University of Potsdam
Hasso Plattner Institute
Information Systems Group

Machine-learning-assisted Corpus Exploration and Visualisation

Dissertation

zur Erlangung des akademischen Grades
"Doktor der Naturwissenschaften"

(Dr. rer. nat.)

in der Wissenschaftsdisziplin
"Informationssysteme"

eingereicht an der
Digital Engineering Fakultät
der Universität Potsdam

von
Tim Repke

Potsdam, 25. November 2021

Gutachter

Prof. Dr. Felix Naumann
Hasso-Plattner-Institut, Universitat Potsdam

Prof. Dr. Michael Gertz
Universitat Heidelberg

Prof. Dr. Robert Jaschke
Humboldt-Universitat zu Berlin

Published online on the
Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-56263
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-562636

Zusammenfassung

Der Großteil unseres Wissens steckt in Textsammlungen, wie etwa Korpora von Büchern, Forschungsartikeln, Nachrichten, sowie Geschäftsunterlagen. Sie bieten somit eine wertvolle Grundlage um neue Erkennisse zu gewinnen oder relevante Informationen zu finden, allerdings sind manuelle Recherchen aufgrund stetig wachsender Datenmengen schier unmöglich. Dank der Digitalisierung können Suchmaschinen Recherchen erheblich unterstützten. Sie bieten jedoch lediglich eine selektive Sicht auf die darunterliegenden Daten und erfordern ein gewisses Vorwissen um aussagekräftige Anfragen zu stellen und die Ergebnisse richtig einzuordnen. Die Fortschritte im Bereich des maschinellen Lernens eröffnen völlig neue Möglichkeiten zur Interaktion mit Daten. Anstatt zahllose Geschäftsdokumente von Hand zu sichten, können Journalisten und Ermittler beispielsweise Techniken aus der Computerlinguistik einsetzen um automatisch Personen oder Orte im Text erkennen. Ein daraus gebildeter sogenannter Knowledge Graph kann Suchmaschinen deutlich verbessern, allerdings ist die Fülle an Informationen weiterhin überwältigend. Eine Übersicht eines gesamten Datensatzes, ähnlich einer geographischen Landkarte, ermöglicht innovative Interaktionsparadigmen und ermöglicht es Nutzern zu erkennen, wie sich bestimmte Informationen in Kontext des Gesamtbilds einfügen.

In dieser Arbeit werden Algorithmen entwickelt um heterogene Daten vorzuverarbeiten und sie auf zweidimensionalen kartenähnlichen Ansichten zu verorten. Traditionell werden zur Verortung hochdimensionale semantische Vektorrepräsentationen der Daten verwendet, die anschließend mit Dimensionsreduktionsalgorithmen auf eine zweidimensionale Ebene projiziert werden. Wir fokussieren uns auf die Visualisierung von Textkorpora und gehen dabei über die Projektion der reinen inhärenten semantischen Struktur hinaus. Hierzu wurden drei Ansätze zur Dimensionsreduktion entwickelt, die zusätzliche Informationen bei der Berechnung der Positionen einbeziehen: (1) Dimensionsreduktion mit mehren Kriterien, bei der sowohl semantische Informationen, als auch inhärente Netzwerkinformationen, die aus den zugrundeliegenden Daten abgeleitet werden, zur Positionsberechnung verwendet werden; (2) Analyse des Einflusses von Initialisierungsstrategien für verschiedene Dimensionsreduktionsalgorithmen, um eine zeitlich kohärente Serie an Projektionen zu erzeugen um Korpora abzubilden, welche im Laufe der Zeit wachsen; (3) Anpassung bereits vorhandener Projektionen auf der Basis einzelner, händisch verschobener Datenpunkte. Diese Arbeit beschreibt darüber hinaus Prototypen für Benutzeroberflächen, die zur Demonstration der beschriebenen Technologien entwickelt wurden.

Abstract

Text collections, such as corpora of books, research articles, news, or business documents are an important resource for knowledge discovery. Exploring large document collections by hand is a cumbersome but necessary task to gain new insights and find relevant information. Our digitised society allows us to utilise algorithms to support the information seeking process, for example with the help of retrieval or recommender systems. However, these systems only provide selective views of the data and require some prior knowledge to issue meaningful queries and asses a system's response. The advancements of machine learning allow us to reduce this gap and better assist the information seeking process. For example, instead of sighting countless business documents by hand, journalists and investigators can employ natural language processing techniques, such as named entity recognition. Although this greatly improves the capabilities of a data exploration platform, the wealth of information is still overwhelming. An overview of the entirety of a dataset in the form of a two-dimensional map-like visualisation may help to circumvent this issue. Such overviews enable novel interaction paradigms for users, which are similar to the exploration of digital geographical maps. In particular, they can provide valuable context by indicating how a piece of information fits into the bigger picture.

This thesis proposes algorithms that appropriately pre-process heterogeneous documents and compute the layout for datasets of all kinds. Traditionally, given high-dimensional semantic representations of the data, so-called dimensionality reduction algorithms are used to compute a layout of the data on a two-dimensional canvas. In this thesis, we focus on text corpora and go beyond only projecting the inherent semantic structure itself. Therefore, we propose three dimensionality reduction approaches that incorporate additional information into the layout process: (1) a multi-objective dimensionality reduction algorithm to jointly visualise semantic information with inherent network information derived from the underlying data; (2) a comparison of initialisation strategies for different dimensionality reduction algorithms to generate a series of layouts for corpora that grow and evolve over time; (3) and an algorithm that updates existing layouts by incorporating user feedback provided by pointwise drag-and-drop edits. This thesis also contains system prototypes to demonstrate the proposed technologies, including pre-processing and layout of the data and presentation in interactive user interfaces.

Acknowledgements

The endeavour of writing this thesis would not have been possible without the help of many kind people. First and foremost, I would like to express my deepest appreciation for all the support my supervisors Prof. Felix Naumann and Prof. Ralf Krestel offered my over the years. The time you have taken for me in regular discussions and giving valuable feedback has been invaluable and helped me to navigate through all the personal and academic challenges to make this achievement possible. Further, I would like to thank my external supervisor Prof. Ulf Leser for all his constructive feedback to help me identify key objectives and form long-term perspectives. Especially in the beginning of this endeavour, the insights into real-world applications and problems provided by our project partners Dirk Thomas and Dr. Oliver Maspfuhl inspired many of the projects I worked.

I am also extremely grateful that I had the pleasure of working in a research group with so many talented and friendly people. Especially Michael Loster, Julian Risch, Alejandro Sierra Múnera, and Nitisha Jain have given me valuable feedback and support over the years; it was a lot of fun having you by my side. During my time at the HPI, I was very fortunate to meet many bright students. I would like to thank my student assistants Benjamin Feldmann, Jan Ehmüller, Lasse Kohlmeyer, Ben Hurdelhey, Olena (Alyona) Vyshnevaska and Paul Wullenweber for their contributions to my research. It has also been a great experience co-teach lectures, seminars, bachelor projects, and master projects that allowed me to discover new research areas and work on exciting projects. A special thank you to the master thesis students Thomas Kellermeier, Noel Danz, and Robert Schwanhold who later also became co-authors. I would also like to thank Prof. Peer Trilcke and Henny Sluyter-Gäthje from the Theodor Fontane Archive and Dennis Mischke for their support in teaching and their interdisciplinary expertise. The same applies to the research group of Prof. Jan Minx with Dr. Max Callaghan and Dr. Finn Müller-Hansen, who also sparked my excitement in new challenges that I will follow beyond this thesis.

Last but not least, I would like to thank my family and friends for giving me the energy and confidence to continuously working on this thesis, but also helping me to take my mind off things when needed.

I would like to dedicate this thesis to my late father, to whom I will be eternally grateful for leading me to where I am today and always encouraging me to pursue goals I never imagined possible.

Contents

Chapter 1

INTRODUCTION

In today's technologically advanced society, data plays a significant role in almost all aspects of life. Our world's knowledge is captured in books and encyclopedias, most of our communication happens electronically, current events are documented by online news sources, and details about scientific or technological advancements are published digitally. In order to utilise this overwhelming wealth of information, we need tools to manage and access this data effectively. Some tools, such as information retrieval or recommender systems, have already become essential commodities for finding relevant information in everyday situations.

Although search engines are widely accessible for seeking information in local databases or even the entire internet, some use cases require specifically tailored solutions. For example, when internal auditors in companies, law enforcement agencies, or journalists obtain a large corpus of internal documents, e.g. released by whistleblowers, they need to retroactively analyse the data and look for hidden information. Such a corpus of internal documents and emails provides a record of all relevant decisions and discussions that drive the daily business. It is the job of auditors, lawyers, or law enforcement find evidence that proofs or disproofs a particular accusation. Journalists on the other hand may follow a more open ended approach in their analysis.

Consider the largest leaks to date, the so-called *Panama Papers*[1], *Paradise Papers*[2], and *Pandora Papers*[3], where several terabytes of data were leaked from offshore investment and law firms. With the help of this data, journalists exposed large-scale fraud and tax evasion schemes that implicated several politicians, senior public officials, and wealthy individuals of illegal or unethical behaviour. To uncover these insights, it took the joint effort of hundreds of journalists around the world more than a year for each leak. However, due to the sheer size of these datasets, manually reading and connecting potentially interesting clues is simply not feasible. To this end, they relied on the support of data mining, namely

[1] Published 2016, 11.5M documents (`https://www.icij.org/investigations/panama-papers/`)
[2] Published 2017, 13.4M documents (`https://www.icij.org/investigations/paradise-papers/`)
[3] Published 2021, 11.9M documents (`https://www.icij.org/investigations/pandora-papers/`)

natural language processing techniques, to automatically preprocess the data and extract named entities, such as organisations, persons, locations, and dates. This shared platform[4] allowed them to build a structured database that they could query and quickly discover possibly interesting connections and new leads. It also allowed them to share their progress for others to follow up on. Throughout the entire process, data provenance is preserved to be able to verify findings based on the original data.

1.1. Problem Descriptions

Several challenges arise when automatically processing large textual corpora to make them accessible for exploration. Especially the sheer amount of information can be overwhelming, especially when users are not yet familiar with a dataset. Traditional information retrieval systems can only present a limited view of the data with the results returned for a given query, even if they are supplemented with information from knowledge graphs. Even formulating the right query in an open-ended information seeking process requires some prior knowledge about the underlying dataset. Machine learning has become an invaluable asset for exploring corpora in the context of investigations. It can be employed in cleaning noisy data, mining salient information, classifying and ranking it for the domain-specific needs, and preparing it for visualisations used in innovative and interactive exploration interfaces.

Preprocessing the Data. Real-world document corpora are inherently heterogeneous and often too noisy for downstream tasks to perform well. For example, text mining methods like named entity extraction work best if they receive well-formed sentences. However, the actual content in documents is often hidden within semi-structured formatting templates. In order to clean the data to improve following processing steps, letterheads, formatting, and other structural artefacts have to be identified. In this thesis, we focus on recovering the salient structure of free-text email message chains.

Mapping a Text Corpus in Two Dimensions. People have been drawing geographic maps for millennia as a reference for navigation and orientation, but it also improved awareness of how elements of the map relate to one another in a global context. Visualising non-geographic data, such as text collections in a similar manner has many benefits. First, it allows us to get a global overview of all aspects of the dataset and quickly identify semantically similar clusters. Second, given a specific document a map can provide context,

[4] ALEPH (`https://alephdata.org`) has become the foundation for many investigative journalists to cope with large sets of documents.

similar to how the GPS position marker on the phone does. Third, information retrieval systems could display search results on the map to highlight related areas. There are numerous other innovative interaction patterns to be developed by transferring concepts from geoinformation systems to analogies in the domain of text visualisation. Although users have to first familiarise themselves with newly drawn map of a corpus, it is much faster than reading hundreds of documents to get even a basic understanding. It can serve as a mental map that can support many aspects of data exploration and information seeking. In order to leverage the full potential of a two-dimensional visualisation, the placement of items from the dataset, the so-called layout, is crucial. In this thesis, we cover different layout algorithms for heterogeneous datasets.

Incorporating Additional Information into Layouts. When computing the layout for a text corpus, documents should be positioned in such a way, that they are near one another if they are semantically similar. However, some domain specific applications may have additional requirements beyond preserving semantic similarity alone. These could include aspects like document type, categories known ahead of time, or other meta-data. In this thesis, we focus specifically on jointly visualising salient network structures and semantic information.

Accounting for Temporal Change. Algorithms for laying out text corpora or other datasets usually consider the data to be static. However, many real-world datasets grow and evolve over time. In use cases where accounting for the temporal aspect of the data is necessary, it could be illustrated directly in the rendering of the layout by distinctive shading or interactively hiding documents outside a selected interval. These approaches will only work in cases where the entire corpus is available ahead of time. In this thesis, we focus on the case where layouts have to be updated as new data arrives. This is particularly challenging, since the updates have to be consistent in order to preserve the general layout that users are already familiar with.

Editing Existing Layouts. Datasets can be interpreted in many different ways. Typically, the use case in which the data is analysed and explored defines the aspects that are most relevant. For example, an economist would structure a set of news articles differently than a physicist, as both have different priorities, interests, and expectations. Also in the context of investigative journalism, a team will have a better understanding as their research evolves. In this thesis, we propose methods to augment the process of updating a existing layouts.

1.2. Outline of this Thesis

In the following, we provide an overview of this thesis, including contributions made by the respective publications related to each chapter. This thesis is structured in two main parts. In the first part (Chapters 2–5), we cover our proposed dimensionality reduction algorithms for creating map-like visualisations of data. In the second part (Chapters 6–7), we focus on domain-specific applications for corpus exploration.

Chapter 2 – An Introduction to Map-like Data Visualisation. In this chapter, we provide an overview of related work in the area of map-like data visualisation. We describe how data can be projected onto a two-dimensional canvas, the map of the data, starting from network representations or high-dimensional vector representations using graph drawing algorithms or dimensionality reduction. Therefore, we provide an overview of such algorithms, summarise how they can be used in practice, and highlight some applications on real world data.

Chapter 3 – Joint Visualisation of Text and Network Data. Corpora, such as academic publications or emails provide several aspects, mainly semantic information from the written text and network information about citations or senders and recipients. Map-like visualisations of such corpora typically consider only one of these aspects. Therefore, we proposed MODiR, a multi-objective dimensionality reduction algorithm that incorporates both aspects. This chapter is based on three publications, including a vision paper, a conference paper[5], and a demo paper.

