Department of Artificial Intelligence
Faculty of Social Sciences
Radboud University Nijmegen

# Graph-based semi-supervised learning of semantic text clusters

MASTER THESIS

Natalie Widmann

s4499972

May 4th, 2017

**Supervisors**:

Dr. Suzan Verberne,
Leiden Centre of Data Science
Leiden University

Dr. Jason Farquhar
Donders Institute
Radboud University Nijmegen

## Abstract

The bag-of-words model is a common approach to represent documents for all kind of text mining tasks. However, the assumed independence of words does not reflect the complexity and context of human natural language. We propose a graph-based representation of collections of documents that include documents and features with their respective syntactic, semantic and frequency-based relations.
Based on semi-supervised learning - an approach that besides using labeled data, also incorporates the structure of unlabeled data for classifier training - the influence of different graph properties on text categorization is investigated. The results show that even though bag-of-words is a powerful approach, adding word relations significantly improves classification performance. Whether syntactic or semantic feature relations are used has, however, no significant influence.
Although, graph-based semi-supervised learning outperforms bag-of-words based supervised and semi-supervised learning approaches when varying the number of labeled documents, it is not able to use the full potential of including unlabeled data.
The big advantage of graph-based methods is their flexibility to perfectly adapt the document representation to a specific text mining task.

# Contents

## 1 Introduction

The majority of the online accessible data is unstructured or semi-structured text data. In order to efficiently handle these enormous amounts, automated information extraction, document categorization and summarization are necessary to give a user a quick overview about a document's content and its relevancy for a specific purpose. However, this is not straightforward as text data is based on natural language. This form of human interaction emerged a long time ago and is still constantly evolving. Communication based on written and spoken language requires knowledge about grammatical structures, semantic concepts, ambiguous word, a sense of humor, irony and sarcasm, as well as cultural understanding and knowledge about the world.
Despite this complex interaction, most research in the field of text mining focuses on one particular aspect of written language: the word frequency. By representing documents with the so-called bag-of-words model words are assumed to be independent of each other. Therefore, no information about syntactic or semantic relations between words or relevant word sequences is captured. Especially, when classifying short documents the semantic relation between words is relevant to identify texts that cover the same topic but use synonyms or closely related words.

In this project we investigate whether a graph-based text representation, that captures syntactic relations, semantic similarities as well as word frequencies, is able to improve performance in a benchmark text categorization task. We use semi-supervised learning as it incorporates the structure of unlabeled data points which is of substantial importance when the amount of unlabeled data greatly exceeds the amount of labeled data.
Based on the widely used 20-Newsgroup dataset[1] we investigate two different aspects:

(1) We propose a graph-based text representation that includes documents and features of documents linked with syntactic, semantic and frequency based relations. With semi-supervised learning we assess the influence and relevancy of different graph properties on categorization performance. Furthermore, the graph-based semi-supervised learning approach is compared to semi-supervised learning based on a standard bag-of-words model.

(2) In the second experiment we focus on investigating the effectiveness of semi-supervised learning by varying the amount of labeled documents. Besides comparing graph-based and bag-of-words based semi-supervised learning, a standard supervised learning approach is included.

Text features related to term frequency are commonly used and have proven to perform decently in text classification tasks. However, a main disadvantage is that two documents are only assigned to the same class if they use the same keywords. This so-called vocabulary gap can be bridged by adding information about the semantic similarity of words which connects synonyms with high weights. Therefore, we expect that a model containing word frequency as well as information about the semantic similarity of words outperforms other feature combinations.
Furthermore, we expect that the graph-based semi-supervised learning approach leads to a better classification performance with less labeled data points than supervised learning as structural information about unlabeled data is included during training. Moreover,

[1]Information and access to the 20-Newsgroup dataset can be obtained here.

compared to bag-of-words based semi-supervised learning the graph-based semi-supervised learning approach has the advantage of including relational information.

The thesis is organized as follows: In Chapter 2 we lay the foundation of text representation and semi-supervised learning by reviewing literature in these fields. A special focus is on the label propagation algorithm which will be used for semi-supervised learning. Chapter 3 gives detailed insights about the dataset and the applied methods used for preprocessing. Graph construction is illustrated by an example and the obtained graph-features and modulations are discussed. In Chapter 4 we describe the structure of the two experiments, the graph-based feature analysis and the number of labeled documents, and outline the measures used for evaluation. Subsequently, Chapter 5 states the results and further insights that are gained by the graph-based semi-supervised learning approach. Besides discussing results and possible improvements, Chapter 6 gives a short overview about the computational complexity, as with graph-based semi-supervised learning methods scalability is a key issue. In Chapter 7 the final conclusion is presented.

# 2 Background

## 2.1 Text Representation

Representing a document in a computer processable way is necessary for any text mining task such as document classification, sentiment analysis, topic modeling, text summarization, etc. Besides preparing the document for digital processing, a good text representation is able to capture relevant information for a specific task while minimizing the dimensionality of the problem such that machine learning algorithms can be applied efficiently (Sonawane and Kulkarni, 2014).
The most popular approach in the field of text mining is vector representation. Documents are placed in a space spanned by the vocabulary of the entire collection (Turney and Pantel, 2010). These, so called, vector space models allow to assess the semantic similarity of documents by computing the distance of the corresponding vectors (Turney and Pantel, 2010). Bag-of-words is a commonly used vector space model that captures the word frequency in documents (Jiang et al., 2010). However, the words spanning the vector space are independent from each other such that it is not possible to include information about sentence structure, semantic or conceptual similarity of words or the structure of the entire document (Sonawane and Kulkarni, 2014). Therefore, bag-of-words models are limited to applications in which frequent words indicate the meaning of a text (Turney and Pantel, 2010) and in which documents that cover the same topic use similar words (Sonawane and Kulkarni, 2014).
Often the most common *n-grams*, short sequences of $n$ words, are added to the simple bag for words model to include context information and small semantic units. Nevertheless, bag-of-words models fail to capture semantic concepts that are expressed with synonyms or are arranged in a slightly different word order (Rousseau et al., 2015).

In this project we will focus on graph-based text representations as they offer possibilities to overcome the limitations of bag-of-words models. Graphs are mathematical constructs consisting of nodes and edges that can efficiently model structural and relational information (Sonawane and Kulkarni, 2014). Dependent on the dataset and the text mining problem,

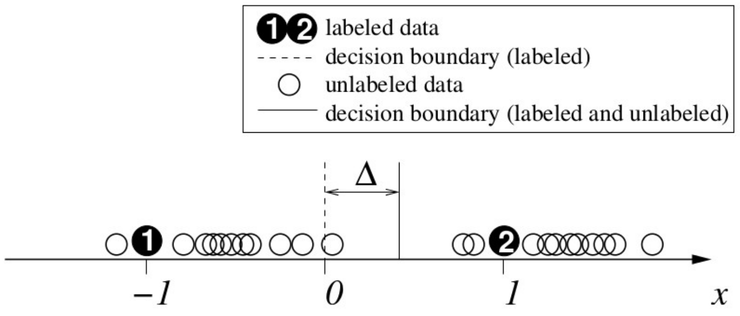
different graph representations of a document collection or a specific text are possible. Often nodes represent document entities and edges the relations between them, such as information about the author, journal, covered topics, etc (Schenker, 2003). Another possibility is to use a graph for each text document by displaying unique words as nodes and use connecting edges to indicate consecutive sequences, semantic similarities, common words in a sentences and paragraphs or word co-occurrences (Sonawane and Kulkarni, 2014).
As many machine learning algorithms rely on similarity or distance measures, as well as the computation of centroids and other numerical values (Schenker, 2003), graph-based text representations require further processing or specially adapted graph-based machine learning methods (Jiang et al., 2010). The required operations are often computationally expensive as graph-based text representations add an additional level of complexity by including different types of structural and semantic information (Jiang et al., 2010).

Despite these limitations, recent research shows a growing interest in graph-based text representations and its efficient combination with machine learning algorithms. After comparing different graph-based text representations to traditional bag-of-words models Sonawane and Kulkarni (2014) conclude that "graph models are the most suitable representations of text documents". Ganesan et al. (2010) use graph structures to summarize the content of short and highly redundant comments such as product reviews. Other applications are graph-based sentiment analysis (Castillo et al., 2015) or semantic clustering (Kannan et al., 2016).

## 2.2 Semi-Supervised Learning

Semi-supervised learning is a fast developing field in machine learning that combines supervised learning, which is based on labeled data, with unsupervised learning, in which underlying structures are discovered based on unlabeled data (Zhu, 2005). The idea is that the distribution of unlabeled data adds relevant information to a supervised learning algorithm and hence, positively influences the ability to assign labels to data points (Chapelle et al., 2009). Figure 1 illustrates how the decision boundary in a binary classification task changes when the structure of unlabeled data points is considered during the learning process (Zhu, 2007). The objective is to assign a class to the data given the points 1 and 2. Without further information, the decision boundary is set half way between 1 and 2 (dashed line) such that the distance to the given data points determines the label.
However, this decision boundary incorrectly assigns class 2 to the point at $x = 0$. In contrast, by including information from unlabeled data points the underlying class structures become clear. The data points are sampled from two Gaussian distributions and hence, by estimating the corresponding means a better decision boundary is computed. The label of point $x = 0$ is corrected as its probability to be drawn from the Gaussian distribution of class 2 is smaller than its probability to be drawn from the distribution of class 1.

Similar, to supervised or unsupervised learning, the field of semi-supervised learning consists of a wide variety of research branches (Chapelle et al., 2009) which include different approaches and algorithms. We focus on graph-based semi-supervised learning because graphs are able to capture multiple complex text properties of a collection of documents (see Section 2.1).



❶❷ labeled data
decision boundary (labeled)
unlabeled data
decision boundary (labeled and unlabeled)
Δ
−1
0
1
x

Figure 1: The illustration, taken from (Zhu, 2007), compares supervised and semi-supervised learning in a binary classification task. While in supervised learning only the labeled data is used to predict the remaining classes (dashed decision boundary), in semi-supervised learning the unlabeled data points provide information about the underlying class distribution such that the classification performance is improved (solid decision boundary).

Another division in semi-supervised learning concerns the type of data for which class labels are assigned. While *transductive* semi-supervised learning is limited to the unlabeled data used for training, *inductive* learning aims to classify also unseen data (Chapelle et al., 2009). In this project we only consider transductive learning.

