **short** textual data

Model selection and hyper-parameter tuning

- **Poor** word embeddings **inhibit** the performance of the GNN models
- Mindful **selection** and **combination** of word embeddings, textual representations and classification models is important
- Proper configuration and selection of a **small** sliding window size
- **Large** sliding window sizes add **redundant** and **noisy** edges between the nodes

Future work directions

- More **centrality** measures
- Diverse **community detection** and graph partitioning algorithms

Future work directions

- More **centrality** measures
- Diverse **community detection** and graph partitioning algorithms
- 'Simplifying Graph Convolutional Networks'
- Combining **classical** and **modern** graph mining methods

Future work directions

- More **centrality** measures
- Diverse **community detection** and graph partitioning algorithms
- 'Simplifying Graph Convolutional Networks'
- Combining **classical** and **modern** graph mining methods
- Iterative interplay between **ML** and **operations research**

Future work directions

- More **centrality** measures
- Diverse **community detection** and graph partitioning algorithms
- 'Simplifying Graph Convolutional Networks'
- Combining **classical** and **modern** graph mining methods
- Iterative interplay between **ML** and **operations research**
- In-memory graph database in combination with Neo4j.

List of publications (1/2)

17 altogether, 111 citations, 8 h-index (as of July 14, 2023)

Journals:

1. Kanakaris, N., Giarelis, N., Siachos, I., & Karacapilidis, N. (2021a). Shall i work with them? a knowledge graph-based approach for predicting future research collaborations. *Entropy*, *23*(6), 664
2. Kanterakis, A., Kanakaris, N., Koutoulakis, M., Pitianou, K., Karacapilidis, N., Koumakis, L., & Potamias, G. (2021). Converting biomedical text annotated resources into fair research objects with an open science platform. *Applied Sciences*, *11*(20). https://doi.org/10.3390/app11209648
3. Kanakaris, N., Giarelis, N., Siachos, I., & Karacapilidis, N. I. (2021b). Making personnel selection smarter through word embeddings: A graph-based approach. *Machine Learning with Applications*
4. Michail, D., Kanakaris, N., & Varlamis, I. (2022). Detection of fake news campaigns using graph convolutional networks. *International Journal of Information Management Data Insights*, *2*(2), 100104. https://doi.org/https://doi.org/10.1016/j.jjimei.2022.100104

Book chapters:

1. Kanakaris, N., Karacapilidis, N., Kournetas, G., & Lazanas, A. (2020b). Combining machine learning and operations research methods to advance the project management practice. In G. H. Parlier, F. Liberatore, & M. Demange (Eds.), *Operations research and enterprise systems* (pp. 135–155). Springer International Publishing
2. Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2022). Medical knowledge graphs in the discovery of future research collaborations. In C.-P. Lim, Y.-W. Chen, A. Vaidya, C. Mahorkar, & L. C. Jain (Eds.), *Handbook of artificial intelligence in healthcare: Vol 2: Practicalities and prospects* (pp. 371–391). Springer International Publishing. https://doi.org/10.1007/978-3-030-83620-7_16
3. *To appear*: Kanakaris, N., Michail, D., Varlamis, I. A Comparative Survey of Graph Databases and Software for Social Network Analytics: The Link Prediction Perspective. *Book name:* Graph Databases and their use in social media and smart cities

List of publications (2/2)

17 altogether, 111 citations, 8 h-index (as of July 14, 2023)

Conferences:

1. Kanakaris., N., Karacapilidis., N., & Lazanas., A. On the advancement of project management through a flexible integration of machine learning and operations research tools. In: In *Proceedings of the 8th international conference on operations research and enterprise systems - icores*. INSTICC. SciTePress, 2019, 362–369. ISBN: 978-989-758-352-0. https://doi.org/10.5220/0007387103620369
2. Kanterakis, A., Iatraki, G., Pityanou, K., Koumakis, L., Kanakaris, N., Karacapilidis, N., & Potamias, G. (2019). Towards reproducible bioinformatics: The openbio-c scientific workflow environment. *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, 221–226. https://doi.org/10.1109/BIBE.2019.00047
3. Kanakaris, N., Karacapilidis, N., & Kournetas, G. On the exploitation of textual descriptions for a better-informed task assignment process. In: In *Proceedings of the 9th international conference on operations research and enterprise systems - icores*, INSTICC. SciTePress, 2020, 304–310. ISBN: 978-989-758-396-4. https://doi.org/10.5220/0009151603040310
4. Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2020b). On a novel representation of multiple textual documents in a single graph. In I. Czarnowski, R. J. Howlett, & L. C. Jain (Eds.), *Intelligent decision technologies* (pp. 105–115). Springer Singapore
5. Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2020a). An innovative graph-based approach to advance feature selection from multiple textual documents. In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), *Artificial intelligence applications and innovations* (pp. 96–106). Springer International Publishing
6. Giarelis, N., Kanakaris, N., & Karacapilidis, N. I. (2020c). On the utilization of structural and textual information of a scientific knowledge graph to discover future research collaborations: A link prediction perspective. *IFIP Working Conference on Database Semantics*
7. Giarelis, N., Kanakaris, N., & Karacapilidis, N. (2021). A comparative assessment of state-of-the-art methods for multilingual unsupervised keyphrase extraction. In I. Maglogiannis, J. Macintyre, & L. Iliadis (Eds.), *Artificial intelligence applications and innovations* (pp. 635–645). Springer International Publishing
8. Kanakaris, N., Siachos, I., & Karacapilidis, N. (2022). Is it a bug or a feature? identifying software bugs using graph attention networks. *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, 1425–1429. https://doi.org/10.1109/ICTAI56018.2022.00215
9. Kanakaris, N., & Karacapilidis, N. (2023). Predicting prices of airbnb listings via graph neural networks and document embeddings: The case of the island of santorini [CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022]. *Procedia Computer Science*, *219*, 705–712. https://doi.org/https://doi.org/10.1016/j.procs.2023.01.342
10. Adamides, E., Giarelis, N., Kanakaris, N., Karacapilidis, N., Konstantinopoulos, K., & Siachos, I. (2023). Leveraging open innovation practices through a novel ict platform. In A. Zimmermann, R. Howlett, & L. C. Jain (Eds.), *Human centred intelligent systems* (pp. 3–12). Springer Nature Singapore

Participation in research projects

inPOINT

- Development of an inclusive platform for supporting and augmenting Open Innovation activities
- National Research Program, "Competitiveness-Entrepreneurship-Innovation", T2EΔK-04389

ECLiPSe

- Energy Saving through Smart Devices Control in Large Passenger and Cruise Ships
- ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ 2014-2020 RIS3 «ΕΝΕΡΓΕΙΑΚΕΣ ΕΦΑΡΜΟΓΕΣ», ΚΩΔΙΚΟΣ ΠΡΑΞΗΣ: ΔΕΡ7-0020392

myCorridor

- Mobility as a Service in a multimodal European cross-border corridor
- EU Project, H2020-MG-2016-2017

OpenBio-C

- An Open and Integrated Collaborative Bioinformatics Platform
- ΕΡΕΥΝΩ-ΔΗΜΙΟΥΡΓΩ-ΚΑΙΝΟΤΟΜΩ, T1EΔK-05275

Who spread the rumors

- Organized action detection tool (misinformation through modeling the spread of false news on social media)
- RESEARCH-CREATE-INNOVATE, T2EDK 3778

Thank you