- T. Repke and R. Krestel. Topic-aware network visualisation to explore large email corpora. In *International Workshop on Big Data Visual Exploration and Analytics (BigVis)*, Proceedings of the International Conference on Extending Database Technology (EDBT), pages 104–107. CEUR-WS.org, 2018

- T. Repke and R. Krestel. Visualising large document collections by jointly modeling text and network structure. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, pages 279–288. ACM Press, 2020. doi: 10.1145/3383583.3398524

- T. Repke and R. Krestel. Exploration interface for jointly visualised text and graph data. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 73–74. ACM Press, 2020. doi: 10.1145/3379336.3381470

[5] Received "The Best Student Paper Honourable Mention" at the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2020).

Chapter 4 – Robust Visualisation of Diachronic Text Collections. Some real-world applications of map-like visualisations are based on constantly changing or growing datasets. For example, when visualising the news landscape, the visualisation may need to be updated on a daily basis. In such an application, the general semantic layout should remain stable over time. To this end, we compared different strategies to produce robust visualisations while still allowing previously unseen topics to establish a new region in the layout.

- T. Repke and R. Krestel. Robust visualisation of dynamic text collections: Measuring and comparing dimensionality reduction algorithms. In *Proceedings of the Conference for Human Information Interaction and Retrieval (CHIIR)*, pages 255–259. ACM Press, 2021. doi: 10.1145/3406522.3446034

Chapter 5 – Computer-assisted Curation of Map-like Visualisations. Dimensionality reduction algorithms only focus on preserving pairwise similarities in the two-dimensional projection of the high-dimensional data. Oftentimes, this process is non-deterministic and there may be several valid projections. However, domain experts may disagree with the resulting layout and want to edit the layout to fit their mental map of the data. For larger datasets, it is infeasible to manually adjust the position of each item in the layout. To this end, we propose EDIMAP to augment the editing process in the last chapter of the first part. Since singular edits can be interpreted in several different ways, we introduce a taxonomy of user edit intents.

- T. Repke and R. Krestel. Interactive curation of semantic representations in digital libraries. In *Proceedings of the International Conference on Asian Digital Libraries (ICADL)*, Lecture Notes in Computer Science (LNCS), pages 1–10. Springer-Verlag, 2021. doi: 10.1007/978-3-030-91669-5_18

Chapter 6 – Segmenting Semi-structured Text in Emails for Exploration. In the previous chapters, we focused on algorithms to create the layout for map-like visualisations while assuming meaningful representations of the data to be given. In this chapter, we consider real-world email data and how to extract the actual written messages from the free text email body. With our state-of-the-art neural network-based approach QUAGGA, we are able to segment the email body by embedding and classifying each line. Our experiments show, that this approach is very robust regarding noise introduced by encoding errors, different languages, or broken formatting. We demonstrate this using two annotated datasets that we introduced as part of this work. Furthermore, we built a system that extracts structured information from raw email data and presents it in an interactive inter-

face. This BEACON[6] system can support investigations by auditors or journalists and was actually used during an internal investigation at a large German bank.

- T. Repke and R. Krestel. Bringing back structure to free text email conversations with recurrent neural networks. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 114–126. Springer-Verlag, 2018. doi: 10.1007/978-3-319-76941-7_9
- T. Repke, R. Krestel, J. Edding, M. Hartmann, J. Hering, D. Kipping, H. Schmidt, N. Scordialo, and A. Zenner. Beacon in the dark: A system for interactive exploration of large email corpora. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 1871–1874. ACM Press, 2018. doi: 10.1145/3269206.3269231

Chapter 7 – Exploration of Online News Comments. Nowadays, comment sections of online news platforms can be overwhelming by the sheer amount of comments. As a result, fewer in-depth discussions emerge. To foster more interactive and engaging discussions, users may benefit from an overview of prior arguments and active topics. With our COMEX platform, we introduce a novel interface to explore reader comments which are represented in a graph of topical similarities and meta-data.

- J. Risch, T. Repke, L. Kohlmeyer, and R. Krestel. ComEx: Comment exploration on online news platforms. In *Joint Proceedings of the Workshops co-located with Conference on Intelligent User Interfaces (IUI)*, pages 1–7. CEUR-WS.org, 2021

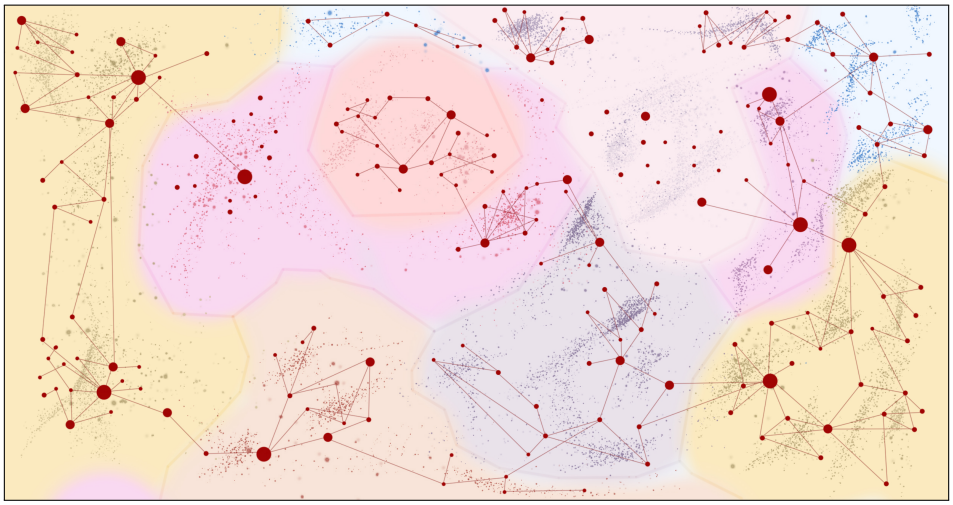
Chapter 8 – Conclusion. The final chapter concludes this thesis and provides a summary of all findings, as well as an outlook of possible research directions for future work.

While working on this thesis, I contributed to and worked on several other publications, which are, however, beyond the scope of this thesis. In particular, there are three workshop papers on the extraction of company relationships [95], rating their relevance [169], and discuss the challenges of maintaining a knowledge graph of financial entities [118]. We also published an introductory book chapter on the extraction and representation of financial entities, including their visualisation [166]. Apart from research on company networks, I also worked on graph-based models for interpretable models of linguistic change of word meanings over time, which resulted in one workshop– and one conference paper [52, 187], as well as embedding of novels [103]. Lastly, we published an article about our experiences hosting an online course for over 10,000 participants [4].

[6] Was mentioned as the runner-up for the "Best Demo Award" at the International Conference on Information and Knowledge Management (CIKM 2018).

Part I.

Map-like Data Visualisation



Chapter 2

An Introduction to Map-like Data Visualisation

Comprehending complex information and large amounts of data can be challenging. Even before computers were around to cope with the ever growing wealth of information, people developed graphical representations to summarise data or convey relations. Data visualisation is often associated with bar charts or line plots for statistical data analyses and mathematical functions. However, information visualisation also offers more advanced methods to condense heterogeneous and complex data, such as text or networks [72]. The high-level overview a visualisation provides can help to uncover otherwise hidden patterns, gain new insights from the underlying dataset, and better understand inherent interrelations [175]. For example, the image on the previous page illustrates a manually drawn vision of how a large corpus could be shown as an abstracted two-dimensional map. Each of the grey dots, which looked at from a birds-eye view appear like shaded areas, would depict a document from the corpus. The coloured regions would represent categorical information. The more prominent red dots and their relationship edges could be based on additional information derived from the dataset itself. Users reading this map would immediately be able to see that documents of the corpus are semantically clustered and how they relate. As part of an interactive exploration platform, they would be able to investigate the different neighbourhoods, identify areas relevant to them, and discover new insights.

In this first part of the thesis, we mainly focus on visualising textual corpora. This chapter in particular provides an introduction and overview of the related work on map-like data visualisation. In the first three sections, we outline the general visualisation pipeline and existing work on visualising networks and text. The remaining sections cover relevant technologies used in a visualisation pipeline to produce two-dimensional layouts of document collections and high-dimensional data, namely semantic representations of text, so called embeddings, dimensionality reduction, and domain-specific applications.

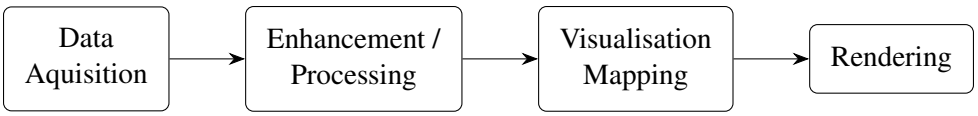


Figure 2.1.: Abstracted overview of the data visualisation pipeline

2.1. The Data Visualisation Pipeline

Felger et al. [59] introduced a reference model for visualisation systems. The model describes the general foundation for the process of transforming raw data into a displayable image as depicted in Figure 2.1. This model has been adapted to different types of scientific visualisations (volume visualisation and flow visualisation) as well as information visualisation [64, 72], which we focus on in this thesis.

The first step of the visualisation pipeline to to *acquire or simulate* the raw input data. In the context of this thesis, this would be the construction of a text corpus by querying libraries, scraping data from the web. We assume a collection of documents already exists and is ready to be loaded in the visualisation pipeline.

Second, the raw data needs to be *pre-processed and enhanced*, which is presumably the most important stage when working with heterogeneous data like text. Depending on the type of source, the actual content has to be extracted first, for example by stripping HTML tags and boilerplate, transforming PDFs to plain-text, or identifying passages of relevant text in semi-structured free-text documents. The extracted text may have to cleaned further by filtering and normalising tokens. Some use-cases could furthermore require additional enrichment of the data using text mining to extract entities, detect topics, or classify documents. The second part of this thesis will cover some domain-specific approaches to clean and enrich data. In this part of the thesis, we focus on creating map-like visualisations of a document collection. Therefore, the raw texts need to be transformed into high-dimensional vectors first. These numerical vectors have to faithfully represent the underlying text in such a way, that a metric to determine the semantic similarity between two texts can be defined. We will cover semantic representations of text later in this chapter.

The derived data is then ready to be *mapped to visualisation objects* in the third step of the reference model. According to Felger et al. [59], this includes the computation of artefacts like contours or histograms, as well as mappings to space, time, colour, or size. Our main focus lies on the latter, in that we aim to generate meaningful layouts of documents in a two-dimensional space, that best represent the inherent semantic and syntactic structures of the corpus. In particular, this incorporates the projection of the high-dimensional representations using dimensionality reduction. The following chapters in this part will cover novel approaches that integrate information beyond the texts themselves into that process.

Lastly, the *abstract visualisation objects* need to be *rendered* into a displayable image with the help of computer graphics algorithms for visibility calculation, shading, compositing, and animation. This step is however beyond the scope of this work. We assume that implementations in real-world applications provide the necessary tools to handle these tasks. We do however include small prototypes to demonstrate how the proposed methods could be employed in different use-cases.

2.2. Network Visualisation

In this thesis, we cover the machine-learning-assisted visualisation of large heterogeneous datasets for exploration with a particular focus on text corpora. Many real-world datasets of that kind also contain auxiliary information, for example inherent networks. These networks can be derived from the meta-data, such as hyperlinks between webpages, communication flow of emails, citations in scientific publications, or relationships between entities mentioned in the text itself. Formally, a network is mathematically defined as a graph with nodes connected by edges, typically without spatial information. The process of visualising a network is known as graph drawing.

Networks are traditionally visualised in so-called node-link diagrams. Thereby, each node is drawn as a dot or circle on a two-dimensional canvas, which are connected by straight lines based on the edges. In order to position the nodes on the canvas, force-directed layout algorithm is commonly used. This concept was first proposed by Fruchterman and Reingold [63] and simulates nodes as particles in a physical system, where attractive forces are applied to nodes connected by an edge, which are counter-balanced by repulsive forces between all nodes. Newer approaches optimise the computational complexity and include local metrics to better represent inherent structures as for example *ForceAtlas2* [85], which is the default network algorithm for the popular network visualisation tool *Gephi*. Eades et al. [50] later improved the scalability even further and while also better emphasising on a clear visual depiction of high-level structural patterns. As a network grows in size, node-link diagrams quickly become unreadable, as the visualisation is cluttered with edges. In order to reduce this visual noise, so-called edge-bundling could be applied. In this way, edges that are close and more or less parallel to one another are smoothly bent to form groups. This method has been shown to improve the readability in certain use cases [12]. Lhuillier et al. [113] wrote a survey on state-of-the-art edge-bundling approaches.

Besides the more traditional node-link diagrams, more artistic approaches for visualising networks use the metaphor of geographical maps by letting the topology reflect connectivity of densely connected social communities [146]. Other metaphors were also proposed, like that of growing “ContactTrees” to highlight how relationships form and change in a social network based on user interactions [181]. Although this reflects temporal aspects of

dynamic networks well, it focuses on one person as the root, thus an overview of the entire network is not possible. CactusTrees [43], on the other hand, represent hierarchical structures with the goal of untangling overlaid bundles of intersecting edges, making distant connections more apparent. As higher order dependencies may get lost in traditional visualisations, HoNVis [198] adds nodes to encode dependencies in chains of interactions.

Usually, a communication network has many nodes and overlapping connections already, so Yang et al. [210] rather focus on discovering overlapping cores to improve the identification of community boundaries to highlight global latent structures. Similarly, Gronemann and Jünger [69] use the metaphor of islands and hills to visualise clustered graphs, making densely connected communities clearly noticeable. But, the edges are bundled and follow valleys of the resulting topology, thus making relationships between other communities hard to follow. MapSets [51] in an algorithm that draws regions around clusters of nodes, such that the bounding shapes are contiguous and non-overlapping, but yet abstract. A similar concept was adapted more recently for temporally dynamic graphs [79]. Another approach to visualise networks at full scale is to aggregate nodes based on their spatial distribution and thereby allowing for a simple exploration with contour lines and heatmap overlays to emphasise latent structures as proposed by Hildenbrand et al. [76]. More modern approaches for graph drawing make use of the recent developments in deep learning. For example, in conjunction with dimensionality reduction, graph embeddings like LINE [196] allow scalability to millions of nodes. Wang et al. [203] propose a deep-learning-based graph drawing approach that can directly map the network structure. They use a set of existing layout examples to train a graph-LSTM-based model that capture their layout characteristics. The trained model can then be used used to generate graph drawings in a similar style for new networks.