By incorporating the structure of unlabeled data the classification performance of standard supervised learning can be improved (Bengio et al., 2006) and hence, the required number of labeled data to reach a similar performance can be reduced. Therefore, semi-supervised learning is mainly applied when a lot of unlabeled data exists, but labeling is time-consuming, costly or requires a lot of effort or expert knowledge (Zhou et al., 2003). This is the case for all kind of text and web categorization tasks, as well as protein or genome sequencing, speech recognition or text parsing. Balcan et al. (2005) show its advantage compared to supervised learning approaches in the field of person identification from webcam images. Similarly, when used for the prediction of gene regulatory networks semi-supervised learning outperforms support vector machines and random forests (Patel and Wang, 2015). Also, for subsets of the here used 20-Newsgroup dataset semi-supervised learning has been applied. Zhou et al. (2003) shows that smoothing the classification function with respect to the intrinsic structure revealed by labeled and unlabeled data points decreases the error rates of 4-class text categorization. Zhu et al. (2003) demonstrate that the semi-supervised learning approach based on Gaussian random fields is able to efficiently exploit the structure of unlabeled data to improve accuracy in binary text classification.
All in all, semi-supervised learning is a promising field of research, especially, as more and more mainly unstructured and unlabeled data becomes online accessible.

### 2.2.1 Label Propagation

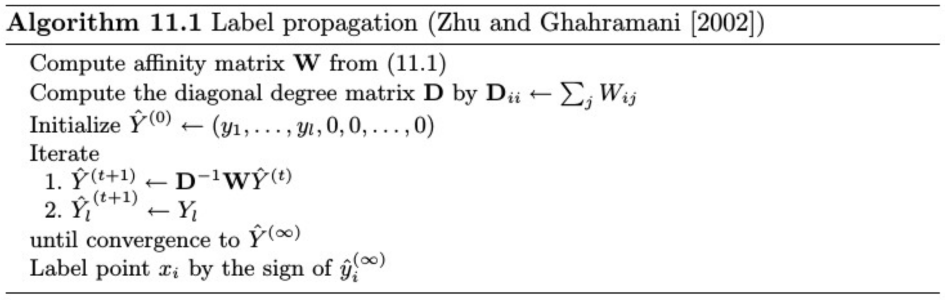
In this section we will have a detailed look at label propagation which is a commonly used graph-based semi-supervised learning approach. Labeled and unlabeled data points are represented as nodes in a graph and connected with weighted edges representing their similarity (Liu et al., 2012). The initially known labels are propagated through the graph until all nodes are assigned to a class (Zhu and Ghahramani, 2002). Bengio et al. (2006) specifies two graph consistency assumptions on which the label propagation algorithm is based:

(1) smoothness assumption – similar or closely related points belong to the same class. As edges in the graph represent similarity, neighboring nodes are likely to have the same label.

(2) cluster assumption – points that lie on a connected graph structure, for example a cluster, are likely to have the same label. Thus, decision boundary are located in low-density regions.

These assumptions play an important role in graph-based semi-supervised learning as their realization and implementation builds the basis for different label propagation algorithms (Zhou et al., 2003).

For an in-depth understanding of label propagation we introduce a mathematical graph notation which is based on Zhu et al. (2005):
Given is a data set $\mathcal{X}$ consisting of labeled samples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_l, y_l)\}$ with $y_i \in \{1, 2, ..., C\}$ and unlabeled samples $\{\mathbf{x}_{l+1}, ...\mathbf{x}_u\}$. $l$ corresponds to the number of labeled data points, $u$ to the number of unlabeled data points and $C$ is the number of different classes. All $n$ data points are either labeled or unlabeled, hence, $n = l + u$, where normally $l << u$. A weighted graph $\mathcal{G} = (\mathcal{X}, \mathbf{W})$ is constructed such that every data point $\mathbf{x}_i$ is represented as a node. Nodes are connected with weighted edges $w_ij$, which together form the weight matrix $\mathbf{W}$.



```
Algorithm 11.1 Label propagation (Zhu and Ghahramani [2002])
  Compute affinity matrix $\mathbf{W}$ from (11.1)
  Compute the diagonal degree matrix $\mathbf{D}$ by $\mathbf{D}_{ii} \leftarrow \sum_j W_{ij}$
  Initialize $\hat{Y}^{(0)} \leftarrow (y_1, \dots, y_l, 0, 0, \dots, 0)$
  Iterate
    1. $\hat{Y}^{(t+1)} \leftarrow \mathbf{D}^{-1}\mathbf{W}\hat{Y}^{(t)}$
    2. $\hat{Y}_l^{(t+1)} \leftarrow Y_l$
  until convergence to $\hat{Y}^{(\infty)}$
  Label point $x_i$ by the sign of $\hat{y}_i^{(\infty)}$
```

Figure 2: Original label propagation algorithm of (Zhu and Ghahramani, 2002). Source: (Bengio et al., 2006).

The original label propagation algorithm was introduced by Zhu and Ghahramani (2002) and since then modified and extended many times. In the following the basic label propagation algorithm, which is also stated in pseudo code in Figure 2, is described:
To estimate the labels of all data points $\hat{Y} = (\hat{Y}_l, \hat{Y}_u)$ the weight matrix $\mathbf{W}$ is normalized by multiplying it with the inverse of its degree matrix $\mathbf{D}$. $\mathbf{D}$ is defined as $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ corresponding to the total number of out-going edges of node $x_i$. Therefore, the matrix product $\mathbf{D}^{-1}\mathbf{W}$ corresponds to the normalized transition matrix indicating how probable it is to transition from one node to another one.

Labels are then propagate through the graph by multiplying the transition probability with the labels: $\hat{Y} = \mathbf{D}^{-1}\mathbf{W}Y$. With $\hat{Y}_l = Y_l$ we ensure that the initial labels are fixed. This process is repeated with $Y = \hat{Y}$ until convergence is reached.

The original label propagation algorithm ensures that the initially known labels do not change over time which is reached by minimizing the following function (Bengio et al., 2006):

$$\sum_{i \in V_l} (\hat{y}_i - y_i)^2 = ||\hat{Y}_l - Y_l||^2 \tag{1}$$

To include the smoothness assumption, rapid changes in the predicted labels $\hat{Y}$ between similar data points are penalized by minimizing the following function:

$$\frac{1}{2} \sum_{i,j \in V} W_{ij} (\hat{y}_i - \hat{y}_j)^2 = \hat{Y}^T (D - W) \hat{Y} = \hat{Y}^T L \hat{Y} \tag{2}$$

The combination of (1) and (2) results in a cost function that contains the trade-off between the smoothness of the predicted labels over the entire graph and the accuracy of the initially given labels $\mathcal{X}_l$ (Liu et al., 2012):

$$C(\hat{Y}) = ||\hat{Y}_l - \hat{Y}_l||^2 + \mu \, \hat{Y}^T L \hat{Y}$$

with $\mu$ regulating the corresponding relevance.
This cost function can be compared to regression: While we want a predicted curve to be as close as possible to the observed data, over-fitting has to be prevented by regularization and smoothing.

This simple idea is the basis for more complex label propagation algorithms that improve performance and running time, and model more complex consistency assumptions or make it possible to include noisy data.

### 2.3 Project Idea

In this project we combine graph-based text representation with semi-supervised learning. We propose a flexible graph representation for collections of documents which includes documents as well as features as nodes. Kannan et al. (2016) uses a similar graph to generate semantic clusters for short e-mail responses. However, while their feature space is limited to short responses of a few words, we extend their approach such that entire text documents are represented. Based on label propagation we investigate the influence of different graph features on a 4- and 20-class text categorization task. By varying the number of labeled documents we compare the graph representation with bag-of-words based supervised and semi-supervised learning.

We propose that a graph-based representation which is able to capture word frequency, sentence structure, and semantic similarity will improve classification in comparison to standard bag-of-words models. Regarding different graph features, we assume that semantic information is, especially for text categorization, more valuable than syntactic information because it allows to overcome the vocabulary gap in short documents.

With a low number of labeled documents we expect the semi-supervised learning approaches to outperform bag-of-words based models, as they incorporate the structure of unlabeled data points during training. On top of that, graph-based semi-supervised learning should be better than bag-of-words based semi-supervised learning as it includes information about syntactic and semantic relations of words.

## 3 Methods

In this section the 20-Newsgroup dataset and the applied methods, from preprocessing, graph construction, feature extraction to graph modulation are described. Figure 3 gives an overview about the involved steps and points out the different settings used for comparison.

Preprocessing, label propagation and analysis are implemented in `python 2.7` under usage of the natural language toolkit (`nltk`)[1] and `scikit-learn`[2]. Furthermore, the graph database management system `Neo4j`[3] is used for text representation.
Preprocessing and graph construction will be demonstrated by a short example, which is based on posts from the baseball newsgroup but modified such that certain properties of the graph construction can be illustrated.

### 3.1 The Dataset

We use the 20-Newsgroup dataset[4], a popular benchmark dataset for text mining tasks, to evaluate graph-based semi-supervised learning and compare it to standard bag-of-words based approaches. The collection consists of about 11 300 newsgroup texts divided into 20 different topics from which some are closely related (Rennie, 2008) while others differ a lot. Table 1 gives an overview about the classes and their respective number of documents. With about 550 to 600 documents, the classes are quite balanced. Exceptions are *talk.religion.misc* with 377 newsgroup posts and *alt.atheism* with 480 documents.

Example – Newsgroup documents

Here are two adapted examples from the *baseball* newsgroup category which we use to illustrate the preprocessing and graph construction steps.

*I'm just so happy that Chicago beat Toronto overtime on Friday!*

*I'm a *classic* Chicago fan. But, in the playoffs the Detroit Toronto games are the BEST.*

[1]http://www.nltk.org/
[2]http://scikit-learn.org
[3]https://neo4j.com/
[4]Different versions of the 20-Newsgroup dataset exist. We use the corpus from `scikit-learn` for which more information can be found here.

Table 1: Overview of topics and respective number of documents in the 20-Newsgroup dataset.

| Topic | N | Topic | N |
|---|---|---|---|
| alt.atheism | 480 | rec.sport.hockey | 600 |
| comp.graphics | 584 | sci.crypt | 595 |
| comp.os.ms-windows.misc | 591 | sci.electronics | 591 |
| comp.sys.ibm.pc.hardware | 590 | sci.med | 594 |
| comp.sys.mac.hardware | 578 | sci.space | 593 |
| comp.windows.x | 593 | soc.religion.christian | 599 |
| misc.forsale | 585 | talk.politics.guns | 546 |
| rec.autos | 594 | talk.politics.mideast | 564 |
| rec.motorcycles | 598 | talk.politics.misc | 465 |
| rec.sport.baseball | 597 | talk.religion.misc | 377 |

## 3.2 Preprocessing

Preprocessing describes the process of obtaining clean text from raw input data (compare Figure 3). The goal is to keep relevant information while reducing the dimensionality of the problem. In text categorization this corresponds to normalizing words, removing special characters as well as irrelevant or noisy information.