The approaches mentioned above aim to incorporate the entire network in the visualisation. As Nguyen et al. [141] discuss, it can be very challenging to faithfully represent all the available information. Especially when analysing graphs on a very condensed level, for example by computing metrics and statistics for graphs, different graphs may share the same results [34]. In order to retain a similar minimalistic level of abstraction, Yoghourdjian et al. [213] developed icon-sized thumbnails to depict a high-level structure view of network for quick comparisons.

In this thesis, we aim to generate visualisations that can be used to explore large-scale datasets. Pienta et al. [152] wrote a concise summary of interaction patterns for sensemaking from networks visualised as node-link diagrams. For further reading on map-like network visualisation, we refer the survey by Beck et al. [17] for dynamic graphs and the survey by Gibson et al. [67] for static networks.

2.3. Text Visualisation

Text visualisation aims to visualise the content of a document or collection of documents to enable users to get a summarised overview and gain quick insights into topics, latent phrases, or trends. A great demonstration of the efficacy of combining text mining and visualisation is the data-driven text visualisation survey by Liu et al. [116]. Aside from summarising visualisations to highlight relationships between relevant publications, they also describe their visualisation pipeline. This pipeline includes scoping (filtering of related work) and coding (classification of concepts), as well as text mining methods to extract additional information that is used in the analysis of the field.

Probably the most popular method to visualise a summary of text are so-called word-clouds [117]. They display the most frequent terms of a collection in a very compact form factor. Although this might be useful for some applications, a significant amount of information is lost, for example the context these words appear. Several approaches enhance word-clouds, for example by vertically aligning multiple word-clouds to visualise temporal trends [111]. The selection of words to include in the visualisation is typically done by a frequency-based ranking of words. FinanVis [140] on the other hand extract named entities first and draw temporal word-clouds of words in close proximity to these entities. In this way, they are not only able to include additional context, but also temporal information of emerging co-occurrences. The positioning of a word in a traditional cloud typically conveys no meaning and is purely based on most efficient coverage of the available space. Scattertext [96], on the other hand, consciously places the most relevant words on a two-dimensional space based on document categories and other univariate attributes in a two-dimensional space for comparative analyses.

Apart from merely summarising a text collection, another application for text visualisation is to reveal hidden patterns in the data. When analysing a very specific domain for example, users may want to gain a better understanding of the different facets of their target word. The proposed FacetAtlas [28] allows for a word-level analysis of relevant aspects. On the other hand, TextDNA [194] focuses on uncovering corpus-level patterns with the help of so-called colour-fields. They use colours and positioning in a grid to visualise word rank and frequencies to expose patterns such as shifts in typographic conventions or cultural influences on word usage. Rule et al. [176] visualise correlation matrices of all State of the Union speeches of US-American presidents and were able to link the emerging patterns in their analysis to known historical shifts, suggesting that this way of visualising corpora can be a powerful tool to gain meaningful insights in use-cases without prior knowledge.

Topic models are a well-known text mining technology to discover structure in a collection of texts. They represent the individual documents as distributions over topics and topics as distributions over words — a concept which can be hard to comprehend. One way to

visualise a dataset with the help of a topic model is to place topics around the circumference of a circle and simulate documents as particles that are drawn towards the topics with their respective topic distribution. Riehmann et al. [171] used this principle and added additional features to interact with that space and represent documents as glyphs with "arms" of varying length to indicate how strong each topic influences each document. In doing so, the resulting scatter-plot for a large corpus may result in a very dense and unclear layout, so Chen et al. [36] developed an algorithm to reduce over-full visualisations by picking representative documents. A different approach is taken by Fortuna et al. [60], who do not show documents directly, but generate a heatmap of the populated canvas and overlay it with salient phrases at more densely populated areas from the underlying documents in that region. Fried and Kobourov [62] extend that concept by drawing clear lines between regions and colouring them. They also add edges between salient phrases based on co-occurrences in the texts. In order to incorporate the temporal aspect of the topics covered in transcripts of a debate, ConToVi [53] uses the same particle analogy described before. Hereby, a particle represent a members of the discussion, where the topic distribution changes over time, thus allowing the particles to float and leave trails. Others use stream-graphs, a form of stacked area charts, to visualise the frequencies of topics over time [42, 49]. These can also be enriched with word-clouds of derived time-sensitive keywords [207].

In this thesis, we use the analogy of a map to visualise the contents of documents by embedding them into a high dimensional semantic space and projecting it on a two-dimensional canvas. Cartograph [188] uses this analogy and implements a scalable pipeline and exploration interface. They pre-render information at different resolutions and use a tiling server, a concept commonly used for digital geographic maps, to allow for responsive interactions with very large datasets. Regions on the map are coloured based on underlying ontologies from a knowledge-base. In the following sections we provide a more detailed overview of the technologies required to create the layout for such a map.

2.4. Semantic Representations of Text

Earlier in this chapter, we introduced the reference model for visualisation pipelines. For our goal to create map-like visualisations of text corpora, we need to compute numerical vector representations in the data enhancement step, which can be used later by a dimensionality reduction algorithm to generate a layout of the data. These high-dimensional vectors have to faithfully represent the entire content of a text from the corpus. This allows us to semantically compare pairs of documents, by calculating the cosine similarity or euclidean distance which can then be replicated in the two-dimensional layout.

A very basic method to achieve that is with so-called bag-of-words vectors. Each dimension of a vector is associated with a word in the dictionary and, given a text, this dimension

dimension would be set to one if it contains that word and zero otherwise. This can be improved by weighting the vectors using tf-idfscores, which use the term frequency and document frequency to emphasise more representative words. However, since the dimensions are orthogonal, similar words and even flections of the same word appear as completely independent concepts.

In recent years, embeddings became more popular as they conserve semantic meaning in their vector representation. Mikolov et al. [131] introduced neural architectures to learn high-dimensional vector representations for words and later for paragraphs [109]. Similar methods are used to learn representations for nodes in a network based on either the structural neighbourhood [57] or additional heterogeneous information [32, 74]. Schlötterer et al. [185] attempted to learn joint representations of network structure and document contents but saw no improvement over conventional models in a series of classification tasks. Literature on graph embeddings is sometimes qualitatively evaluated by visualising the dimensionality reduced embedding space [215]. More specifically, Hamilton et al. [71] have shown that simple document and word embeddings can be enriched by using graph convolutions over a network of co-occurrence statistics. State-of-the-art language models like BERT [45], GPT-2 [157], Electra [38], and their derivatives have become so expressive nowadays, that incorporating additional sources of information is barely needed. In most use-cases, using general pre-trained models may already be sufficient and can be fine-tuning to a dataset or task if necessary. There are also editable neural networks to adapt models when almost no training data is available [192]. Even though these large models generalise very well, language still keeps evolving over time. Some use-cases may therefore benefit from using dynamic word embeddings [13].

2.5. Dimensionality Reduction

The goal of dimensionality reduction is to represent high-dimensional data in a lower-dimensional space while preserving the characteristics of the original data as best as possible. Typically, they aim to reproduce the same distribution of pairwise distances between points from the high-dimensional space on the lower-dimensional space. The most common application of dimensionality reduction is the projection of high-dimensional data into two dimensions for the purpose of visual interpretation. Generally, these methods follow one of three mathematical models. *Linear* models, such as principal component analysis (PCA) [148] can be calculated very efficiently and have proven to reduce input spaces to improve the performance of downstream tasks. This is due to the notion of intrinsic dimensionality which coincides with the indiscriminability of distances and features [81]. Although PCA-based can be used in visualisations for initial data exploration, other approaches are able to better preserve characteristics of a dataset in two dimensions. For example, the *non-linear* Sammon mapping [183] tries to preserve the structure of inter-point

distances from the high-dimensional space in the low-dimensional space. The resulting visualisations are generally better than PCA to show relatedness of individual data points. This was later improved even further with multidimensional scaling (MDS) [127] and its ISOMAP variant [200]. Others introduced collective component analysis (CoCo) for dimensionality reduction from multiple heterogeneous feature spaces [190]. A very different approach was taken with self organising maps (SOM) [104], where, as the name suggests, items of the dataset are allowed to flow freely in the two-dimensional space and iteratively move towards a globally optimal layout based on the objective function.

Lastly, there are *probabilistic* models like stochastic neighbour embeddings (SNE) [77]. They are similar to MDS in that they use inter-point distances but model these distances as probability distributions. The t-distributed SNE (tSNE) has proven to produce competitive results for visualising datasets while preserving characteristics [121], however its nondeterministic nature may produce greatly varying results. Even with some optimisations and efficient implementations, tSNE does not scale well. To this end, Poličar et al. [153] extended tSNE by introducing batch processing to reduce the dimensionality of large datasets. FltSNE is another optimisation of tSNE that significantly reduces the computational complexity [115]. Other variants are not only highly optimised, but can also be applied to growing datasets [46] or time-varying data [153]. Kobak et al. [102] have demonstrated, that reducing the degrees of freedom in the objective function helps to reveal finer local clustering structures. It is also important to note, that the initialisation has a significant impact on the final layout [101, 153]. Another way to influence the resulting projection is to use supervision based on user annotations or a classifier, which has the advantage, that the projection model could be reused to produce comparable visualisations of new data from the same domain [54, 70].

Most of the recent state-of-the-art algorithms for dimensionality reduction like LargeVis [197] and UMAP [123, 124] scale almost linearly by using efficient nearest neighbourhood approximations in the high-dimensional space and spectral embeddings [139] to initialise positions of points in the low-dimensional space to reduce the number of fine-tuning iterations. Others directly embed the similarity graph using a neural model [203] or as most recently proposed by minimising distortion functions [2]. Since UMAP has become very popular since its inception, there have been several adaptations, for example an autoencoder model using the same objective functions [177], a variant that better reflects local densities [136], and the addition of alignment between layouts of varying data [123]. Although UMAP is often considered to produce better results than tSNE, Kobak and Linderman [100] found, that UMAP does not preserve the global structure of a dataset any better than tSNE, as the initialisation has the most significant impact on the outcome.

The rise of deep learning has also impacted the research on dimensionality reduction. Even though tSNE, UMAP, and their derivatives have shown to produce representative projections of the high-dimensional data by preserving local and global pairwise similarities in

two dimensions, the concept of proximity in a high-dimensional space may not be qualitatively meaningful [1]. Deep learning models however offer a computationally viable way to use more complex objectives even for very large datasets. TopoAE [134] uses techniques from topological data analysis to train autoencoders that significantly improve the preservation of topological structures. A similar approach later improved the preservation of local geometries [114]. There are also deep learning approaches for semi-supervised dimensionality reduction, for example by incorporating a classification loss [8] or introducing pairwise constraints from domain knowledge [128]. These two example require a partial annotation of the data ahead of time. DeepSI [19] on the other hand allows for a feedback loop between human interaction and deep learning to enable the computer-assisted creation of custom representations. While all algorithms mentioned in this section can be applied to any high-dimensional dataset, there are also dimensionality reduction approaches that are specifically designed for meaningful representations of textual datasets [132, 186].

Choosing the most fitting algorithm for a specific use-case and tuning the hyper-parameters to obtain a satisfying result can be challenging. Hilasaca and Paulovich [75] propose a method to compare different dimensionality reduction techniques, identify their differences and creating hybrid projections by mixing existing projections.

2.6. Map-like Text Visualisation in Practice

In this section, we will look at the final stage in the visualisation pipeline by covering inspiring domain-specific interactive visualisations as well as challenges and possible solutions that are encountered in practical applications. In particular, we will focus on applications that follow the Spacial Paradigm for Information Retrieval and Exploration (SPIRE) [208]. This paradigm was first used in 1994 as part of a software for analysts to browse a patent corpus and discover relationships between the documents that are plotted on a two-dimensional canvas[41]. Even in more recent years, there have been several improvements in patent data visualisation for visual analytics and impact discovery [80, 94]. There are numerous applications that could benefit from interactive visualisations that are tightly integrated with an information retrieval system to analyse large heterogeneous data collections. For example, investigative journalists need to untangle and order huge amounts of information, search entities, and visualise found patterns [31, 39]. Similar datasets are of interest in the context of computational forensics [61]. Auditing firms and law enforcement need to sift through huge amounts of data to gather evidence of criminal activity, often involving communication networks and documents [93]. There are also analyses based on map-like visualisations of scientific publications in computer science [62], philosophy [143], and climate change research [25]. While these examples only utilise the textual information, VOSViewer provides an exploration interface for the underlying bibliometric networks [202]. Others proposed domain-independent map-like text visualisa-

tions, for example to jointly explore Wikipedia articles and associated categorical information [188] or fully data-driven categorisation through clustering [44]. It has been shown that displaying individual documents in their global context of large book corpora is very effective [102, 186], as the interactive maps allow users to gain insights that would otherwise remain hidden. The OpenSyllabus Project[1] has started to integrate such an interface to explore their collection. Pang et al. [146] conducted a user study and found that transferring concepts and analogies from geographic maps to these artificial maps helps users to get a better overview of their digital library. More recently, Ambavi et al. [5] developed a platform to explore COVID-19 information with multi-faced search integrated to a two-dimensional visualisation. Others take the analogy of a geographical map back to the actual map of the world by projecting documents to real-world locations based on mentioned entities [125] or mapping extracted information to map overlays [26]. Until now, we only mentioned maps of textual data. It is worth mentioning though, that there are applications for other data sources, for example for medical treatments [55, 206], sleep patterns in somnology research [21], bird voices for ornithologists [178], and mapping genomes [16].

In order to utilise the full potential of map-like visualisations, integrating information retrieval technologies is crucial. Exploring areas of semantically related documents could be considered as a bottom-up approach for gaining insights. With the addition of a search functionality, users can then identify the areas of interested from the top down. The most basic approach would be to highlight the documents on the map that fit to a query. Adding a heatmap overlay that updates based on the density of retrieved documents has been shown to be more effective and easier to interpret [66]. Users also need a simple way to interpret what a neighbourhood of documents on the map contains. To this end, Klouche et al. [99] strategically place keyphrases on the map to summarise the contents. Based on search results, these keywords are updated in order to be more descriptive for the specific context. Systems like Kyrix-S could augment the potentially computationally intensive context-sensitive updates when zooming, panning, selecting documents, or searching [199]. Ji et al. [88] developed a system to analyse the features of an deep-learning-based information retrieval system in a two-dimensional space. In this way, they were able to improve feature-selection decisions for complex models and fine-tuning a model to domain-specific user-cases. SciNoon is a scientific search engine that exposes several parameters of the underlying information retrieval system in such a way, that users can intuitively fine-tune their search results and build mind-map-like summary view of the data [137].

In the previous sections, we also mentioned the importance of representing datasets that evolve or grow over time. In practical applications, it is essential that users are able to keep track of the changes between different versions of the projected data. This could be done by placing fixed points in the two-dimensional space that act as anchors [122]. However, this

[1] `http://galaxy.opensyllabus.org/`

requires some level of intervention and is not fully data-driven. To this end, Archambault and Purchase [9] conducted a series of experiments to test how to best preserve the mental map of dynamic graph drawings. Preserving the general layout of the dataset is not the only important aspect to consider. Also the descriptive keywords have to retain some level of familiarity between the different versions [82].

As we have shown, there are numerous use-case and domain-specific applications for map-like visualisations. However, a fundamental challenge for all of them is to provide a meaningful and expressive two-dimensional representation of the data. For example, the embedding models of textual corpora used by the dimensionality reduction algorithm, contain many ambiguities and overlapping word senses [10] as well as semantic and syntactic subspaces [160]. Different layout objectives may produce contradicting results and the challenges of processing big data need to be addressed [20]. In order to evaluate the quality of a projection of the input data is, Mokbel et al. [133] proposed point-wise quality measures that can be overlaid on the map to show how well similarities were preserved. Depending on the use-case, users may want the ability to partially manipulate the layout directly without the need to change hyper-parameters and computing a new layout. For large datasets, manual edits without computer assistance becomes infeasible. We identify three ways in the related work to incorporate user feedback into the layout process. First, by preconditioning the layout process. Here, dimensionality reduction algorithms either use (partially) user annotated data [8, 130], manually annotated pairs of very similar or dissimilar items from a dataset [128], or the map is initialised by placing a few items on the empty canvas [179]. Second, by interactive model parametrisation, which allows users to update the layout by changing model parameters [190], composing a mixture of multiple models [75], or simultaneously learning a representation across multiple views [119]. Third, by directly editing existing layouts, where users can edit the position of points in an existing two-dimensional layout by dragging.

For a more in-depth overview of map-like visualisations, we refer interested readers to this special issue on mapping knowledge [191] and the recent survey by Hogräfer et al. [78].

2.7. The Map of Related Work

In this chapter, we provided an overview of the related work on map-like data visualisation. Figure 2.2 shows a practical example of such a visualisation to provide an overview of the related work covered in this thesis. Of the 216 references cited in the thesis, we were able to automatically retrieve complete bibliographic information from SemanticScholar[2] for 93 papers based on the title or DOI. Among others, this includes the authors, publishing year and venue, respective references, citations, as well as abstracts. To enrich the dataset,

[2] `https://www.semanticscholar.org/`

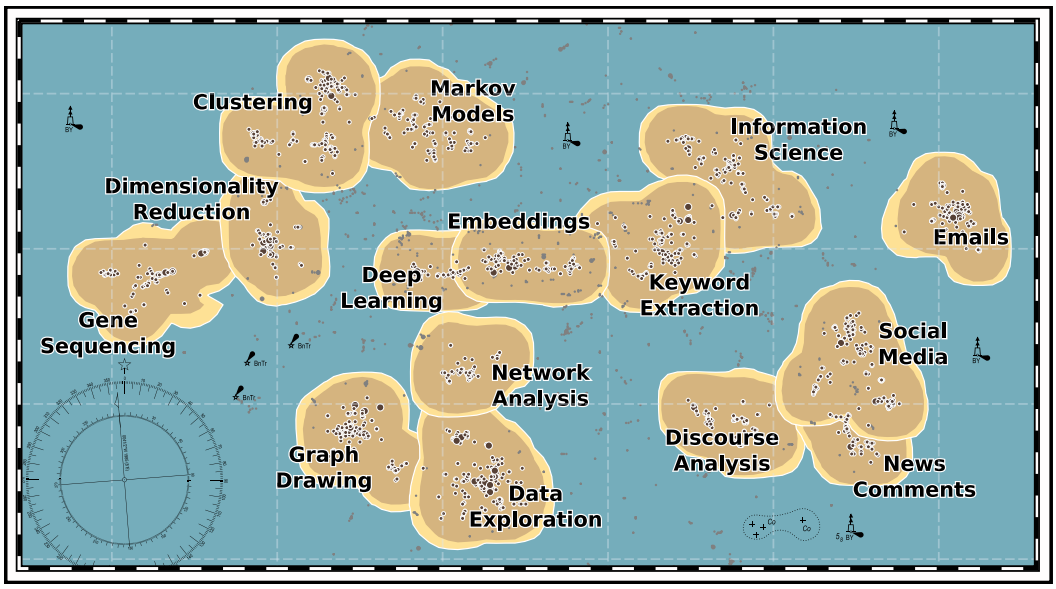


Figure 2.2.: Visualisation of the research landscape of literature cited in this thesis. Each paper is drawn as a dot on the map and clusters thereof form islands. The layout of papers, formation of islands, and generation of keywords for overlay is fully data-driven. The map was later stylised by hand.

we also retrieved that information of all the references that these papers cited which leas to an overall number of 2,916 papers.

The abstracts of these papers are embedded using a pre-trained Sentence–BERT [161] model and projected to two dimensions with openTSNE [154] and are drawn as dots on the map.We then clustered this two-dimensional space using HDBSCAN [27], a density-based clustering algorithm, which resulted in 15 clusters. Note, that the number of clusters is not set directly and is a result of hyper-parameter value of the minimum cluster size. The mean cluster size is 112 (σ =34.9) and 1236 papers were considered outliers.

Each cluster forms an "island" on the nautically themed chart. In order to calculate the shape of the island, we compute a grid-based density map for each cluster, where the density below a threshold forms the different shade of the coastal area. Candidates for island names are based on the tf-idf scores of n-grams of length 1 and 2 on 15 meta-documents of concatenated abstracts for each cluster. The most fitting of the top ten candidates was placed by hand along with symbols typically seen on nautical charts to improve the aesthetics.

This fully data-driven approach leads to an overview which represents the research landscape this thesis covers very well. The chain of four islands on the north east is mostly

covered in this chapter. “Gene Sequencing” may seem to be off-topic but stems from the fact, that UMAP [123] was developed by computational biologists. The three islands in the south correspond to graph drawing, social network analysis and visual search topics. Language models and text mining is represented by the chain of four islands in the centre of the chart. Just like this thesis is structured in two parts, the islands in the east are more separated from the rest, covering the processing of email data (Chapter 6) and news comment analysis (Chapter 7).

Chapter 3

JOINT VISUALISATION OF TEXT AND NETWORK DATA

Real-world text corpora typically contain heterogeneous documents which are associated with auxiliary information. This additional information can be in the form of meta-data, such as the time a document was created or by whom it was edited. They can also contain inherent graph structures, for example the communication network of an email corpus or the citations and co-authorships in academic publications. The texts themselves may also describe relationships between entities, which can be revealed with the help of named entity recognition and relationship extraction. Depending on the use case, all these different sources of information may be valuable for users to explore and analyse.

Problem Statement Typically, when it comes to visualising large corpora with inherent graph structures, either the textual content or the network graph is used. In this way, parts of the available information is lost in either one and users would have to manually find the connections between the two aspects. We propose to jointly visualise the syntactic and semantic information in a single layout of the corpus. This comes with the challenge, that both aspects contain possibly contradicting information. The example, a visualisation of the communication graph of an email corpus should draw communities such that they are clearly separable. The same applies to topical clusters of a visualisation of the messages. However, there may be topical overlaps between communities.

Contributions In this chapter, we propose MODIR, an algorithm that not only visualises the semantic information encoded in the documents' content but also the relationships expressed by the inherent network information. Our algorithm based on multi-objective optimisation to jointly position embedded documents and graph nodes in a two-dimensional landscape. We illustrate the effectiveness of our approach with real-world datasets and show that we can capture the semantics of large document collections better than visualisations based on either the content or the network information. We also developed a prototype to interactively explore this new type of visualisation.

3.1. Introduction

(a) Word cloud (content) (b) Company network

Figure 3.1.: Duality of business news articles.

Exploring large document collections is a cumbersome, but necessary task to gain an overview or to find interesting, serendipitous information. Depending on the collection, the exploration either focuses on the content, for example by using topic modelling methods to get an overview, or on the network formed by the connections of documents among each other. Most digital library collections exhibit this duality; they can be represented as *text* or *network*. Figure 3.1 exemplifies how the two representations can be visualised, e.g. using word clouds and graphs. The duality is most apparent in collections of web pages, where links connect the pages with each other to form the web graph. But it is also prevalent in email collections or corpora originating from communication in social networks, such as chats, blogs, or tweets. Often, analysing the communication network is more revealing than focusing on the content. While these are some examples of document collections that exhibit explicit network structure, most document collections can be enriched with network structure by extracting information from the content or by analysing the documents' metadata. For example, bibliometrics makes heavy use of both types of information: content of documents (research publications, patents, etc.) and co-author and (co-)citation networks. Visualising corpora is inevitable to analyse or explore the collections. But usually either the content or the network structure is neglected, missing out on important relations and insights about the document collection at hand.

In more heterogeneous data collections, exploration or getting an overview of datasets is insurmountable with current tools. The sheer amount of documents prohibits simple visualisations of networks or meaningful keyword-driven summaries of the textual content. Examples of these extremely difficult cases are in the context of data-driven journalism, computational forensics, or auditing. Data-driven journalism [39] often has to deal with leaked, unstructured, very heterogeneous data, e.g. in the context of the Panama Papers, where journalists needed to untangle and order huge amounts of information, search entities, and visualise found patterns [31]. Similar datasets are of interest in the context of computational forensics [61]. Auditing firms and law enforcement need to sift through huge amounts of data to gather evidence of criminal activity, often involving communication networks and documents [93]. Users investigating such data want to be able to quickly gain an overview of its entirety, since the large amount of heterogeneous data renders experts' investigations by hand infeasible. Computer-aided exploration tools can

support their work to identify irregularities, inappropriate content, or suspicious patterns. Current tools[1] lack sufficient semantic support, for example by incorporating document embeddings [131] and the ability to combine text and network information intuitively.

We propose MODIR, a scalable **M**ulti-**O**bjective **Di**mensionality **R**eduction algorithm, and show how it can be used to generate an overview of entire document collections with inherent network information in a single interactive visualisation. Special graph databases enable the efficient storage of large relationship networks and provide interfaces to query or analyse the data. However, without prior knowledge, it is practically impossible to gain an overview or quick insights into global network structures. Although traditional node-link visualisations of a network can provide this overview, all semantic information from associated textual content is lost completely.

Technically, our goal is to combine network layouts with dimensionality reduction of high-dimensional semantic embedding spaces. Giving an overview over latent structures and topics in one visualisation may significantly improve the exploration of a corpus by users unfamiliar with the domain and terminology. This means, we have to integrate multiple aspects of the documents, namely the semantics of the textual content and the relations and connections inherent to the collection, into a single visualisation. The challenge is to provide an intuitive, two-dimensional representation of both the network and the text, while balancing potentially contradicting objectives of these representations.

In contrast to existing dimensionality reduction methods, such as tSNE [121], we propose a novel approach to transform high-dimensional data into two dimensions while *optimising multiple constraints* simultaneously to ensure an optimal layout of semantic information extracted from text and the associated network. To minimise the computational complexity that would come from a naive combination of network drawing and dimensionality reduction algorithms, we formally use the notion of a hypergraph. In this way, we are able to move repeated expensive computations from the iterative document-centred optimisation to a preprocessing step that constructs the hypergraph. We use real-world document collections from different domains to demonstrate the effectiveness and flexibility of our approach. MODIR-generated representations are compared to a series of baselines and state-of-the-art visualisation and dimensionality reduction methods. We further show that our integrated view of these document collections is superior to approaches focusing on text-only or network-only information when computing their visualisations. This work focuses on describing and evaluating multi-objective dimensionality-reduction. We also demonstrate how to utilise the visualisation in an exploration interface, which is available on our website.[2].

[1] e.g. `https://www.nuix.com/` or `https://linkurio.us/`
[2] `https://hpi.de/naumann/s/modir`

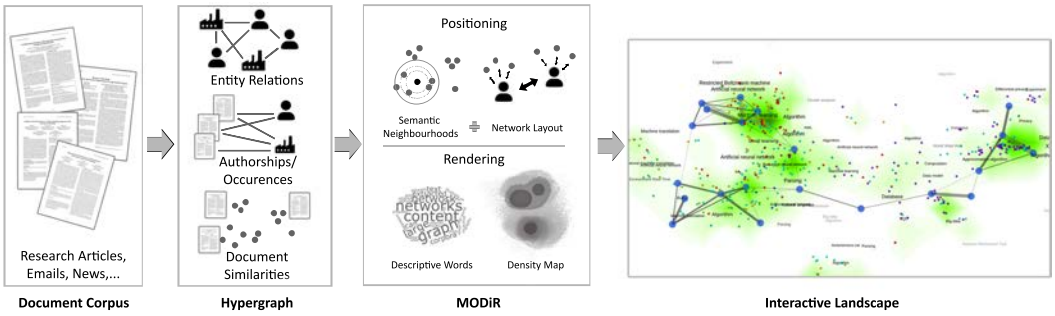


Figure 3.2.: Overview of MODIR for joint visualisation of research articles with co-authorship networks, email corpora, and more

With MODIR we bridge the gap between text and network visualisation by jointly reducing the dimensionality of the input data. Therefore we subdivided this part into three sections to highlight related work in the areas of text visualisation, representation learning, as well as dimensionality reduction. Other work that tries to jointly model text and networks but without dimensionality reduction and without a focus on visualisation is *LINE* [196]. They generate information networks consisting of different types of nodes, e.g. words from document content and authors from document metadata. Another tool that investigates combining graph structure with textual elements is *VOSviewer* [202]. They construct and visualise bibliographic networks that provide a multi-view interface to explore and filter keywords and network aspects of such datasets. In our work we go beyond building a network from textual data but instead project the textual data into a latent space.

3.2. Multi-objective Dimensionality Reduction

Visualisations of complex datasets are restricted to two or three dimensions for users to grasp the structure and patterns of the data. We integrate multiple kinds of information (i.e., documents and persons) into a joint visualisation as depicted on the far right in Figure 3.2, which we call *landscape*. This landscape consists of a base-layer containing all documents depicted as dots forming the *document landscape*; nodes and their connections are placed on top of this base-layer as circles connected by lines forming the *graph layer*. In this section, we describe the MODIR algorithm which integrates multiple objectives during the layout process to find an overall good fit of the data within the different layers. Our approach is derived from state-of-the-art methods for drawing either the network layer or the document landscape. We formally model the data as part of a hypergraph, which we abstractly depict on the left in Figure 3.2. This allows for a more simple implementation

of the algorithm and easier data structures that operate on (cached) sets as opposed to traversing a "normal" graph structure.