For the 20-Newsgroup dataset we use the `sklearn` built-in option to remove headers, (e.g. subject title, mail address), footers (e.g. personal signature, favorite quote, etc.) and references from different posts, in order to avoid duplicate data or content-independent information. However, to make a graph-based representation of the entire document collection computationally feasible, further preprocessing is required. Therefore, we reduce the amount of unique words in the text documents:
The documents are tokenized based on Verberne et al. (2016). All tokens are changed to lowercase letters and lemmatized such that inflectional forms of a term are reduced to a common basis (Manning et al., 2008). For example, plural forms are changed to singular forms, such as *houses* → *house*, *mice* → *mouse*, and tense inflections are changed to the original verb, e.g. *showed* → *show*, *written* → *write*. For lemmatization the `nltk WordNetLemmatizer` is used.
Commonly used words such as articles or conjunctions often do not contribute to the meaning of a sentence and hence, are removed based on a predefined list of stopwords[5]. Furthermore, words that appear less than 10 times in the entire document collection, as well as words that appear in more than 50% of the documents are removed. The remaining unique words are organized in a vocabulary and documents that do not contain any word from the vocabulary are removed from the dataset. The remaining documents are separated into lists of sentences containing the preprocessed unique words.

[5]See here for the full list of stopwords

Example – Preprocessing

Preprocessing converts the text documents into lists of sentences. The sentences themselves are split into lemmatized words from which stopwords are removed. Applying preprocessing to the example documents results in:

[just, happy, chicago, beat, toronto, overtime, friday]

[classic, chicago, fan]
[playoff, detroit, toronto, game, best]

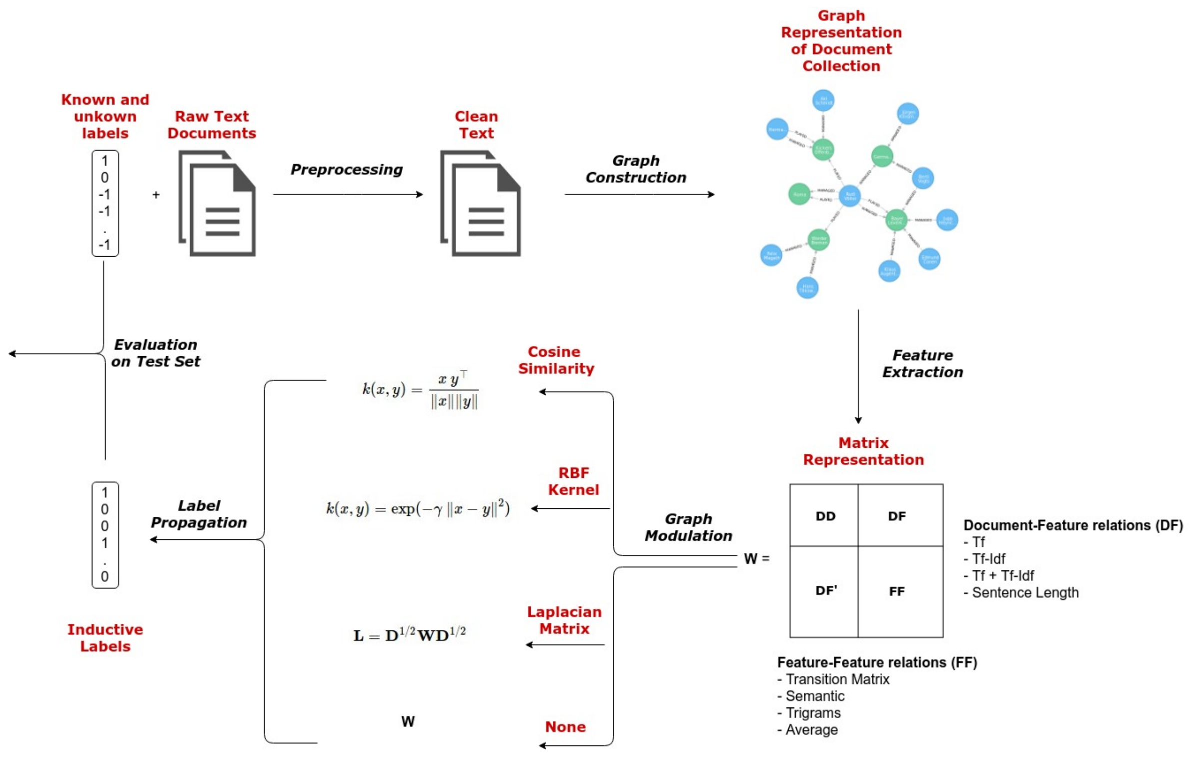


Figure 3: Visualization of the step-by-step process for graph-based semi-supervised learning. The raw documents and the corresponding label vector that indicates unlabeled data with -1 are the starting point of the approach. By preprocessing the documents, clean text that is split in sentences represented as lists of words is obtained. In graph construction, we iterate through the sentences and add respective words and relations to the graph until the full collection of documents is represented. In order to use the graph properties for label propagation, relevant features have to be extracted and converted into a square matrix consisting of document-document, document-feature and feature-feature relations. We will investigate different graph properties, as well as graph modulation which induce the similarity and cluster assumptions required for good performance of the algorithm. As we are interested in comparing the influence of different graph properties on the performance the obtained labels of a test set will be compared to the true labels.

## 3.3 Graph Construction

Based on the preprocessed texts a graph of the entire document collection is constructed. For this purpose, we use the `Neo4j` graph database in which nodes represent entities while edges illustrate their relation. Both, nodes and edges, can contain further properties such as names, identifiers or weights.

We distinguish between two different node types:

> **Document Nodes** represent a document, in this case a newsgroup post, and contain a unique name as well as a property indicating their label.
>
> **Feature Nodes** represent a feature of the collection, such as unique words and the special features *$Start$* and *$End$* which indicate the beginning and end of a sentence. Feature nodes contain a name and a numerical identifier.

For each document a document node with its corresponding properties is created. Similarly, every unique word is added to the graph as a feature node. For every word that appears in a document, their corresponding document and features nodes are connected via an *is_in* edge. Syntactic structure between words is captured by relating successive words within a sentence with a *followed_by* edge. The beginning and end of a sentence are furthermore connected to the *$Start$* and *$End$* feature nodes.
Both edge types, *followed_by* and *is_in*, have a count that is increased as soon as a word appears more often in the same document or a sequence of words is repeated within the entire collection. To normalize the frequency counts they are divided by the total number of in-coming edges of the node the edge is pointing to. Thus, all edge weights range between 0 and 1. The document-feature edge indicates the relative frequency of a word in a document while the relation between feature indicates the probability of transitioning to a certain word.
Please note, that due to the removal of stopwords, we do not capture the actual syntax of the documents. Features that normally not occur next to each other in a sentence are in the graph representation connected with a *followed_by* edge. However, when looking at the entire dataset, the edge weight of such unusual word combinations that do not reflect syntactic information, go to zero while more frequent and relevant word sequences take over. Also, by normalizing with the in-degree, not the probability of transitioning from node $x$ to node $y$, but the probability of $x$ being the predecessor of $y$ is captured. This illustrates a fundamental problem of directed edge weights in a graph-based representation to which we come back in the dicussion section. However, for the purpose of document classification the actual syntax is not as relevant as the possibility to capture relevant word sequences like n-grams with different length.

Example – Graph Construction

Figure 4 shows the graph representation of the preprocessed example documents. Feature nodes (blue) are connected to the documents (green) they appear in. The feature-feature edges connect the unique words to sentences [...]

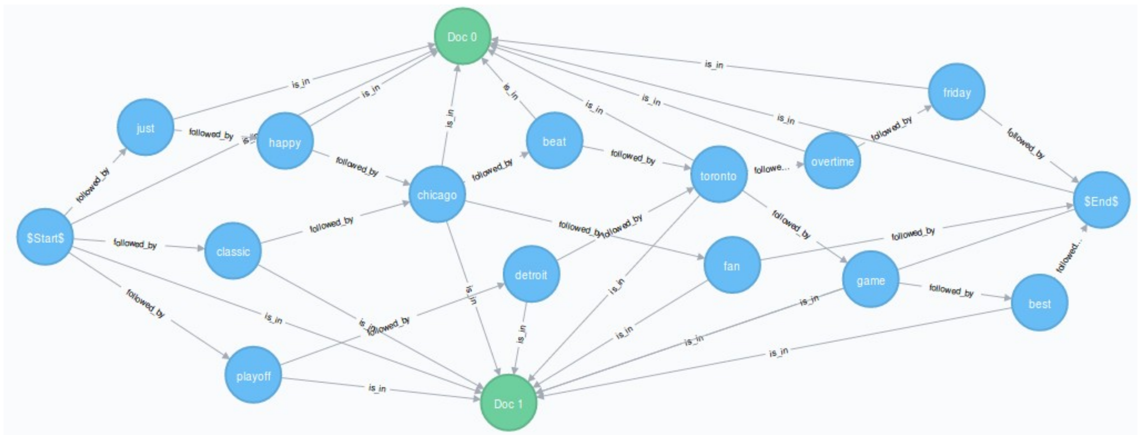
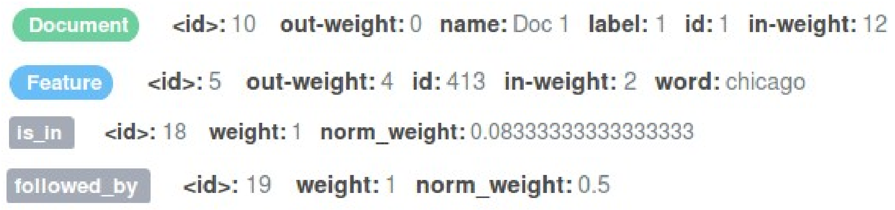


Figure 4: Graph-based representation of the two example documents. Each document is represented as a green node with a unique name and an identifier. The blue nodes correspond to the features, unique words and *$Start$*, *$End$* features, of the documents. Edges between feature nodes, labeled as $followed\_by$ indicate the syntactic structure of the documents while the weight of the *is_in* edges corresponds to the normalized word frequency.

[...] In Figure 5 the properties of specific nodes and relations of the example graph are illustrated. The document node for the example text (right side in the box) is called *Doc 1* and has no out-going edges, but 12 in-coming edges which corresponds to the number of words (8) plus *$Start$* and *$End$* multiplied by 2, as there are two sentences in the post. For example, the word *chicago* appears once in the post and therefore has a *is_in* weight of 1 which normalized by the total number of in-weights is $\frac{1}{12} = 0.08\overline{3}$. Regarding syntactic information, the feature node *chicago* has two in-coming edges which means it has two preceding words. In this example, they are from different posts. Furthermore, the feature node has four out-going edges which include the two succeeding words, as well as the *is_in* relation to each of the documents. The normalized weight to the preceding feature node representing the word *classic* is 0.5.