We assume that documents are given as high-dimensional vectors and entities are linked among one another and to the documents. These links are used as restrictions during the multi-objective dimensionality reduction of document vectors. Let $\boldsymbol{x}^{(i)} \in \mathbb{X} \subset \mathbb{R}^d$ be the set of n documents in their d-dimensional representation and $\boldsymbol{y}^{(i)} \in \mathbb{Y} \subset \mathbb{R}^2$ the respective positions on the document landscape. Let $\mathscr{H}(\mathscr{V}, \mathscr{E})$ be a hypergraph based on the network information inferred from the document corpus, with vertices $\mathscr{V} = \mathbb{X} \bigcup \mathbb{P}$, where $\mathbb{X}$ are the documents and $p_i \in \mathbb{P}$ are the entities in the network and hyperedges $e_k \in \mathscr{E}$ describing the relation between documents and entities. For each pair of entities $p_m, p_n \in \mathbb{P}$ that are connected in the context of documents $\boldsymbol{x}^{(i)}, \ldots \in \mathbb{X}$, there is a hyperedge $e_k = \{p_m, p_n, \boldsymbol{x}^{(i)}, \ldots\}$. Analogously, the same definition applies to $\mathbb{Y}$. Further, $\mathscr{H}^{\mathbb{Y}}$ or $\mathscr{H}^{\mathbb{X}}$ is used to explicitly state the respective document representation used. The position in the graph layer $\pi : \mathbb{P} \to \mathbb{R}^2$ of an entity p_m is defined as

$$\pi(p_m; \mathscr{H}^{\mathbb{Y}}) = \frac{1}{N_{p_m}} \sum_{e_k \in \mathscr{E}_{p_m}} \sum_{\boldsymbol{y}^{(i)} \in e_k \setminus \mathbb{P}} \boldsymbol{y}^{(i)}, \tag{3.1}$$

where $\mathscr{E}_{p_m} \subset \mathscr{H}^{\mathbb{Y}}$ is the set of hyperedges containing p_m and N_{p_m} is the number of documents p_m is associated with:

$$N_{p_m} := \left| \{\boldsymbol{x}^{(i)} \in \mathbb{X} | \exists e_k \in \mathscr{E} : \boldsymbol{x}^{(i)} \in e_k \wedge p_m \in e_k\} \right|.$$

This effectively places an entity at the centre of its respective documents. More elaborate methods like a density-based weighted average are also applicable to mitigate the influence of outliers. For simplicity we will abbreviate $\pi(p_m; \mathscr{H}^{\mathbb{Y}})$ as π_m.

Let $\psi : \mathbb{X} \to \mathbb{Y}$ be the projection $\psi(\boldsymbol{x}^{(i)}; \boldsymbol{W}) = \boldsymbol{W}_{i,:} = \boldsymbol{y}^{(i)}$, where $\boldsymbol{W} \in \mathbb{R}^{2 \times n}$ is the projection matrix leant by MODIR based on multiple objectives $\varphi_{\{1,2,3\}}$ using gradient descend, as defined later in this section. The objectives are weighted by manually set parameters $\theta_{\{1,2,3\}}$ to balance the effects that favour principles focused on either the graph layer or the document landscape, as they may contradict one another. Given a high-dimensional hypergraph $\mathscr{H}^{\mathbb{X}}$, the matrix $\boldsymbol{W}$, and a entity projection π, we define the resulting multi-objective dimensionality reduction function as

$$\Psi(\mathscr{H}^{\mathbb{X}}, \boldsymbol{W}, \pi) = \mathscr{H}^{\mathbb{Y}}.$$

We summarise the most important definitions in Table 3.1.

In the following paragraphs, we will formally introduce MODIR's objectives. *Objectives (1) and (2)* are inspired by tSNE and use the neighbourhood context of documents in $\mathbb{X}$ to position similar documents near one another and unrelated ones further apart in $\mathbb{Y}$. *Objective (3)* attracts documents based on co-occurrence in hyperedges so that the resulting π_m

Table 3.1.: Overview of Symbols

Symbol	Description
$\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}$	Document vector and its position on the landscape
p_i, π_i	Entity in the graph and its position on the landscape
φ_1	Objective to pairwise attract similar documents
φ_2	Objective to pairwise repel dissimilar documents
φ_3	Objective to attract pairs of documents and entities
$\theta_{\{1,2,3\}}$	Weights for influence of objectives on Φ
$\mathbb{X}^{k,\boldsymbol{x}^{(i)}}$	Semantic neighbourhood of $\boldsymbol{x}^{(i)}$ with size k
$\bar{\mathbb{X}}^{l,\boldsymbol{x}^{(i)}}$	Non-similar neighbourhood of $\boldsymbol{x}^{(i)}$ with size l
$\mathscr{E}^{\mathbb{X}}_{\boldsymbol{x}^{(i)}}$	Set of documents connected to $\boldsymbol{x}^{(i)}$ via any entity; sampled down to size s

will be closer if they are well connected in the graph. This third objective also implicitly brings documents closer to their respective entities.

Objective (1): Similar documents are near one another. Semantically similar documents should be closer on the document landscape and dissimilar ones further apart. To measure the semantic similarity of documents, Maaten and Hinton [121] used a naïve bag-of-words representation. Although tSNE preserves the inherent semantic structure in two-dimensional representations from these sparse vectors [151], we opted to use document embeddings. This has the advantage that, when only part of the data is visualised, the embedding model can still be trained on a larger set of documents and thus retain the additional information. Objective (1) is inspired by the efficient usage of context words in word2vec [131]. Corresponding to the skip-gram model, we define the context $\mathbb{X}^{k,\boldsymbol{x}^{(i)}} \subset \mathbb{X}$ of a document $\boldsymbol{x}^{(i)}$ by its k nearest neighbours in the embedding space. The first objective is defined as

$$\varphi_1(x^{(i)}) = \sigma\Big(\sum_{\boldsymbol{x}^{(j)} \in \mathbb{X}^{k,\boldsymbol{x}^{(i)}}} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\| - \|\boldsymbol{y}^{(i)} - \boldsymbol{y}^{(j)}\| \Big) \tag{3.2}$$

with σ being the sigmoid function and $\|\cdot\|$ the Euclidean norm. Distances are normalised based on the context to make them comparable between the high-dimensional and two-dimensional space and rescaled by the sigmoid.

Objective (2): Dissimilar documents are apart from one another. The optimal solution to the previously defined objective would be to project all documents onto the same point on the two-dimensional canvas. In order to counteract that, we introduce

negative examples for each pair of context documents. We do so by sampling a set of l documents that are not in the k neighbourhood of $\boldsymbol{x}^{(i)}$. Let $\bar{\mathbb{X}}^{l,\boldsymbol{x}^{(i)}} \subset \mathbb{X} \setminus \mathbb{X}^{k,\boldsymbol{x}^{(i)}}$ be the set of negative samples for $\boldsymbol{x}^{(i)}$, then the second objective is defined as

$$\varphi_2(\boldsymbol{x}^{(i)}) = -\sigma\Big(\sum_{\boldsymbol{x}^{(j)} \in \bar{\mathbb{X}}^{l,\boldsymbol{x}^{(i)}}} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\| - \|\boldsymbol{y}^{(i)} - \boldsymbol{y}^{(j)}\| \Big). \tag{3.3}$$

This objective prevents crowding on the centre of the landscape and helps to better preserve the global structure.

Objective (3): Connected entities are near one another and their documents. This object serves two purposes: All documents $\boldsymbol{y}^{(i)}$ associated with an entity p_m are placed near its π_m position in the graph layer and two entities π_m and π_n are forced near one another if they are connected.

Let $\mathscr{E}_{\boldsymbol{y}^{(i)}} \subset \mathscr{E}$ be the set of hyperedges in the hypergraph $\mathscr{H}$ containing the document $\boldsymbol{y}^{(i)}$ and $\mathscr{E}^{\mathbb{Y}}_{\boldsymbol{y}^{(i)}} = \bigcup_{e_k \in \mathscr{E}_{\boldsymbol{y}^{(i)}}} e_k \setminus \mathbb{P}$ all documents that are linked to $\boldsymbol{y}^{(i)}$ through an entity, then the third objective is defined as

$$\varphi_3(\boldsymbol{y}^{(i)}) = \sigma\Big(\sum_{\boldsymbol{y}^{(j)} \in \mathscr{E}^{\mathbb{Y}}_{\boldsymbol{y}^{(i)}}} \|\boldsymbol{y}^{(i)} - \boldsymbol{y}^{(j)}\| \Big), \tag{3.4}$$

which, when minimised, attracts documents that are related through entities. This has two implicit effects: An entity p_m gets closer to its documents as they are attracted to π_m without having to explicitly compute this position using Equation 3.1. Also, related entities p_m, p_n are attracted to one another since they appear in the same hyperedges. The computational complexity of this objective is strongly related to the connectedness of entities in the graph. For dense graphs, we propose a heuristic by only using a subset of s documents from the context $\mathscr{E}^{\mathbb{Y}}_{\boldsymbol{y}^{(i)}}$ of $\boldsymbol{y}^{(i)}$. An objective modelling a repulsive force as in force-directed graph layouts is not needed as the first two objectives $\varphi_{\{1,2\}}$ provide enough counteracting force.

Algorithm. The positions of entities and documents on the landscape are calculated using the previously defined objectives as follows. First, we construct the hypergraph $\mathscr{H}^{\mathbb{X}}$ with document contexts including the set of k-neighbourhoods $\mathbb{X}^{k,\boldsymbol{x}^{(i)}}$. Relevant pairwise distances can be stored in an adjacency matrix so reduce computational overhead in Equations 3.2 and 3.3. For more efficient training, the randomly sampled l negative neighbourhoods $\bar{\mathbb{X}}^{l,\boldsymbol{x}^{(i)}}$ can be prepared ahead of time and then only masked during later. The s-neighbourhoods for entities in Equation 3.4 $\mathscr{E}^{\mathbb{Y}}_{\boldsymbol{y}^{(i)}}$ can only be prepared with references, as $\mathbb{Y}_{\boldsymbol{y}^{(i)}}$ updates with each iteration. We designed the algorithm to move as much repetitive

computations to pre-processing ahead of time or each epoch. Creating these sets is very efficient using Hierarchical Navigable Small World graphs (HNSW) for approximate nearest neighbour search [14]. Overall we are able to reduce the pre-processing complexity to $\mathcal{O}(n \log n)$ and for each iteration $\mathcal{O}(kln)$, with $k, l \ll n$ near linear. After generating the context sets, we use gradient descend to update the projection matrix $\boldsymbol{W}$ (rows are $\boldsymbol{y}^{(i)}$) with learning rate η reducing the overall error Φ as defined by

$$\Phi(x_i) = \theta_1 \varphi_1(\boldsymbol{x}^{(i)}) + \theta_2 \varphi_2(\boldsymbol{x}^{(i)}) + \theta_3 \varphi_3(\boldsymbol{x}^{(i)}). \tag{3.5}$$

Selecting appropriate values for the hyperparameters k, l, s, and $\theta_{\{1,2,3\}}$ is critical to produce meaningful results. We found $l = k$ in all experiments to produce the best results as this way for every similar document the model has one dissimilar document to compare. Inspired by tSNE [121], we limit hyperparameters by setting k and s dynamically for each document based on a user-defined perplexity. With these adaptations, the only parameters to be set are the perplexity β that roughly determines the context size, the learning rate η, and the objective weights, which can often stay at a default setting. A reference implementation including a modular processing pipeline for different datasets, approaches, and experiments is available online.[3]

3.3. Evaluation

Our approach can be used in a variety of different scenarios. Communication datasets, such as emails, are particularly interesting, since understanding this data or getting an overview of it necessitates the analysis and visualisation of both, content and meta-data. While there exist these kinds of document collections, i. e.the Enron corpus [98], they typically lack ground truth for evaluation purposes. Other kinds of document collections are more accessible regarding evaluation: research publications and their co-authorship network. Therefore we focus our experiments on collections of scientific articles and how they can be visualised using their content and information about co-authorship. Results of dimensionality reduction can be subjective, so as in prior work on dimensionality reduction [121, 123, 188], we will compare our approach to a variety of baselines in a qualitative discussion of the results as well as a series of quantitative experiments. To the best of our knowledge, there are no algorithms that use multiple objectives for dimensionality reduction of high-dimensional data. Popular approaches for common dimensionality reduction are tSNE and PCA. Although UMAP has grown in popularity since publishing the papers this chapter is based on, we did not update our experiments since results are mostly comparable to tSNE with the correct hyper-parameter settings [102]. As baselines, we use the original optimised implementation of tSNE[4] written in C as provided by the authors.

[3] `https://hpi.de/naumann/s/modir`
[4] `https://lvdmaaten.github.io/tsne/`

Table 3.3.: Venue-based community assignments and number of articles in Semantic Scholar (S2) and AMiner (AM)

		# Articles	
Community	Venues	S2	AM
Data Mining	KDD, ICDM, CIKM, WSDM	4,728	13,699
Database	SIGMOD, VLDB, ICDE, EDBT	7,155	14,888
ML	NeurIPS, AAAI, ICML, IJCAI	10,374	41,815
NLP	EMNLP, ACL, CoNLL, COLING	41,815	22,523
Comp Vision	CVPR, ICCV, ICIP, SIGGRAPH	11,898	43,558
HCI	CHI, IUI, UIST, CSCW	8,608	33,615

3.3.1. Datasets

The motivation for this chapter is to visualise inherent network structure along with their respective text documents for exploring and understanding large document collections. We argue, that our approach is applicable to any given document collection with inherent graph structures, so we include a variety of examples for evaluation. We apply MODIR to the Enron corpus [98] which originally consists of around 600,000 messages belonging to 158 users and QUAGGA [163] to extract individual emails from quoted conversations, remove duplicates, extract additional correspondents from inline metadata, and try to combine the aliases of people. Assessing the quality of a given layout requires very specific domain knowledge including deep understanding of semantic structure across all documents and a close familiarity with entity relations. Due to the lack of a gold standard or domain knowledge on our side, we consider additional sources.

Academic co-authorship networks and their respective publications have well defined labels provided by venues or communities, so there are no ambiguities or additional annotations needed. We make use of two processed and publicly available corpora of research articles, the AMiner[5] network (AM) [195] published in 2008 with over two million papers by 1.7 million authors and the recently published Semantic Scholar[6] Open Corpus (S2) [7] with over 45 million articles at the time of writing the paper these experiments are based on [165]. Both corpora cover a range of different scientific fields. Semantic Scholar for example integrates multiple data sources like DBLP and PubMed and mostly covers computer science, neuroscience, and biomedical research. Unlike DBLP however, S2 and AM not only contain bibliographic metadata, such as authors, date, venue, citations, but also

[5] https://aminer.org/billboard/aminernetwork
[6] https://api.semanticscholar.org/corpus/

Table 3.4.: Number of documents, entities, and their connections in filtered datasets used in this thesis

Dataset	# Documents	# Nodes	# Edges
AMiner (AM)	49,670	56,449	110,146
SemanticScholar (S2)	170,098	183,198	701,442
SmallScholar (S2b)	489	24	39

abstracts to most articles, that we use to train document embeddings using the Doc2Vec model in Gensim [7]. Similar to Cavallari et al. [30], we remove articles with missing information and select from six communities that are aggregated by venues as listed in Table 3.5. In this way, we reduce the size and also remove clearly unrelated computer science articles and biomedical studies. For in depth comparisons, we reduce the S2 dataset to 24 hand-picked authors, their co-authors, and their papers (S2b).

Note, that the characteristics of the networks differ greatly as the ratio between documents, nodes, and edges in Table 3.4 shows. In an email corpus, a larger number of documents is attributed to fewer nodes and the distribution has a high variance (some people write few emails, others write a lot). In the academic corpora on the other hand, the number of documents per author is usually relatively low.

3.3.2. Hyperparameter Settings

For MODIR, the context sizes are the most important parameters. Generally, small numbers for k, l, s perform better. This is in line with our expectations, as each item $\boldsymbol{x}^{(i)}$ will also be in the context of its respective neighbours and therefore amplify its attractive force. A large number for k for example will force all points towards the centre of the canvas or if even larger, produce random scatter as the gradients amplify. In our experiments we use $k = 10$, for datasets with a few thousand samples, k should usually be below l. We also found, that the negative context is best with $l = 20$ for all sizes.

Furthermore, we set both $\theta_1 = \theta_2 = 1.0$ for all experiments because the influence on selecting k, l is much larger. The graph context is also set to $s = 10$ (in our dataset the number of entities is close to the number of documents), the objective weight can be freely adjusted between around $0.8 \leq \theta_3 \leq 1.2$ to set the influence of the entity network. Similar to the semantic neighbourhoods in the first and second objective, the choice of s is significantly

[7] `https://radimrehurek.com/gensim/`; embedding size: 64 dimensions, vocabulary size: 20k tokens, trained for 500 epochs

Table 3.5.: Clustering Quality for Semantic Scholar (S2) and AMiner (AM)

	Doc2Vec		tSNE		PCA		MODIR	
Community	S2	AM	S2	AM	S2	AM	S2	AM
Data Mining	0.49	0.39	0.30	0.55	0.52	0.55	0.39	0.42
Database	0.49	0.82	0.64	0.34	0.47	0.34	0.69	0.32
ML	0.51	0.35	0.21	0.23	0.38	0.23	0.35	0.23
NLP	0.58	0.76	0.73	0.34	0.81	0.34	0.73	0.68
Comp Vision	0.51	0.67	0.56	0.39	0.49	0.39	0.54	0.29
HCI	0.64	0.68	0.47	0.41	0.61	0.41	0.39	0.38
Average	0.54	0.61	0.49	0.37	**0.54**	0.38	0.53	**0.39**

more influential than θ_3. Setting $\theta_1 = \theta_2 = 0$ to get a network-only layout would not work as the optimum would be placing all points at the point of origin. However, it is possible to "turn off" the influence of the network information on the layout by setting $\theta_3 = 0$.

The speed of convergence depends on the learning rate η and thus dictates the number of maximum iterations. Early stopping with a threshold on the update rate could be implemented. Depending on the size of the dataset and a fixed learning rate of $\eta = 0.01$, MODIR generally converges after 10 to 200 iterations, for larger and more connected data it is advisable to use a higher learning rate in the first epoch for initialisation and then reducing it to very small updates. For better comparability, we use a constant number of iterations of $T = 100$. In our experiments using tSNE, we set the perplexity to $Perp(P_i) = 5$, $\theta = 0.5$ and run it for 1,000 iterations.

3.3.3. Quantitative Evaluation

As Maaten and Hinton [121] state, it is by definition impossible to fully represent the structure of intrinsically high-dimensional data, such as a set of document embeddings, in two dimensions. However, stochastic neighbour embeddings are able to capture intrinsic structures well in two dimensional representations [102]. To measure this capability, we compare the ability of k-means++ [11] to cluster the high- and two-dimensional space. We set the number of clusters to the number or research communities ($k = 6$) and calculate the percentage of papers for each community per cluster. Therefore we assign each community to the cluster with most respective papers and make sure to use a clustering with an even distribution. Results are listed in Table 3.5 for tSNE, PCA, MODIR, and the original high dimensional embedding averaged over five runs. We see, that as expected due to

Table 3.6.: AtEdge-length of resulting graph layouts

Algorithm	AMiner	SemanticScholar	Enron
tSNE	5.32	4.09	3.89
PCA	5.00	3.91	3.60
MODIR	**4.79**	**2.94**	**2.59**

topical overlap of communities, even original embeddings cannot be accurately clustered. Interestingly though, there seems to be a significant difference between AMiner (AM) and S2 although the sets of papers intersect, which we assume is due to the fact, that S2 is larger and additionally contains more recent papers. Although PCA often does not generate visualisations in which classes can be clearly distinguished, the clustering algorithm is still able to separate them with competitive results compared to tSNE and MODIR.

MODIR not only aims to produce a good document landscape, but also a good layout of the network layer. Graph layouts are well studied, thus we refer to related work on aesthetics [156] and readability [141]. While these are very elaborate and consider many aspects, we decided to use Noack's normalised AtEdge-length [142]:

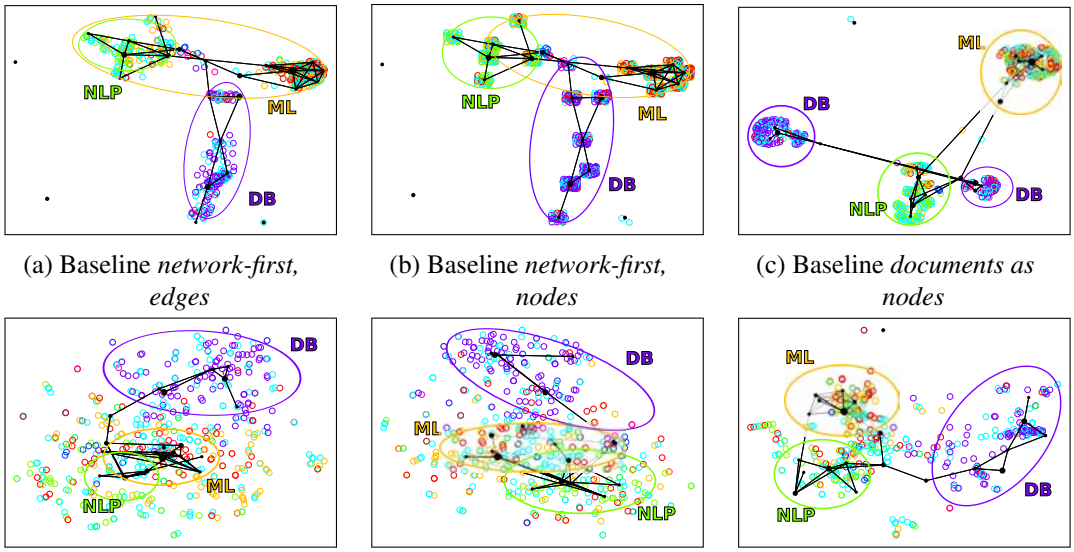
$$AtEdge = \frac{\sum_i \sum_j \|\pi_i - \pi_j\|}{|E|} \Big/ \frac{\sum_i \sum_j \|\pi_i - \pi_j\|}{|\mathscr{P}|^2}.$$

It describes how well the space utilisation is by measuring whether edges are as short as possible with respect to the size and density of the graph. Table 3.6 contains the results.

Although the AtEdge metric is comparable for layouts of the same graph, it is not comparable between datasets as can be seen by the fact, that a larger number of edges causes an overall lower score. The AtEdge length produced by PCA is generally better than that of tSNE while MODIR outperforms both as our approach specifically includes an optimised network layout. The better performance of PCA over tSNE can be explained by the resulting layouts being more densely clustered in one spot. Although the AtEdge length aims to give a lower score for too close positioning, it is not able to balance that to the many very long edges in the layout produced by tSNE.

3.3.4. Qualitative Evaluation

Apart from a purely quantitative evaluation, we use the hand-selected Semantic Scholar dataset (S2b) to visually compare network-centric baselines (a-c), document-focused baselines (d-e) and MODIR (f) in Figure 3.3. Papers are depicted as circles where the stroke



(a) Baseline *network-first, edges*

(b) Baseline *network-first, nodes*

(c) Baseline *documents as nodes*

(d) Baseline *documents-first*

(e) Baseline *node+document tSNE*

(f) MODIR

Figure 3.3.: Semantic Scholar co-authorship network (S2b), subsampled for readability; (a) the network is laid out first, documents are randomly placed along edges; (b) the network is laid out first, documents are randomly placed around nodes; (c) documents are part of the network layout as nodes in the graph that replace author-author edges; (d) the document landscape is laid out first, nodes are positioned at the centre of their associated documents; (e) tSNE is applied on papers and authors together, where documents are aggregated to represent authors

colour corresponds to the communities, black lines and dots are authors and their co-authorships, size corresponds to the number of publications. For better readability and comparability, the number of drawn points is reduced and three communities are marked.

In Figure 3.3a we use the weighted co-authorship network drawn using [63] and scatter the papers along their respective edges after the graph is laid out. We see, that active collaboration is easy to identify as densely populated edges and research communities of selected areas are mostly coherent and unconnected researchers are spatially separated from others. Although it is possible to distinguish the different communities in the graph layer,the document landscape is not as clear. The ML researchers are split apart from the rest of the NLP community, which in turn is overcrowded. Figure 3.3b uses the same network layout but places articles randomly around their first author, which makes it easy to spot the scientific communities by colour. Lastly, we include papers as nodes and co-authorship edges are connected through them during the network layout in Figure 3.3c. This produces a very

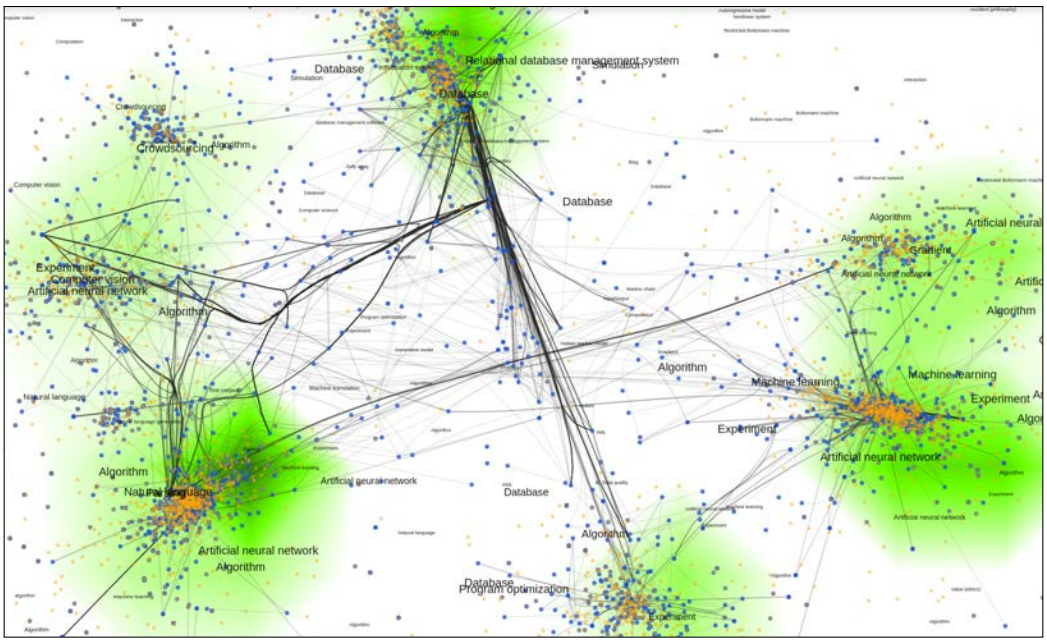


Figure 3.4.: MODIR visualisation of Semantic Scholar (S2), all six communities become clear. Authors are blue dots, papers are orange dots, green density map is based on all papers, black opaque edges connect co-authors.

clean looking layout compared with the other baselines, however papers lump together and are not evenly distributed. Furthermore, semantic nuances between papers are mostly lost which becomes most apparent in the now separated database clusters. Also, the semantic overlap between the ML and NLP communities is not noticeable.

Figure 3.3d positions documents using tSNE and places researchers using Equation 3.1. We see that articles are positioned on the landscape so that research areas are distinctly recognisable by colour. Papers that could not be assigned to a specific area are scattered across the entire landscape. The collaboration network is laid out surprisingly good. The research interests of the authors are coherent between the network and the document landscape, it even shows the close relation between NLP and ML, while showing a clear separation to database related topics. Nonetheless, the network should be loosened for better readability, for example members of the same research group who frequently co-author papers tend to collide.

Unconnected authors are almost not visible as they drift toward densely populated areas in the middle. In Figure 3.3e, we included authors as virtual documents as the sum of their papers during the tSNE reduction. This shows some improvement, as the network layout is more loose and fewer edges overlap and the issue with collapsing research groups is also

mostly mitigated. The semantic overlap of ML and NLP is nicely captured along with the difference to the database papers. However, the network is not clearly readable.