Document <id>: 10 out-weight: 0 name: Doc 1 label: 1 id: 1 in-weight: 12

Feature <id>: 5 out-weight: 4 id: 413 in-weight: 2 word: chicago

is_in <id>: 18 weight: 1 norm_weight: 0.08333333333333333

followed_by <id>: 19 weight: 1 norm_weight: 0.5

Figure 5: Both node types, document (green) and feature (blue), have an **id**, an **in-weight**, **out-weight** and a comprehensive identifier, **name** for document nodes and **word** for feature nodes. Document nodes have an extra **label** property indicating their corresponding class, if known. Both relation types (gray), *is_in*, and *followed_by* have a **weight** describing the frequency of the particular relation and its normalized weight. The **norm_weight** is obtained by dividing through the total number of in-weights of the node they are pointing to. <**id**> corresponds to the internal `Neo4j` identifier.

## 3.4 Graph-based Features

While graphs are able to capture multiple complex relations, in order to apply conventional semi-supervised learning approaches, such as label propagation, a matrix representation of the graph is necessary. In this section, we illustrate the matrix representation and look in more detail at the features which are captured in the graph or can be inferred from it.

In the matrix representation each node with its respective relations to all other nodes is stated. As we distinguish between document and feature nodes in the graph we will also use this difference to analyze the matrix representation $\mathbf{W}$. With $n$ document and $m$ feature nodes the corresponding matrix $\mathbf{W}$ has the shape $n + m \times n + m$. It can be divided into the following parts:

$$\mathbf{W} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n} & | & df_{1,1} & df_{1,2} & df_{1,3} & \cdots df_{1,m} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n} & | & df_{2,1} & df_{2,2} & df_{2,3} & \cdots df_{2,m} \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,n} & | & df_{n,1} & df_{n,2} & \cdots & df_{n,m} \\ -- & -- & -- & -- & -- & -- & -- & -- & -- \\ fd_{1,1} & fd_{1,2} & \cdots & fd_{1,n} & | & f_{1,1} & f_{1,2} & \cdots & f_{1,m} \\ fd_{2,1} & fd_{2,2} & \cdots & fd_{2,n} & | & f_{2,1} & f_{2,2} & \cdots & f_{2,m} \\ fd_{3,1} & fd_{3,2} & \cdots & fd_{3,n} & | & f_{3,1} & f_{3,2} & \cdots & f_{3,m} \\ \vdots & \vdots & \ddots & \vdots & | & \vdots & \vdots & \ddots & \vdots \\ fd_{m,1} & fd_{m,2} & \cdots & fd_{m,n} & | & f_{m,1} & f_{m,2} & \cdots & f_{m,m} \end{pmatrix}$$

Where $d$ indicates a document-document relation, $df$ and $fd$ a document-feature relation and $f$ a feature-feature relation. For further analysis we split the matrix into four parts: a document $\mathbf{DD}$, a feature $\mathbf{FF}$ and a document-feature $\mathbf{DF}$ matrix.

$$W = \begin{pmatrix} \mathbf{DD} & \mathbf{DF} \\ \mathbf{DF}^T & \mathbf{FF} \end{pmatrix}$$

### Document Matrix

The document matrix, $\mathbf{DD}$, describes the relations or similarity between documents. This can, for example, be based on the author, publication time, journal, topic or any other measure that directly or indirectly relates two documents. As we do not include such information, the document matrix corresponds to a $n \times n$ dimensional identity matrix $\mathbf{DD} = \mathbf{I}_n$. This reflects that each document is identical to itself, while having no specified relation to other documents.

### Document-Feature Matrix

The document-feature matrix, $\mathbf{DF}$, captures information that relates document with feature nodes, for example based on their frequency. With $n$ document and $m$ feature nodes, $\mathbf{DF}$ is a $n \times m$ matrix. In the following we list the different document feature relations that we consider in our experiments:

*Term frequency (Tf)* – When word appears in a document their corresponding nodes are connected with an *is_in* edge. The edge includes a frequency count which, in order to obtain the normalized term frequency, is divided by the total number of words appearing in a document. Thus, the weights range between 0 and 1 and sum up to 1 within a document.

*Term frequency - inverse document frequency (Tf-Idf)* – When weighting the importance of a word within a document, *Tf-Idf* takes besides the term frequency also its frequency in the entire document collection into account. The underlying assumption is that very frequent terms in a collection of documents do not carry relevant information to differentiate documents with a certain label from documents with another one. For example, in a collection of articles about sports the word *hockey* carries more information than the word *sport* even though the latter might be more frequent. We use `scikit-learn` *Tf-Idf* implementation which is computed as:

$$Tf\text{-}Idf(t) = Tf(t,d) \cdot Idf(t)$$
$$Idf(t) = log(\frac{1+N}{1+Df(d,t)}) + 1$$

with $N$ being the total number of documents and $Df(d,t)$ being the number of documents that contain the term $t$.

*Tf+Tf-Idf* – As the weight matrix $\mathbf{W}$ includes twice the document-feature matrix, once as $\mathbf{DF}$ and once as $\mathbf{DF}^T$ the previously discussed features can be combined.

*Sentence Count* – During graph construction two additional features nodes, \$Start\$ and \$End\$, are introduced to emphasize the sentence structure. Like other feature nodes they are connected to the document nodes they appear in and hence, give information about the number of sentences. Note that the sentence count is not a full document-feature matrix but only adds the \$*Start*\$ and \$*End*\$ columns, with their respective document relations, to one of the previously described matrices.

**Feature Matrix**

The feature matrix **FF** describes how features relate to each other. Again, all kind of measures such as syntactic or semantic similarity, sentence co-occurrences, etc. are possible. With $m$ features, which are in our case unique words, **FF** is a $m \times m$ matrix. Dependent on the chosen weight measure the matrix is either symmetric or asymmetric.
The following text features are used in this report:

*Transition Matrix* – By constructing the graph such that successive words in a sentence are connected with an edge, the feature-feature relation indicates how often a certain feature $f_j$ follows feature $f_i$. Normalizing the count weights leads to a transition probability matrix. Here, the transition probability indicates how probable it is that feature $f_i$ precedes feature $j$. The removal of stopwords during preprocessing introduces noise, but nevertheless consistent syntactic structure is captured and common word sequences have a high transition probability.

*Trigrams* – The transition probability can also be interpreted as an indicator for a path between two feature nodes. When multiplying such a connectivity matrix $n$ times with itself, paths of length $n$ are detected (Raluca Tanase, 2009). We use the square matrix of **W** to investigate the influence of trigram relations on text classification performance.

*Context Similarity* – If words appear in similar contexts, they are often from a certain word class or family that can be substituted with each other (Lyon, 2015), for example, weekdays like *Monday* and *Thursday.* The same is true for words from other semantic classes and subclasses such as animals, birds, objects, body parts, etc. Including semantic similarity in the matrix representation could link documents with the same category even though they use different words.
The context of a feature is defined as all the words in its direct environment as predecessors or successors. In order to measure the similarity of two words $w_1$ and $w_2$ with their respective context sets $C_1$ and $C_2$, the Jaccard index is used:

$$J(C_1, C_2) = \frac{\|C_1 \cap C_2\|}{\|C_1 \cup C_2\|}$$

As in text data the sentence structure is relevant, we distinguish between the preceding and succeeding context of a word. Thus, the context similarity between two features is computed by averaging the Jaccard indices of the preceding and succeeding context. To ensure the computational feasibility of this method, only feature pairs with a normalized transition probability higher than 0.1 are considered.

*Average* – To combine these different graph properties we include the average of their corresponding feature matrices in the analysis.

## 3.5 Graph Modulation

Besides including relevant graph properties, to successfully use the modeled matrix **W** in label propagation, it has to fulfill the smoothness and cluster assumptions discussed in Section 2.2.1.
Kernel methods compute the pairwise similarity of all elements in a matrix and are often used to implement smoothness in semi-supervised learning. We use and combine three kernel methods to investigate their effect on text categorization performance:

*Cosine Similarity* – is commonly used to measure document similarity in the vector space model. It is defined as the dot product of two vectors divided by their magnitude:

$$k(x, y) = cos\Theta = \frac{x\ y^\top}{\|x\|\|y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2}\sqrt{\sum_{i=1}^n y_i^2}}$$

If vectors are orthogonal their cosine similarity is 0, whereas it is 1 if they are parallel. Vectors with opposing directions have a cosine similarity of -1.

*RBF Kernel* – is commonly used in support vector machines and defined as:

$$k(x, y) = \exp(-\gamma\ \|x - y\|^2) = \exp(-\frac{1}{\sigma^{-2}}\ \|x - y\|^2)$$

where $\sigma^{-2}$ describes the variance of a Gaussian distribution. $\gamma = \frac{1}{\sigma^{-2}}$ indicates the reach or influence of a single data point. The higher $\gamma$, the more local, the lower the more global the reach. Even small changes in the $\gamma$-value have significant influence on the model (Wang and Zhang, 2008). Figure 6, taken from Pedregosa et al. (2011), illustrates these effects.

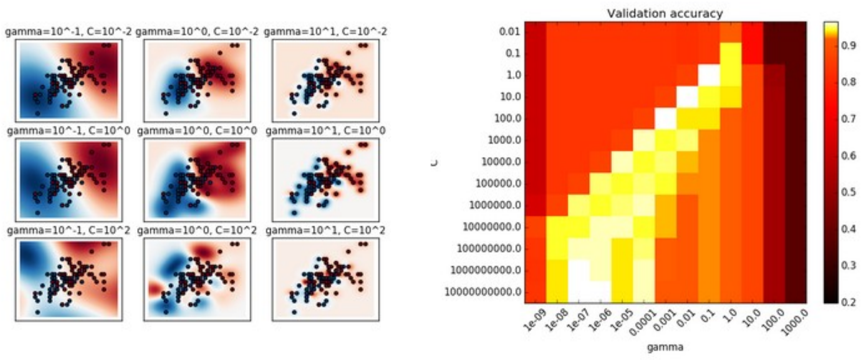


Figure 6: Left: Illustration of how the $\gamma$ parameter determines the influence of a single training example in a binary classification task with an SVM, which is why the $C$ parameter, which influences the smoothness of the decision surface, is included. While with a too low $\gamma$ the complexity of the data can not be captured (upper left), a too high $\gamma$ yields a too small radius of influence such that only one training sample at a time is included (lower right). Right: Influence of $C$ and $\gamma$ on cross-validation accuracy is illustrated as a heat map. Already small changes in $\gamma$ lead to significantly different results. Figures are taken from (Pedregosa et al., 2011).

*Laplacian Matrix* – Liu et al. (2012) argues that while the RBF kernel effectively establishes local consistency, additionally smoothing with the Laplacian improves global consistency because unreliable edges between points that are relatively far apart are removed. The normalized Laplacian is applied in addition to the RBF kernel and is defined as:

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$$
$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

with **W** being the weight matrix and **D** its corresponding degree matrix, defined as $\mathbf{D} = diag([D_{11}, ...D_{nn}])$ with $D_{ii} = \sum_{j=1}^{n} W_{ij}$.

## 4 Experiments

The experiment section is split in two parts:

(1) analysis of the influence of graph-based features on text categorization performance

(2) comparison of graph-based semi-supervised learning and bag-of-words based (semi-) supervised learning approaches with a varying number of labeled documents

Furthermore, to ensure the stability of the observed effects, both experiments are conducted on two text categorization tasks that differ in the number of classes.

### 4.1 Text Categorization Tasks

The *rec* subset in the 20-Newsgroup data consists of *rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey* and is used as a 4-class categorization task. 20-class categorization is performed on the entire 20-Newsgroup dataset. See Table 1 for an overview of the different categories and their corresponding number of documents.

The `sklearn` 20-Newsgroup dataset includes about 300 empty documents. Moreover, the removal of stopwords or too less frequent words during preprocessing leads to further empty documents that are excluded from the dataset. In 4-class categorization from the originally 2389 documents 102 are empty. Similarly, in 20-class categorization 335 empty documents are removed such that 10979 remain. Table 2 illustrates these numbers and gives information about the number of unique words in the different text categorization tasks. In 4-class categorization 1836 features are identified in the document collection, while in 20-class categorization there are 7002 features consisting of the 7000 most frequent unique words and the special features *$Start$* and *$End$*. The average document length in the 20-Newsgroup data is 190 words.

Table 2: Characteristics of the two text categorization datasets. The total number of documents is assessed after preprocessing where empty documents are removed. The number of features corresponds to all unique words plus the *$Start$* and *$End$* features that indicate the beginning and end of a sentence.

| Categorization Task | Classes | Removed documents | Total number of documents | Number of features |
|---|---|---|---|---|
| 4-class | autos, motorcycles baseball, hockey | 102 | 2287 | 1836 |
| 20-class | all 20-Newsgroup classes | 335 | 10979 | 7002 |

## 4.2 Graph-based Feature Analysis

As discussed in Section 3.4 a graph-based representation of a collection of documents has the advantage to capture different properties and relations between documents and features. However, standard semi-supervised learning algorithms require a matrix as an input which in turn limits the possibility to express the different text properties and their complex way of interaction.
Therefore, we investigate how the different graph-based features influence the performance of semi-supervised learning on text categorization tasks. For semi-supervised learning, slightly modified versions of the label propagation and label spreading algorithm in
`sklearn.semi_supervised` are used. The modification ensures that implemented graph modulations such as the RBF kernel can be circumvented or replaced by different modulations.

First, we analyze the effect of different document-feature combinations. As illustrated in Section 3.4 and Figure 3, we use the normalized term frequency ($Tf$), normalized term frequency - inverse document frequency (*Tf-Idf*) or even both (*Tf+Tf-Idf*), as the **DF** matrix occurs twice in the weight matrix $\mathbf{W}$. For the feature-feature relations the transition matrix is used as default. These different matrix settings are compared to standard semi-supervised learning based on a *Tf-Idf* weighted bag-of-words model. This condition is called *baseline* in the result section.
Furthermore, for each setting different combinations of graph-modulations such as cosine similarity, RBF kernel and Laplacian matrix are applied. The RBF kernel parameter is set to $\gamma = 5$ after a 3-fold cross-validated parameter search in the interval of $[0.15, 10]$. This value is used for all experiments and settings that include an RBF kernel.

As *Tf+Tf-Idf* emerges to have the highest performance with different graph modulations, this will be used as default for the **DF** matrix when different feature-feature relations are analyzed. As discussed in Section 3.4, the **FF** matrix can be represented as transition matrix, trigrams, context similarity or the average of these properties. Additionally we investigate how adding information about sentence length to the transition matrix influences semi-supervised learning performance in the selected categorization tasks. Again, the results are listed with different graph modulations.

For both text categorization tasks and all different settings, 100 documents from each class are randomly selected as labeled data points. To ensure that the performance is not dependent on the selected labeled documents, the experiment is repeated three times and the resulting mean F-scores and standard deviations are reported.

## 4.3 Number of labeled documents

As underlined throughout this report, for many tasks in the field of text mining an immense amount of unlabeled data is accessible while labeled data is rare or associated with time consuming manual effort. Therefore, the second experimental part deals with the question how the number of labeled documents used for semi-supervised learning influences the performance and how this compares to bag-of-words based standard approaches in supervised and semi-supervised learning.

From the previous experiment the best performing **DF** and **FF** matrices with their corresponding graph modulation is chosen: For document-feature relations it is the *Tf+Tf-Idf* condition, while it is context similarity for the feature-feature relations. In addition to the RBF kernel with $\gamma = 5$, the Laplacian matrix is used for graph modulation. This setting will be referred to as graph-based semi-supervised learning (*graph-based SSL*).
To evaluate the influence of the graph-based text representation, this setting is compared to semi-supervised learning with the same graph modulations based on a *Tf-Idf* vector representation (*bag-of-words SSL*). Furthermore, to assess the performance of semi-supervised learning compared to a standard supervised learning approach, a linear support classification implemented in `sklearn.svm.LinearSVC` is used. It is trained on *Tf-Idf* vector representation with default parameters. It will be referred to as bag-of-words based supervised learning (*bag-of-words SL*).
The different approaches are compared by varying the number of labeled documents from 1 to 350 documents per category. The labeled documents are selected randomly. To marginalize the effect of the random selection the full experiment is conducted ten times and the average F-scores with their corresponding standard deviations are reported.
The experiment is conducted for both, the 4-class and the 20-class, categorization task.

## 4.4 Evaluation

The true labels in the 20-Newsgroup dataset are used to evaluate the prediction outcomes of the unlabeled documents in the different learning approaches.
In information retrieval, *precision*, the percentage of correctly as relevant identified documents, and *recall*, the percentage of relevant documents that have been identified, are commonly used to evaluate the performance of an algorithm (Forman, 2003). Transferred to a binary classification task, a relevant document corresponds to a document that belongs to the class of interest. Precision and recall are then computed as:

$$precision = \frac{TP}{TP + FP}$$
$$recall = \frac{TP}{TP + FN}$$

where *TN* stands for the number of true negatives, *TP* for the number of true positives, *FN* for the number of false negatives and *FP* for the number of falsely positive classified documents in a binary classification task. Both measures can be combined to the *F-score* which corresponds to the harmonic mean of precision and recall:

$$F\text{-}score = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

This form of evaluating a binary classification task can be extended to multiple classes. The, so called macro average, computes precision, recall and F-score for each class and then averages the results. Thus, each class contributes equally, independent of its size, to the average. Despite the discussion whether a proportionally weighted contribution to the average is more accurate, we use the macro average as the class sizes in the 20-Newsgroup dataset are relatively balanced.

# 5 Results

The result section is separated into the graph-based feature analysis and the classifier performance with a varying number of labeled documents. Both sections are further split into 4- and 20-class categorization. At the end, we discuss additional insights obtained by graph-based semi-supervised learning.

## 5.1 Graph-based Feature Analysis

For both categorization tasks the results are summarized in two tables.
In the first table different document-feature relations and graph modulations are compared. Note that the label propagation algorithm is based on matrix multiplication which requires a square matrix as input. Therefore, without a graph modulation the *Tf-Idf* vector representation does not yield any result.
The second table focuses on the effect of different feature-feature relations, while using *Tf+Tf-Idf* as **DF** matrix.

Both tables display the mean F-scores, with their corresponding standard deviations, averaged over three runs in which 100 randomly selected documents of each class are used as labeled data. In every row the best result is highlighted and results that differ significantly from it are marked with an asterisk. Statistical significance is assessed with an unpaired t-test at significance level 0.05.

### 5.1.1 4-class Text Categorization

Without the usage of an RBF kernel the mean F-scores are, independent of the used document-feature relations, with around 10% impractically low (compare first two rows, *None* and *cos*, in Table 3). Adding an RBF kernel with $\gamma = 5$ improves the performance in all conditions, except for the *Tf* condition. In combination with *cos* the F-scores further increase. Nevertheless, the results obtained in the different *Tf* settings are significantly lower than the highest F-score in each row. Also, the standard deviations of 21.93% and 7.69% are unusually high.

Table 3: Comparison of document-feature matrices in 4-class text categorization task. The table displays the mean F-scores, averaged over 3 runs with 100 randomly selected training documents per class, and their corresponding standard deviations. The discussed document-feature matrices are combine with different graph modulations, where $\gamma = 5$ for the RBF kernel. As a baseline semi-supervised learning based on the *Tf-Idf* bag-of-words model is used. In each row, the best results are printed in bold and F-scores that differ significantly are marked with an asterisk.

| | Baseline | Tf | Tf-Idf | Tf + Tf-Idf |
|---|---|---|---|---|
| None | - | 9.87 ± 0.00 | 9.95 ± 0.19 | 9.95 ± 0.21 |
| cos | 10.08 ± 0.19 | 9.91 ± 0.00 | 10.08 ± 0.19 | 10.08 ± 0.19 |
| RBF | 75.10 ± 2.00 | 10.19* ± 0.23 | 74.32 ± 2.97 | **77.76** ± 0.43 |
| RBF + cos | 70.37* ± 1.38 | 47.78* ± 21.93 | 73.25* ± 0.41 | **75.72** ± 0.71 |
| RBF + Laplace | 76.44* ± 0.51 | 9.91* ± 0.23 | 74.75 ± 2.61 | **77.96** ± 0.39 |
| RBF + cos + Laplace | 73.40 ± 1.19 | 26.97* ± 7.69 | 74.75* ± 0.37 | **75.05** ± 0.16 |

The *Tf+Tf-Idf* condition has with about 75.4% (with *cos*) to 77.8% (without *cos*) constantly the highest F-scores. Compared to the *Baseline*, which corresponds to semi-supervised learning based on the *Tf-Idf* bag-of-words representation, the F-scores differ significantly or are, in the *RBF+cos+Laplace* condition, with $p = 0.076$ close to be statistically significant. An exception is the *RBF* condition.
Compared to the *Tf-Idf* condition the best results differ significantly as soon as cosine similarity is involved in graph modulation. With an F-score of 77.96% the *Tf-Idf* condition paired with the *RBF+Laplace* graph modulation yields the overall best performance.

Therefore, we use *Tf+Tf-Idf* as **DF** matrix for the comparison of the different graph-based feature-feature relations (see Table 4). With an F-score of about 78.1%, *context similarity* yields the best performance, but only if no *cosine similarity* is applied. Also, note that the *transition matrix*, *trigrams* and the *average* only differ slightly and without statistical significance from the highest F-score.