With MODIR, the three research communities become clearly distinguishable, both in the graph layer and in the document landscape. Nodes of well connected communities are close together, yet are not too close locally, and separate spatially from other communities. The document landscape is laid out more clearly, as papers from different fields are grouped to mostly distinct clusters. Obviously there is still a slight overlap as a result of semantic similarities. As previously pointed out, this visualisation also correctly reveals, that the ML and NLP communities are more closely related to each other (both use machine learning) than to DB. The authorship of documents however can only be conveyed through interaction, so this information is not present in the static visualisations shown here. Based on these results we argue, that the network information improves the (visual) community detection. The document embeddings of articles can only reflect the semantic similarities, which may overlap. In conjunction with information from the co-authorship network, the underlying embeddings are put into their context and thus are more meaningful in a joint visualisation.

Further, we provide additional visualisations of our algorithms with more data. Figure 3.4 shows the academic corpus (S2), from which we selected all papers from co-authors around high-impact authors from six research communities as described above. We see how the network information influences the landscape and communities become clearly visible. Although the global structure of both semantics and network is readable, an additional objective to discourage overlapping edges could further improve the result. For better interpretability we used a baseline approach to extract position-based keyphrases to overlay them on the landscape. Our prototype for exploring such a visualisation, which we will describer a the end of this chapter, offers the user basic interactions to explore the landscape by zooming, panning, highlighting parts of the landscape where a search term appears, or looking up entities and categories (if available).

3.3.5. Runtime Analysis

Our proposed dimensionality reduction algorithm uses gradient descend to update the parameters of the projection by minimising three objectives. Therefore, we want to make sure that the optimisation is able to converge over the course of iterations. As all objectives aim to either minimise or maximise the Euclidean distance between relevant data points, they each individually provide gradients that ensure convergence. The overall objective is formed by their weighted sum and thus has no significant influence on the gradient space.

With MODIR it is our goal to provide a scalable approach that can reduce large sets of high-dimensional data. In their paper, the authors state that tSNE becomes infeasible to

Table 3.7.: Runtime of tSNE, PCA, and MODIR in minutes for all datasets at different size. MODIR is executed with (1) and without (2) Objective 3.

Dataset	PCA	tSNE	MODIR (1)	MODIR (2)
AMiner (50%)	**0:01**	16:21	6:57	7:24
S2 (50%)	**0:02**	134:22	10:37	11:51
Enron (50%)	**0:03**	113:35	15:02	18:16
AMiner	**0:02**	48:19	24:45	25:43
S2	**0:06**	392:32	74:29	78:28
Enron	**0:07**	345:38	83:20	87:12

apply to datasets with more that 10,000 samples and thus introduced a number of optimisations, namely Barnes-Hut and pre-reducing the data with PCA, which we use in our the experiments [121]. We apply PCA, tSNE, and MODIR to the complete datasets as well as reduced subsets for which we use stratified sampling if labels are available. All experiments are run on a machine with 256GB RAM and a Intel Xeon E7 Octa-Core with 2.67 GHz. For a fair comparison, we run our algorithm with and without the network based objective and do not use GPU acceleration although it would be possible. We repeat each experiment five times and show the averaged results. It is important to note, that the runtime of tSNE and MODIR is slightly influenced by the perplexity or context size. For example, we ran tSNE with different perplexity settings and saw differences of up to 6 minutes for a perplexity of 5 to 100 on the full AMiner dataset.

Table 3.7 lists the results of our runtime experiments. PCA clearly outperforms all other approaches by far, which is not too surprising, given that it only needs a few matrix operations for its deterministic mathematical model. On the other hand, tSNE and MODIR are based on iterative optimisation. tSNE optimises the Kulbach-Leibler divergence of neighbourhood probability distributions, for which the pairwise distances in the low-dimensional space have to be calculated at each optimisation step. Thus, the execution time rapidly increases as the dataset grows, even with optimisations to reduce the number of comparisons. In MODIR, the iterative optimisation scales almost linearly with the number of documents. For example, an iteration for the full AMiner dataset takes around 23s. The expensive computation of all pairwise distances is only done once during pre-processing and during iterations only selectively based on the context. Comparing the runtime of MODIR with and without Objective 3, we see only slight differences. In contrast to the number of items in the dataset, the small additional context is not significant and the already calculated pairwise distances can be reused. It is important to note, that we trade off runtime complexity with space complexity. Using mathematical frameworks however, the data can be stored very efficiently and is only slightly larger than the input size.

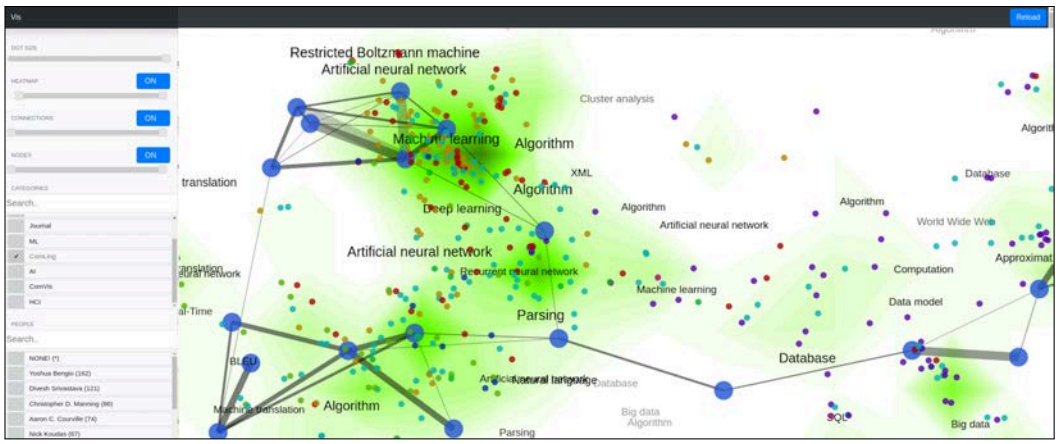


Figure 3.5.: Screenshot of the MODIR interface prototype showing an excerpt of a citation network.

3.4. User Interface

In data exploration scenarios, the goal of getting an overview of the datasets at hand is insurmountable with current tools. The sheer amount of data prohibits simple visualisations of networks or meaningful keyword-driven summaries of the textual content. Data-driven journalism [39] often has to deal with leaked, unstructured, very heterogeneous data, e.g. in the context of the Panama Papers, where journalists needed to untangle and order huge amounts of information, search entities, and visualise found patterns [31]. Similar datasets are of interest in the context of computational forensics [61]. Auditing firms and law enforcement need to sift through huge amounts of data to gather evidence of criminal activity, often involving communication networks and documents [93]. Users investigating such data want to be able to quickly gain an overview of its entirety, since the large amount of heterogeneous data renders experts' investigations by hand infeasible.

In this section, we illustrate the effectiveness of our joint visualisation approach with an exploration interface for different real world datasets. Special graph databases enable the efficient storage of large relationship networks and provide interfaces to query or analyse the data. However, without prior knowledge, it is practically impossible to gain an overview or quick insights into global network structures. Although traditional node-link visualisations of a graph can provide this overview, all semantic information from associated textual content is lost completely. MODIR combines network layouts with dimensionality reduction of high-dimensional semantic embedding spaces. Giving an overview over latent structures and topics in one visualisation may significantly improve the exploration of a corpus by users unfamiliar with the domain and terminology. Figure 3.5 shows a screenshot of

our interactive prototype implementation combining graph and content information.

Social networks are commonly visualised as node-link drawings, where people are shown as circles of varying size based on a weight metric and connections between them as lines connecting the circles. The layout of nodes should visually convey the inherent structure of the network graph. Beside the network, we also visualise associated textual content (documents, such as posts, emails, papers, etc.). Similar to Cartograph [188], we base the visualisation on a *document landscape*. Salient structures of the text corpus become visible in the form of more densely populated regions. To align the network and documents, the graph layout is adjusted to place the circle for a node near the documents associated with it. We focus on integrating all three principles into a single joint visualisation.

Users exploring such data, e.g. journalists investigating leaked data or young scientists starting research in an unfamiliar field, need to be able to interact with the visualisation. Our prototype allows users to explore the generated landscape as a digital map with zooming and panning. The user can select from categories or entities to shift the focus, which highlights characterising keywords and adjusts a heatmap based on the density of points to only consider related documents. We extract region-specific keywords and place them on top of the landscape. This way, the meaning of an area becomes clear and supports fast navigation. As users zoom in, more keywords appear for the region in focus.

3.5. Conclusion

In this chapter we discussed how to visualise large document collections by jointly visualising text and network aspects on a single canvas. To this end, we identified three principles that should be balanced by a visualisation algorithm. From those we derived formal objectives that are used by a gradient descend algorithm. We have shown how to use that to generate landscapes which consist of a base-layer, where the embedded unstructured texts are positioned such that their closeness in the *document landscape* reflects semantic similarity. Secondly, the landscape consists of a *graph layer* onto which the inherent network is drawn such that well connected nodes are close to one another. Lastly, both aspects can be balanced so that nodes are close to the documents they are associated with while preserving the graph-induced neighbourhood. We proposed MODIR, a novel multi-objective dimensionality reduction algorithm which iteratively optimises the document and network layout to generate insightful visualisations using the objectives mentioned above. In comparison with baseline approaches, this multi-objective approach provided best balanced overall results as measured by various metrics. In particular, we have shown that MODIR outperforms state-of-the-art algorithms, such as tSNE. We also implemented an initial prototype for an intuitive and interactive exploration of multiple datasets [164]. One such example is shown in Figure 3.4 We have shown the effectiveness of MODIR using a

number of different large document collections by measuring the topical clustering quality of the document landscape and the network layout of the graph layer. Additionally we used different visualisations to inspect calculated layouts.

While our prototype of MODIR allows basic interactions, we look into improving the look-and-feel further in future work. For easy interpretability and fast exploration we found it useful to have an overlay of keywords. These help to semantically distinguish different areas of the landscape. In our preliminary work we used tf-idf on meta-documents, for which we concatenate actual documents. The simplest approach aggregates documents within a cell of a virtual grid across the landscape. Our more advanced approach, as used in the example shown above, uses density based clustering to group documents. Furthermore, we used established keyphrase extraction algorithms instead of selecting words with the highest tf-idf score. However, in all our experiments we see room for improvement as words seem repetitive or not relevant enough. In future work we hope to focus on the problem of selection and placement of descriptive keywords or keyphrases. This will improve the way users are able to navigate the landscape.

Chapter 4

Robust Visualisation of Diachronic Text Collections

Map-like visualisations are supposed to provide intuitive ways to explore large document collections. State-of-the-art approaches usually transform high-dimensional representations of documents into two-dimensional vectors using dimensionality reduction algorithms. These vectors are then placed into a landscape while retaining semantic information regarding similarity from the high-dimensional representation. Most state-of-the-art dimensionality reduction algorithms were developed with static collections in mind. However, many "real-world" document collections, such as news articles, scientific literature, patents, Wikipedia, or tweets, to name a few, grow and evolve over time. Visualising the temporal change of these collections poses various challenges for out-of-the-box dimensionality reduction algorithms.

Problem Statement The term "dynamic" or "changing" dataset can be interpreted in two ways: (1) The representation of the data changes, or (2) new items are added to the dataset, while existing ones remain unchanged or are removed. In this chapter, we focus on the latter case. There are different approaches for visualising dynamic datasets in two-dimensions. If the full dataset is available, the dimensionality reduction algorithms could be applied to the entire dataset. In this case, the dynamic aspect would have to be part of the way the data is rendered, for example by hiding items outside a defined interval. However, if new data is to be added later, the layout would have to be recomputed, potentially rearranging everything users have grown accustomed to.

Contributions In this chapter, we propose strategies to adapt state-of-the-art dimensionality reduction algorithms to update existing layouts of a dataset. As the dataset grows, new layouts can be computed, which are based on previous versions of the dataset. These strategies ensure that landscapes at different intervals of the collection are robust with regard to spatio-temporal coherence. Furthermore, we propose metrics to measure the stability over time and compare several popular dimensionality reduction algorithms.

4.1. Introduction

Interaction with large document collections can take many forms. Keyword-based search interfaces are certainly the most popular among them. However, in more exploratory settings, users may find a 2-dimensional document landscape more compelling. They enable the user to visually explore a corpus in its entirety. There are several map-like visualisations for book corpora[1] [102] as well as philosophy [143] and physics literature [2]. These show the effectiveness of displaying individual documents in their global context, allowing the user to gain insights that would otherwise remain hidden. We focus on the aspect *where* a document should be placed on such a landscape, especially when the underlying document collection is growing. Most commonly, dimensionality reduction is applied to a high-dimensional representation of the original documents in the corpus. The most popular among them are tSNE and UMAP [121, 124]. However, these algorithms were developed with static data in mind. As the corpus grows over time, for example as new research is published or news events unfold, the landscape has to be updated and should be coherent with earlier versions so that users can reliably associate semantics to specific regions in their mental model of the data.

Dimensionality reduction algorithms typically start with an initial layout of the data in a two dimensional target space. This may either be an approximate layout, often using principle component analysis (PCA) or spectral embeddings, or sometimes a random distribution. The initialisation has a significant impact on the final layout after optimisation [153]. In this work, we compare several initialisation strategies in order to make landscape updates robust to undesired effects. Furthermore, we propose quantitative evaluation metrics to measure stability and layout quality.

Prior work has shown that consistent updates are possible under specific circumstances. Rauber et al. [158] propose a dynamic tSNE approach by introducing a displacement penalty between layouts. However, their focus was change in the representation of individual items in the dataset, not a growing corpus. LION-tSNE adds new data to the initial layout by calculating the position using k nearest neighbours of the original data [24, 46]. Poličar et al. [153] extended tSNE by introducing batch processing to reduce the dimensionality of large datasets.

While batch processing assumes a relatively uniform distribution of all aspects of the data across batches, we assume the opposite. New topics may emerge over time and need to be fitted to earlier representations but may also require slight displacement of older data to make room. Similar issues are also considered in representation learning for dynamic word embeddings [13] or editable neural networks [192]. In both cases, unsupervised

[1] `http://galaxy.opensyllabus.org/`
[2] `https://github.com/ds3-nyu-archive/ds-dialect-map-ui`

learning models are adapted with new objectives or to fit new data. In these scenarios, only pairwise relations need to be stable. For visualisation purposes however, the global rotation or distortion in landscape updates has to be limited.

There are several application that use dimensionality reduction to visualise document collections. Early work includes InVis [90, 147] and the work by Chen et al. [36] who used probabilistic multidimensional projection models. Schmidt [186] use stable random projections to visualise 15-million books from the Hathi Trust collection in a single explorable scatterplot. Systems like Kyrix-S could augment this visualisation with the capability to aggregate information on lasso-selected data or query data [199]. For expert systems, certain characteristics of the layout of a landscape may be required. Pezzotti et al. [151] proposed an algorithm to steer the dimensionality reduction algorithm during the iterative optimisation process.