Table 4: Comparison of feature-feature matrices in 4-class text categorization task. The table displays the mean F-scores which are averaged over 3 runs with 100 randomly selected training documents per class and their corresponding standard deviations. *Tf+Tf-Idf* is used as **DF** matrix, combined with different feature-features relations and evaluated based on different graph modulations with $\gamma = 5$ for the RBF kernel. In each row, the best results are printed in bold and F-scores that differ significantly are marked with an asterisk.

| | Transition Matrix | Trigram | Context Similarity | Average | Transition Matrix Sentence Length |
|---|---|---|---|---|---|
| RBF | 77.76 ± 0.43 | 77.89 ± 0.32 | **78.14** ± 0.57 | 78.11 ± 0.28 | 60.30* ± 2.16 |
| RBF + cos | **75.72** ± 0.71 | 74.90 ± 0.55 | 72.01* ± 0.65 | 72.26* ± 0.79 | 75.56 ± 1.53 |
| RBF + Laplace | 77.96 ± 0.39 | 77.96 ± 0.25 | **78.13** ± 0.62 | 78.10 ± 0.31 | 63.22* ± 3.54 |
| RBF + cos + Laplace | 75.05 ± 0.16 | 71.72 ± 1.71 | 71.89 ± 1.77 | 70.62 ± 1.53 | **75.53** ± 3.29 |

Adding information about the sentence length to the *transition matrix* decreases performance by $15 - 17\%$ when the F-score is assessed without *cosine similarity*. However, when *cosine similarity* is involved in graph modulation, the performance does not significantly differ from the best results which are around 75.5%. Furthermore, in all different sentence length settings the corresponding standard deviations are higher than usual.

### 5.1.2 20-class Text Categorization

In general, the results of 20-class text categorization are consistent with the previously discussed performance of graph-based semi-supervised learning in 4-class text categorization. Table 5 displays the mean F-scores based on different **DF** settings. Without the usage of an RBF kernel the performances of about 0.5% are impractically low (compare *None* and *cos* row in Table 5).

When using an RBF kernel, the *Tf+Tf-Idf* and *Tf-Idf* document-feature relations have, with about 56% (with *cos*) and 62% (without *cos*), constantly the highest F-scores. In all four graph modulations, *RBF*, *RBF+cos*, *RBF+Laplace* and *RBF+Laplace+cos*, are these results significantly higher than the F-scores in the *Baseline* and *Tf* condition.

Table 5: Comparison of document-feature matrices in 20-class text categorization task. The table displays the mean F-scores, averaged over 3 runs with 100 randomly selected training documents per class, and their corresponding standard deviations. The discussed document-feature matrices are combine with different graph modulations, where $\gamma = 5$ for the RBF kernel. As a baseline semi-supervised learning based on the *Tf-Idf* bag-of-words model is used. In each row, the best results are printed in bold and F-scores that differ significantly are marked with an asterisk.

| | Baseline | Tf | Tf-Idf | Tf - Tf-Idf |
|---|---|---|---|---|
| None | - | 0.50 ±0.01 | 0.52 ±0.00 | 0.49 ±0.00 |
| cos | 0.52 ±0.00 | 0.52 ±0.00 | 0.52 ±0.00 | 0.52 ±0.00 |
| RBF | 52.88* ±1.91 | 0.52* ±0.02 | 60.82 ±1.70 | **61.97** ±0.36 |
| RBF + cos | 46.67* ±0.37 | 9.55* ±6.51 | 54.91±0.68 | **55.48** ±0.61 |
| RBF + Laplace | 59.69* ±0.81 | 0.68* ±0.28 | 61.00 ±1.44 | **62.01** ±0.28 |
| RBF + Laplace + cos | 48.05 *±0.25 | 7.01* ±2.69 | **57.37** ±0.21 | 56.76 ±0.82 |

Table 6 shows the mean F-scores for 20-class text categorization when different graph-based feature-feature relations are combined with the *Tf+Tf-Idf* **DF** matrix. Again, *context similarity* shows the highest F-scores (63%) when no *cosine similarity* is applied. However there is no significant difference when other feature-feature relations are used as a **FF** matrix. Adding *cosine similarity* yields a performance drop of about 6%.
Similar to 4-class categorization, information about *sentence length* leads without *cosine similarity* to a significant deviation form the best results, while when it is applied there is no significant difference between the resulting F-score and the respectively best performing feature setting.

In Section 6 we come back to these results and discuss possible explanations for the observed effects.

Table 6: Comparison of feature-feature matrices in 20-class text categorization task. The table displays the mean F-scores which are averaged over 3 runs with 100 randomly selected training documents per class and their corresponding standard deviations. *Tf+Tf-Idf* is used as **DF** matrix, combined with different feature-features relations and evaluated based on different graph modulations with $\gamma = 5$ for the RBF kernel. In each row, the best results are printed in bold and F-scores that differ significantly are marked with an asterisk.

| | Transition Matrix | Tri-gram | Context Similarity | Average | Transition Matrix Sentence Length |
|---|---|---|---|---|---|
| RBF | 61.97 ±0.36 | 62.00 ±0.35 | **62.07** ±0.37 | 62.03 ±0.34 | 46.73* ±2.87 |
| RBF + cos | **55.48** ±0.61 | 55.36 ±0.51 | 52.70* ±0.41 | 52.88* ±0.35 | 54.27 ±0.55 |
| RBF + Laplace | 62.01 ±0.28 | 62.04 ±0.28 | **62.09** ±0.28 | 62.05 ±0.31 | 52.08* ±1.33 |
| RBF + Laplace + cos | **56.76** ±0.82 | 56.05 ±0.79 | 55.30 ±0.63 | 55.05 ±0.82 | 56.53 ±0.64 |

## 5.2 Number of labeled Documents

Figure 7 and Figure 8 display how the number of labeled documents influences 4- and 20-class text categorization performance. Shown are the mean F-scores that are averaged over 10 runs in which the labeled documents, that are used for training, are randomly selected. The corresponding standard deviations are plotted as error bars.

Graph-based semi-supervised learning with *Tf+Tf-Idf* and *context similarity* as **FF** matrix (graph-based SSL), is compared to *Tf-Idf* based bag-of-words semi-supervised (bag-of-words SSL) and supervised learning (bag-of-words SL). For both semi-supervised learning approaches the graph is modulated by an RBF kernel with $\gamma = 5$ and the normalized Laplacian matrix.
The number of labeled documents per class range from 1 to 350.

### 5.2.1 4-class Text Categorization

All classifiers improve the mean F-scores with an increasing number of labeled documents (compare Figure 7). While *bag-of-words SSL* is, when using only one labeled document per class, with an F-score of 18.86% lower than chance level, *bag-of-words SL* and *graph-based SSL* start of with 34.49% and 35.11%. With 350 labeled documents the F-scores of *bag-of-words* and *graph-based SSL* are with 83.5%, about 7% higher than the *bag-of-words SL* mean F-score.

In general, *graph-based SSL* has constantly the best performance. However, with a small number of labeled documents the *bag-of-words SL* approach shows a similar performance. Increasing the number of labeled documents, though, increases their difference in performance. The opposite is true for *bag-of-words SSL*. While with a low number of labeled documents it performs worse than chance, from about 70 labeled documents per class its F-scores are equivalent to the ones obtained from *graph-based SSL*.

Also note the decrease of the standard deviation with an increasing number of labeled documents which is especially apparent in both semi-supervised learning approaches. For supervised learning the standard deviations seem to be consistent.

A one-way repeated ANOVA with $F(2, 20) = 69.118$, $p = 1.04 \cdot 10^{-9}$ shows that the different classification methods significantly influence the average F-score in a 4-class text categorization task. A post hoc test with Tukey-Kramer correction shows that *graph-based SSL* differs with $p = 4.04 \cdot 10^{-5}$ significantly from *bag-of-words SSL*, as well as from *bag-of-words SL* ($p = 1.53 \cdot 10^{-5}$). With $p = 8.9 \cdot 10^{-4}$ also *bag-of-words SSL* and *bag-of-words SL* differ significantly.

### 5.2.2 20-class Text Categorization

The results for 20-class categorization are displayed in Figure 8. Compared to 4-class categorization similar effects can be observed. *Bag-of-words SL* and *graph-based SSL* start off with a mean F-score of about 16.38% for one labeled document per class, while *bag-of-words SSL* shows with 7.24% the lowest result. Nevertheless, with about 50 labeled documents *bag-of-words SSL* catches up with the supervised learning approach. Together with *graph-*

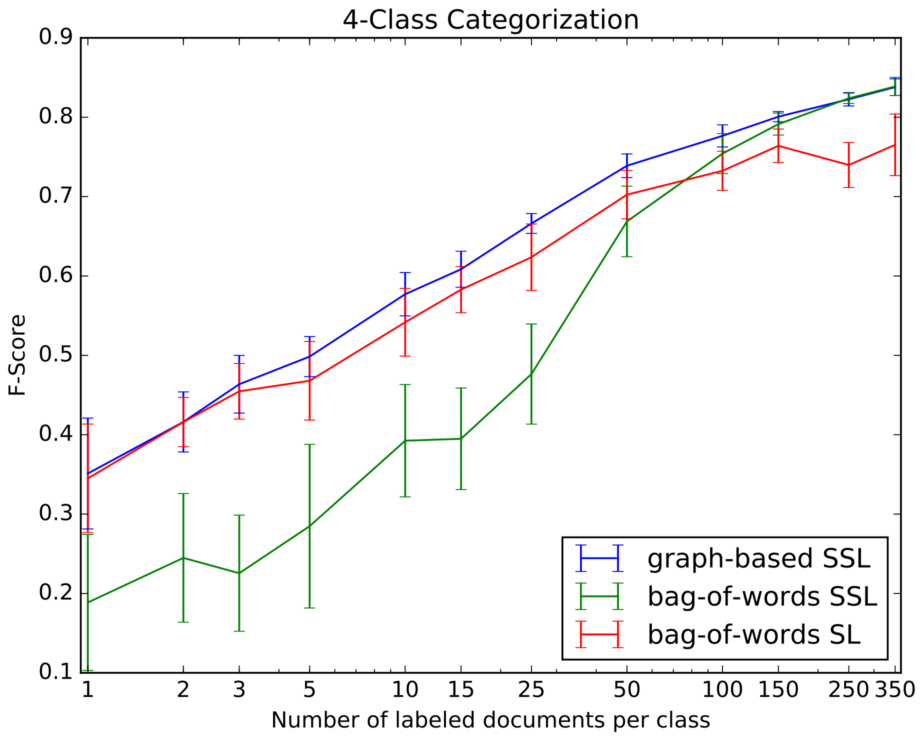


Figure 7: Classifier comparison for 4-class text categorization with a varying number of labeled documents. Displayed are the mean F-scores averaged over 10 independent runs, in which the labeled documents are selected randomly for each class. The error bars show the corresponding standard deviation.

*based SSL* the highest F-score is about 68% for 350 labeled documents per class.

Again, the semi-supervised learning approaches show a decreasing standard deviation.
A one-way repeated ANOVA with $F(2, 20) = 41.88$, $p = 7.08 \cdot 10^{-8}$ shows that the different classification methods have a significant influence on the average F-score in 20-class text categorization. A post hoc test with Tukey-Kramer correction indicates that the *graph-based SSL* differs with $p = 1.57 \cdot 10^{-5}$ significantly from *bag-of-words SSL*, as well as from *bag-of-words SL* ($p = 1.24 \cdot 10^{-5}$). Between *bag-of-words SSL* and *bag-of-words SL* no significant difference is found.

All in all, *graph-based SSL* outperforms the other approaches in 20-class text categorization. However, dependent on the number of labeled documents *bag-of-words SSL* or *bag-of-words SL* yield equivalent F-scores.
With respect to the standard deviations, *graph-based SSL* seems to be more stable than *bag-of-words SSL*.

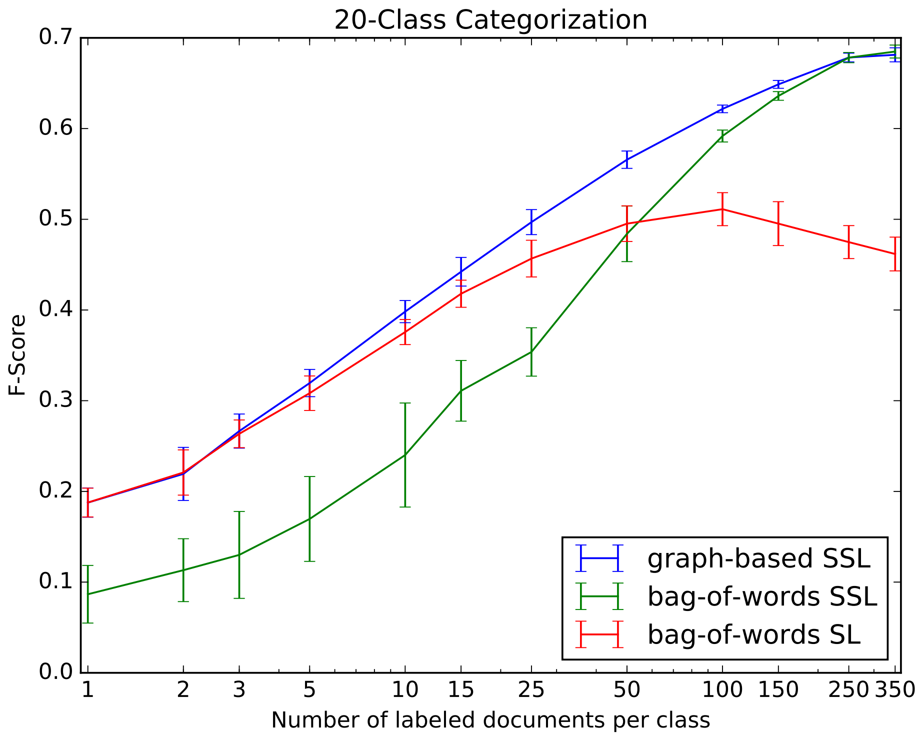


Figure 8: Classifier comparison for 20-class text categorization with a varying number of labeled documents. Displayed are the mean F-scores averaged over 10 independent runs, in which the labeled documents are selected randomly for each class. The error bars show the corresponding standard deviation.

## 5.3 Additional Insights

Before discussing in detail the results of the two main experiments we will have a quick look at additional insights we gain from the graph-based semi-supervised learning approach.
First of all, we evaluate the transition probabilities. We assume that words that appear often in a particular sequence have a high normalized transition probability. To evaluate whether the proposed graph representation is able to capture such frequent bigrams, we state all word pairs with a normalized transition probability higher than 0.9 in Table 7.

The word pairs on the left side of the table are familiar terms including cities, countries, commonly used expressions or proper names. In contrast, the word pairs on the right side might need further explanation.
A web search reveals the quote *'Skepticism is the chastity of the intellect, and it is shameful to surrender it too soon or to the first comer [...]'* from George Santayana which can be linked to the first four word pairs. Presumably, these keywords are not very frequent in the rest of the document collection and therefore, repeating this quote a few times, for example

Table 7: Word pairs in the graph representation with a normalized transition probability of at least 0.9.

| Word 1 | Word 2 | Word 1 | Word 2 |
|---|---|---|---|
| new | york | skepticism | chastity |
| new | zealand | chastity | intellect |
| san | diego | intellect | geb@cadredslpittedu |
| los | angeles | geb@cadredslpittedu | shameful |
| star | trek | serdar | argic |
| vice | versa | bank | n3jxp |
| radio | shack | go | quaker |

in the foot note of a post, leads to very high transition probabilities.
Other word pairs can be explained in a similar way. *bank n3jxp* is related to a user of the newsgroup channel and *serdar argic* is the pseudonym of one of the first newsgroup spam bots that appeared in 1994 (Serdar Argic, 2017). The expression *go quakers* is obviously related to the *Quaker* basketball and football team of the University of Pennsylvania.

All in all, these word pairs with a very high transition probability indicate that the graph representation is able to capture relevant word sequences. See Section 6 for a more in-depth discussion about how high transition probabilities emerge and their influence on the semi-supervised learning.

In semi-supervised learning the labeled nodes are propagated through the graph until all nodes are assigned to a label. As we use documents and words as nodes in the graph representation, these words will also be assigned to a label. Besides the label also its corresponding probability is given.
This allows us to inspect the words that are typical for a certain class. Figure 9 displays the word clouds for four example categories, *talk.politics.guns, soc.religion.christian, sci.space* and *sci.crypt*. The more space a word takes in the cloud, the higher is its probability to belong to the specific class.
In general, the most prominent words in a word cloud are highly associated with their respective class. However, an in-depth analysis of the word clouds goes beyond the purpose of this report. But we like to underline this feature of semi-supervised learning and discuss possible applications of feature labeling in Section 6.

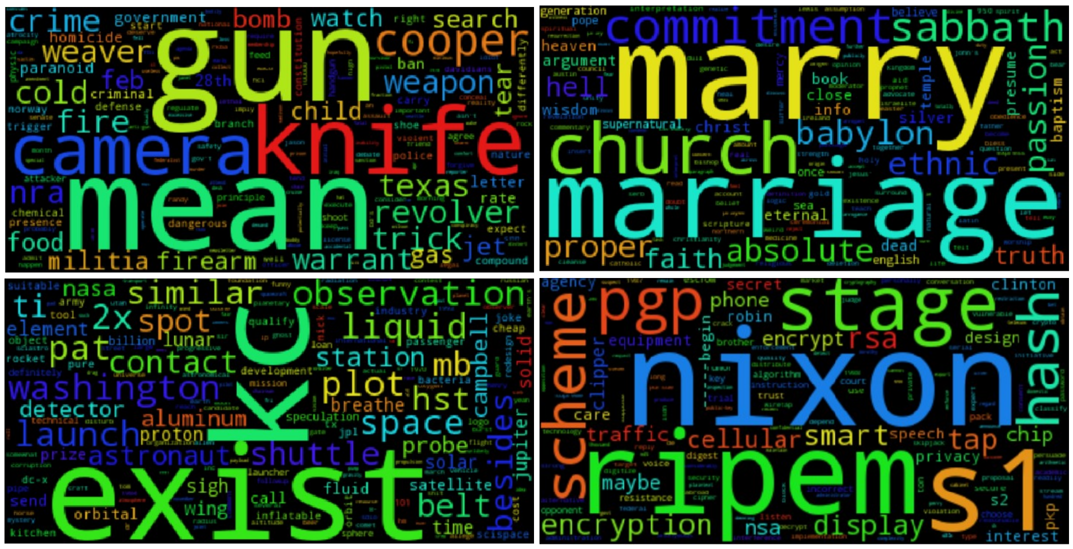


Figure 9: Example word clouds that correspond to the classes *talk.politics.guns, soc.religion.christian, sci.space* and *sci.crypt*. The bigger a word is displayed the higher is gits probability to belong to the respective class.

## 6 Discussion

In this section we review the obtained results, make suggestions for improvements, discuss the computational complexity of graph-based semi-supervised learning and propose fields of applications that benefit most from this approach.

### 6.1 Discussion of Results

For both experiments, the results of 4- and 20-class text categorization are quite consistent. The performance of semi-supervised learning without an RBF kernel is impractically low. Consequently, the proposed graph representation does not transmit sufficient information about node relations necessary for label propagation. However, applying an RBF kernel to the weight matrix conveys this information. As the $\gamma$ parameter of the RBF kernel has a significant influence on the matrix and the corresponding text categorization performance (Wang and Zhang, 2008), a cross-validated parameter optimization is required in order to ensure good results while avoiding over-fitting.

A main finding of this report is that graph-based semi-supervised learning - by including feature-feature relation - outperforms bag-of-words based semi-supervised learning. While both approaches use *Tf-Idf* weighting, in the graph representation also document-document and feature-feature relations are stated. As the identity matrix is used to represent document-document relations, the feature-feature relations are responsible for the improvement in text categorization.
However, as the different feature-feature relations, *transition matrix, trigrams, context sim-*

*ilarity* or their *average*, yield very similar results, it seems irrelevant which of them is used. Therefore, our first hypothesis can only be confirmed half way: Even though word relations improves semi-supervised learning, context similarity does not show a significant advantage compared to syntactic feature relations.

A possible explanation for this that the document-feature relations are represented twice in the weight matrix. Thus, adding feature-feature relations has an effect on the classification performance, but small changes in the **FF** matrix do not lead to significant differences.
Moreover, the documents are quite short, especially after removing stopwords. This leads to relatively high *Tf-Idf* weights compared to the feature-feature relations in the **FF** matrix. This also explains the big influence of sentence length on the performance of semi-supervised learning: In a document that consists of a few short sentences, the probability of a word to connect to a *\$Start\$* or *\$End\$* feature is higher than its probability to transition to any other word in the entire vocabulary.

Therefore, further research that investigates a way of normalizing graph features for label propagation is needed. Moreover, optimizing the feature weights dependent on the specific text mining task, could lead to significant improvements in the performance of graph-based semi-supervised learning.

In the second experiment we compare graph-based semi-supervised learning with bag-of-words based supervised and semi-supervised learning. By varying the number of labeled documents the effectiveness of the different classifiers is compared.
Graph-based semi-supervised learning constantly yields the highest F-scores for both tasks, 4- and 20-class text categorization. However, dependent on the number of labeled documents the other two approaches have a similar performance.
In contrast to our hypothesis, the supervised learning approach achieves, with a very low number of training documents per class, the same results as graph-based semi-supervised learning. It seems that the usage of unlabeled documents does not improve text categorization.

This is either due to the dataset itself, for example when the structure of the unlabeled documents does not relate to the respective classes, or the document representation is not able to transmit this information to a particular semi-supervised learning approach. Nigam et al. (2006) demonstrates that for the 20-Newsgroup dataset unlabeled documents increase categorization accuracy when using generative semi-supervised learning with Expectation-Maximization. Also, (Su et al., 2011) yield comparable results and further shows that increasing the number of unlabeled data from 1000 to 10000 documents improves performance, while adding further unlabeled documents does not result in a significantly better accuracy.
Thus, in contrast to our findings, semi-supervised learning is advantageous for the categorization of the 20-Newsgroup dataset. However, generative and graph-based semi-supervised learning are very different approaches, what makes it difficult to directly compare the results. While, for generative mixture models it is guaranteed that unlabeled data improves accuracy, provided that the assumption that each mixture component corresponds to the documents of a specific class is met (Zhu, 2005), no such claim is made for graph-based methods.

Nevertheless, in combination with label propagation the proposed graph representation is not able to exploit the full potential of unlabeled documents. In the following possible reasons are discussed.

To make graph-based semi-supervised learning efficient, it is essential that similar documents are connected with high edge weights such that labels can propagate easily. As we only establish an indirect connection between documents by relating them based on their words, label propagation becomes more difficult. Although some words might be good indicators for a certain class, labeling them might block the path to a different document class using the words.
Also, the high number of nodes in the graph representation, which emerges from using documents as well as feature as nodes, yields a higher level of complexity for the identification of similar documents.

Another possible problem might be that by using sentence structure as a feature for graph representation, the resulting weight matrix is asymmetric. For instance, the probability of *the* preceding *house* is higher than the probability of *house* preceding *the*, leading to a bidirectional connection with different edge weights. Although Table 7 shows that these relations are able to capture relevant n-grams, the transition matrix is highly influenced by the word frequency and whether in- or out-degree are used for normalization. Using the in-degree the transition probability indicates the likelihood that feature $f_i$ precedes feature $f_j$. Thus, words that have a low total frequency, such as *york, angeles, chastity*, and at the same time appear often after certain words, like *new, los, skepticism,..* will result in a high transition probability. Using the out-degree for normalization yields the probability of transitioning from feature $f_i$ to $f_j$. Here, word pairs with seldom words at beginning, such as *york state, kansas city*, have a high transition probability. Therefore, the computation of syntactic feature introduces a certain degree of ambiguity. This can be resolved by using a symmetric measure for similarity such as word co-occurrence within a sentence.

Despite not using the full potential of unlabeled documents, the graph-based semi-supervised approach yields, as expected, better results than bag-of-words based semi-supervised learning. Thus, in the next section we discuss its potential and different fields of applications.

## 6.2 Potential of graph-based Semi-Supervised Learning

As illustrated in section 5.3 semi-supervised learning offers the possibility to get a better understanding of the collection of documents. Transition probabilities indicate frequent word sequences without a manual definition of the n-gram length. Furthermore, words with a high context similarity give insights into the overall content of the documents and show how well word families are represented.

As we include word nodes in the graph representation, with label propagation they are also assigned to a certain class. This does not only contribute to the understanding of the decision making process, but also offers new application possibilities. Word clouds (see Figure 9) illustrate the relevancy of terms in a specific class and indicate possible subcategories.
The assigned feature class can also be used to infer information for a different categorization task. For instance, feature labeling in sentiment analysis indicates which words are characteristic for expressing a certain sentiment. This information can be used for a similar task

on a dataset with no initial labels. Kannan et al. (2016) use this idea to group short e-mail replies into semantic clusters containing different ways of expressing a similar objective, like agreeing on an appointment or saying thanks.
Another field of application for graph-based semi-supervised learning and the idea of exploiting labeled features offers network analysis. For instance, voter targeting aims to identify the key properties, like education, gender and age, of people voting for a certain party or being yet undecided.

The main advantage of graph-based semi-supervised learning is its flexibility to adapt to a specific text mining task by selecting appropriate graph properties and adapting their influence respectively. Although we focus on different document-feature and feature-feature relations, the similarity of documents is an important property that can contribute valuable information for text categorization. It can be assessed by including information about the author, journal, year of publication, etc.
Besides the discussed feature relations, other graph properties are conceivable, such as word co-occurrence in a sentence, semantic similarity based on word embeddings like *word2vec*, etc.

However, this process of feature engineering which includes the selection of graph properties and their weighting, is time-consuming and requires the manual effort to understand the dataset and to know which features are appropriate for a given task.

## 6.3 Computational Complexity

The complexity of a problem is related to the resources, like time and space, required to compute a solution as a function of its input size (Ruohonen, 2013). The proposed graph representation of a document collection requires iterating through all documents and their corresponding words. Every word and relation is checked for existence and if necessary modified or created. This leads to a time complexity linear to the number of documents $n$ plus the number of words $m$ $\mathcal{O}(n+m)$.
The advantage of graph representations is that properties based on neighboring relations, e.g. all words that appear after a certain feature, are obtained by traversing through the graph which is with $\mathcal{O}(p)$ proportional to the total number of neighboring nodes $p$.
However, for the purpose of label propagation a matrix that contains the similarity of each node to all others, has to be constructed. As kernel operations, like cosine similarity or the RBF kernel, corresponds to the dot-product of feature vectors (Zelenko et al., 2003) are able to do so. For a $n+m$ dimensional feature space a kernel function requires the computation of a $n+m \times n+m$ feature matrix and its storage (Cesa-Bianchi et al., 2015).
Semi-supervised learning scales poorly with the size of the data (Liu et al., 2012) as it includes expensive matrix modulations such as the computation of the matrix inverse which is $O((n+m)^3)$ in the worst case (Zhu et al., 2005). However, iterative approaches such as label propagation can be reduced to a complexity of $O(n+m)$ per iteration when sparse matrices are used (Zhu et al., 2005). For graph construction (see Jain et al. (2013)), modulation and semi-supervised learning (see Ravi and Diao (2015)) also more efficient algorithms and approximations exist. We do not further discuss these methods, as the focus of this paper is not on the reduction of the computational complexity of graph-based semi-supervised learning which is a research field on its own.

All in all, scalability of graph-based semi-supervised learning is limited, especially when documents as well as features are represented in the graph. Therefore, graph-based semi-supervised learning is in particular interesting for categorizing smaller datasets.

# 7 Conclusion

In this thesis, we show advantages and disadvantages of graph-based semi-supervised learning compared to bag-of-words based supervised and semi-supervised learning. While adding feature-feature relations significantly improves text categorization performance, it does not make a difference whether syntactic or semantic relations are used. Especially for the categorization of short texts where bag-of-words based approaches often suffer from sparseness or the usage of synonyms, graph-based semi-supervised learning can lead to a better performance.

Although the proposed approach does not exploit the full potential of including unlabeled documents in semi-supervised learning, its possibility to make use of the feature labeling brings numerous advantages. Another large asset of graph-based representations is their flexibility to include different node types and relations such that the features used for semi-supervised learning can be perfectly adapted to a specific text mining task.

A next step is to include document-document relations to establish similarities based on information about the author, journal or publication date as well as user interaction data such as the number of clicks, upvotes and queries, which is especially relevant for web page classification. However, further research is required in order to weight the graph properties and benefit from including unlabeled documents.

## References

Maria-Florina Balcan, Avrim Blum, Patrick Pakyan Choi, John D Lafferty, Brian Pantano, Mugizi Robert Rwebangira, and Xiaojin Zhu. Person identification in webcam images: An application of semi-supervised learning. 2005.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 11 label propagation and quadratic criterion. 2006.

Esteban Castillo, Ofelia Cervantes, Darnes Vilarino, David Báez, and Alfredo Sánchez. Udlap: sentiment analysis using a graph based representation. *SemEval-2015*, page 556, 2015.

Nicolo Cesa-Bianchi, Yishay Mansour, and Ohad Shamir. On the complexity of learning with kernels. In *COLT*, pages 297–325, 2015.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20 (3):542–542, 2009.

George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, 2003.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

Nilesh Jain, Guangdeng Liao, and Theodore L Willke. Graphbuilder: scalable graph etl framework. In *First International Workshop on Graph Data Management Experiences and Systems*, page 4. ACM, 2013.

Chuntao Jiang, Frans Coenen, Robert Sanderson, and Michele Zito. Text classification using graph mining-based feature extraction. *Knowledge-Based Systems*, 23(4):302–308, 2010.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, László Lukács, Marina Ganea, Peter Young, et al. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, volume 36, pages 495–503, 2016.

Wei Liu, Jun Wang, and Shih-Fu Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

William Lyon. Natural language processing with neo4j - mining paradigmatic word associations, 2015. URL `http://www.lyonwj.com/2015/06/16/nlp-with-neo4j/`. [accessed January 30, 2017].

Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

Kamal Nigam, Andrew McCallum, and Tom Mitchell. Semi-supervised text classification using em. *Semi-Supervised Learning*, pages 33–56, 2006.

Nihir Patel and Jason T. L. Wang. Semi-supervised prediction of gene regulatory networks using machine learning algorithms. *Journal of Biosciences*, 40(4):731–740, 2015.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Remus Radu Raluca Tanase. The mathematics of web search - lecture 2: Directed graphs - transition, 2009. URL http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture2/lecture2.html. [accessed March 8, 2017].

Sujith Ravi and Qiming Diao. Large scale distributed semi-supervised learning using streaming approximation. *arXiv preprint arXiv:1512.01752*, 2015.

Jason Rennie. 20 newsgroup, 2008. URL http://qwone.com/~jason/20Newsgroups/. [accessed February 5, 2017].

François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. Text categorization as a graph classification problem. In *ACL*, volume 15, page 107, 2015.

Keijo Ruohonen. Graph theory. 2013.

Adam Schenker. Graph-theoretic techniques for web content mining. 2003.

Serdar Argic. Serdar Argic — Wikipedia, the free encyclopedia, 2017. URL https://en.wikipedia.org/w/index.php?title=Serdar_Argic&oldid=758565385. [accessed March 8, 2017].

SS Sonawane and PA Kulkarni. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19), 2014.

Jiang Su, Jelber S Shirab, and Stan Matwin. Large scale text classification using semi-supervised multinomial naive bayes. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 97–104, 2011.

Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

Suzan Verberne, Maya Sappelli, Djoerd Hiemstra, and Wessel Kraaij. Evaluation and analysis of term scoring methods for term extraction. *Information Retrieval Journal*, 19(5): 510–545, 2016.

Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):55–67, 2008.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328, 2003.

Xiaojin Zhu. Semi-supervised learning literature survey. 2005.

Xiaojin Zhu. Semi-supervised learning tutorial. 2007.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.

Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs.* Carnegie Mellon University, language technologies institute, school of computer science, 2005.