In the following sections of this chapter, we define strategies for initialising the layout process. We implemented them for several state-of-the-art dimensionality reduction algorithms. Finally, we evaluate the strategies using novel layout quality metrics and close with a qualitative discussion.

4.2. Robustness Through Initialisation

The initialisation has a significant impact on the final result of a dimensionality reduction algorithm. Thus, we hypothesise, that targeted initialisation strategies across visualisations for different intervals of a document collection leads to spatially coherent results. Obviously, some change to the position of earlier documents may be required to make room for emerging topics. This change however should be minimal. Furthermore, emerging topics should fit within the semantic structure of an existing layout. The development of new dimensionality reduction models is beyond the scope of this preliminary work. Thus, we only consider small adaptations of existing algorithms as baselines. For our experiments, we assume that the document collection is strictly growing and that the number of cumulated documents from earlier intervals to be larger than the number of documents in the next interval. To achieve coherence across intervals, we propose the following initialisation strategies.

The *naïve approach* is to use the cumulated documents and the new documents as input to the dimensionality reduction algorithm and only fix parameters that could influence the layout, including the random state. Spectral embeddings or principle component analysis, commonly used for initialisation, both are deterministic when applied to the same data. In case the number of newly added documents is small enough compared to the number of already present documents, this may be a valid approach.

The *kNN approach* uses the layout of earlier documents to approximately place new documents within the existing layout. For each new document, we take k nearest neighbours in the high-dimensional space of earlier documents and calculate their average position in the two-dimensional space to place the new document. Alternatively the impact of outliers could be reduced by using the median or weighting the average by the inverse distances. We initialise the dimensionality reduction algorithms with these positions to optimise the two-dimensional layout.

Lastly, we also use *algorithm-specific approaches*. For example, UMAP [124] and Parametric-UMAP [177] provide a method to project previously unseen documents to the target space. Initially, this works similar to the kNN approach, but an internal encoder model refines the positions of new documents. OpenTSNE [154] implements advanced nearest neighbour methods to calculate affinities between data points in the high-dimensional space. These models can also be used as an initialisation in addition to the previously described kNN approach.

4.3. Evaluation

In order to evaluate the different initialisation strategies described above, we implemented an abstract interface to implement all strategies for all algorithms. Prior work has shown that most dimensionality reduction algorithms are able to project previously unseen data into the two-dimensional space in one way or another. This only works when the new data is sampled from a similar distribution as the already seen data. This is obviously rarely the case for real-world document collections. We therefore setup an experiment to isolate one particular scenario, namely the emergence of a new topic over time. To this end, we simulate the dynamics of a growing document collection. We use the 20-newsgroup dataset [108] and completely hide documents from one category in the initial interval. Over time, only documents from the previously hidden category are added. This way, we simulate an emerging new aspect or topic in the document collection. The experimental results are averaged across multiple runs with different initially hidden categories.

The 20-newsgroup dataset contains around 18,000 texts assigned to one of 20 categories. We represent texts using 10,000-dimensional, tf.idf-weighted bag-of-words vectors. In interval one, we consider all documents from 19 categories but ignore all documents from the remaining category. For the second interval, we add 50 documents from the initially ignored category, for the third we add 200 more, and for the fourth we add the remaining documents from the class up to 1000 documents. This way, we simulate slow and rapid growth of the emerging topic over intervals and get an approximately uniform distribution over the categories.

We implemented common programming interfaces for the most popular algorithms FIt-SNE [115], OpenTSNE [154], UMAP [124], Parametric-UMAP [177], and LargeVis [197]. The code is available for reproducibility on GitHub.[3]

Evaluation Metrics In this work, we propose strategies to adapt dimensionality reduction algorithms to produce coherent representations for document collections that grow and semantically evolve over time. An intuitively usable series of visualisations of the document collection should be relatively stable, such that an area on the landscape is always associated with a fixed semantic meaning. Furthermore, documents that existed in earlier versions of the landscape should remain near their original position for best usability. However, some displacement may be necessary to make space for emerging topics.

To evaluate the effectiveness of the proposed initialisation strategies with regard to these expectations, we measure 8 characteristics of the produced series of landscapes. Thereby we evaluate both, the landscapes themselves and the coherence within a series of landscapes. We base the individual evaluations on the work by Li et al. [114]. Among others, they used (1) the local Kullback-Leibler divergence (*L-KL*) to compare pairwise local distances in the original space and the landscape (lower is better), (2) a continuity score (*Cont*) which measures the overlap of point-wise k-neighbourhoods in both spaces, original and landscape (higher is better), (3) an accuracy score (*Acc*) that calculates how well the labels in the k-neighbourhood predict the label of each document on the landscape (higher is better), (4) and the trustworthiness (*Trust*) by calculating the number of points that appear in the k-neighbourhoods of the original space but not in the same neighbourhood on the landscape (higher is better).

However, these metrics, as well as others they used, only measure how well the landscape represents the high-dimensional space. Since our documents have category labels, we can also measure quality of their visual separation. To incorporate label information in the evaluation, we use (5) the average normalised mutual information (*NMI*) (higher is better), (6) the *Spread* of labels across the landscape (lower is better), (7) and the *Overlap* of Gaussian kernels that are fit to documents of each category (lower is better).

Furthermore, we measure the coherence in a series of landscapes by (8) the average normalised displacement (*Disp*) of data points from one landscape to the next (lower is better), (9) and the average overlap of Gaussian kernels (*Stab*) for each category between two landscapes to show how stable semantic areas are (higher is better).

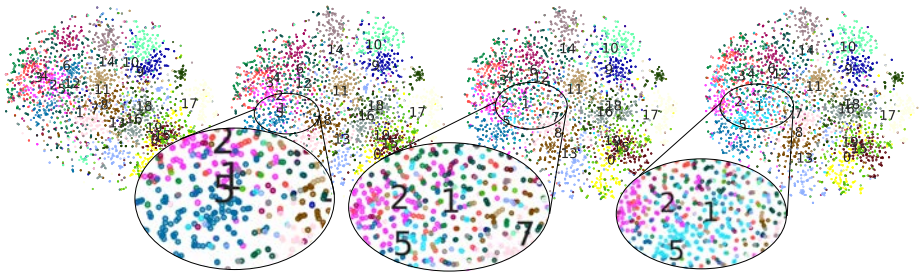
Results In the following, we provide an overview of the results from our experiments. Table 4.1 contains the results of our quantitative evaluation. Due to space constraints, they

[3] `https://github.com/TimRepke/adaptive-landscape`

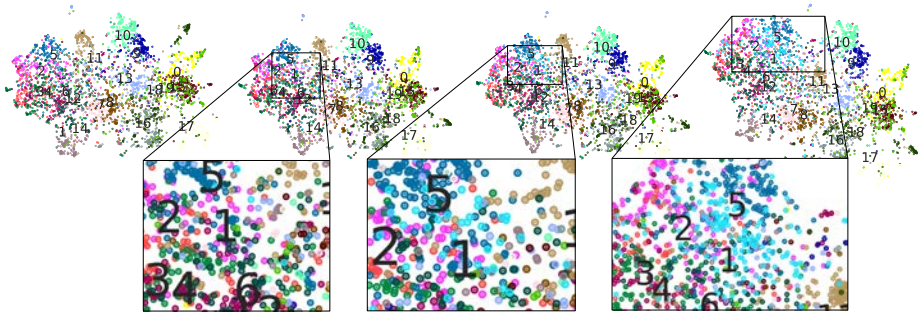
Table 4.1.: Stability and Layout Quality Metrics, best score in **bold**, second best in *italics*

Algorithm	Strategy	L-KL	Acc	Cont	Trust	NMI	Spread	Overlap	Disp	Stab
OpenTSNE	naive	0.05	0.65	0.90	*0.92*	0.52	0.27	0.61	0.31	*0.74*
OpenTSNE	specific	*0.05*	**0.66**	0.91	**0.92**	**0.53**	0.28	0.64	*0.07*	0.86
FItSNE	naive	0.05	0.65	0.91	0.91	0.52	0.28	0.62	0.20	0.79
FItSNE	kNN	0.05	*0.65*	0.91	0.91	*0.52*	0.27	0.59	0.09	**0.86**
UMAP	naive	0.05	0.55	**0.92**	0.82	0.45	*0.19*	*0.17*	0.13	0.58
UMAP	specific	0.05	0.56	*0.92*	0.82	0.45	0.19	0.18	0.12	0.49
ParaUMAP	naive	0.07	0.42	0.87	0.72	0.36	**0.18**	*0.17*	0.33	0.26
ParaUMAP	specific	0.06	0.45	0.87	0.74	0.37	0.20	**0.16**	**0.05**	0.85
LargeVis	naive	**0.04**	0.64	0.91	0.91	0.52	0.23	0.52	0.44	0.49

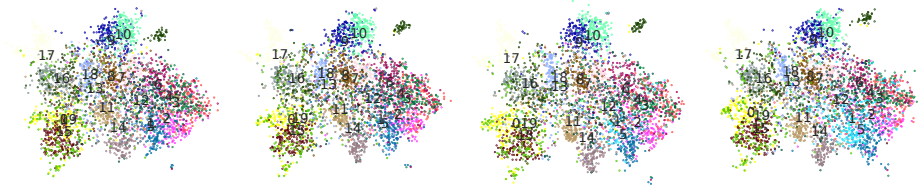
are limited to the 20-newsgroup dataset and only provide a comparison of the naive approach for each dimensionality reduction algorithm and its best performing initialisation strategy. Based on these results and those not shown, none of the approaches and initialisation strategies appear to stand out. Comparing the tSNE-based and UMAP-based algorithms, we observe a trade-off in the results. UMAP-based approaches generally produce more stable layouts across intervals, as seen in the last four columns of the table. However, their ability to represent the original space decreases as a new category grows, which is not well separated from other documents in the existing landscapes. This becomes most apparent in the average NMI scores. Vice versa, tSNE-based approaches generally seem to incorporate documents from the emerging category better, while generally leading to less stable layouts, which is especially obvious when looking at all initialisation strategies. We also observed, that parameters for degrees of freedom or cluster separation appear to worsen all scores. We assume, that the separation amplifies the impact of documents, that appear at less ideal positions, given their category labels. Examples in Figure 4.1 were chosen based on most stable layouts, each corresponding to the non-naive strategy row in Table 4.1. These examples show, how documents from an emerging computer-related topic is embedded into the landscape around already existing similar topics. For the visualisation, we reduced the number of documents but still first add few documents from the initially hidden category and increase the growth rate. All algorithms lead to the least stability of the overall layout between the third and fourth interval. We can also see, that UMAP-based approaches tend to produce a more densely populated landscape than tSNE-based approaches.



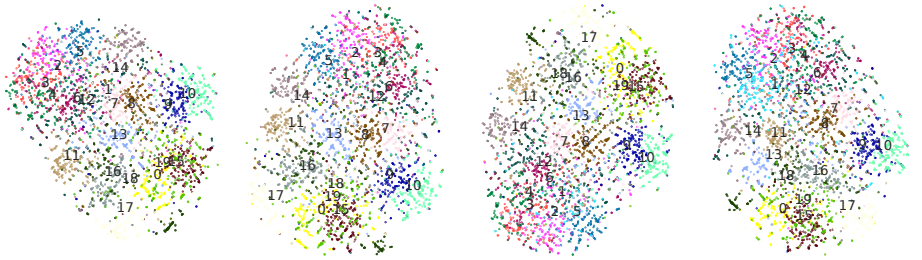
(a) OpenTSNE with perplexity based affinity initialisation strategy



(b) UMAP with with algorithm specific transformer strategy



(c) Parametric-UMAP with algorithm specific transformer strategy



(d) LargeVis with naïve strategy (no cross-interval initialisation)

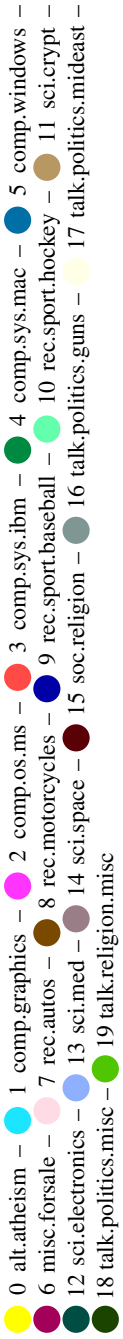


Figure 4.1.: Scatterplots of selected results over four time intervals; 400 posts from each category; category "1 – comp.graphics" is hidden first and then growing over time by first adding 10, then 50, and finally 140 posts in later intervals

4.4. Conclusion

In this chapter we have shown the results of our comparison of different dimensionality reduction algorithms to visualising document collections that grow and semantically evolve over time. To this end, we use the analogy of document landscapes for intuitive exploration of the global and local structure of the underlying document collection. These document landscapes need to be updated to incorporate new documents of the growing corpus. After each update, or even across several updates, users should still be able to recognise a familiar global placement of semantic regions. We compared several state-of-the-art dimensionality reduction algorithms and the influence of different initialisation strategies to achieve stable updates across intervals. We have shown that even simple strategies already improve these objectives over a naive approach. However, our preliminary findings also show that there is still room for improvement. For example, documents on emerging topics do not fit well in the overall structure and are hard to distinguish from older documents.

In future work, we will focus on that issue in particular and develop a cross-interval loss, so that the optimisation process does not solely rely on a good initialisation. Furthermore, another interesting direction for future work is to update underlying document embedding models instead of only using a static vector space representation of the documents.

Chapter 5

COMPUTER-ASSISTED CURATION OF MAP-LIKE VISUALISATIONS

Large and heterogeneous datasets can be overwhelming and require appropriate tools for exploration. In the previous chapters of this thesis, we have shown that computer-generated maps can help to get an overview and explore the data. Given high-dimensional vector representations of the data, dimensionality reduction algorithms are used to create two-dimensional map-like visualisations. These algorithms compute similarity neighbourhoods within the vector representations and replicate them on a two-dimensional canvas.

Problem Statement However, the computed layout of the data may not adhere to the expectation of domain experts. For example, to get an overview over large, heterogeneous document collections, different semantic aspects need to be considered. A financial expert may want the layout to reflect different aspects of the data as opposed to an environmental expert. Therefore, users need the ability to edit the two-dimensional projection by implicitly telling the layout algorithm which aspects to focus on.

Contributions To this end, we propose the novel approach EDIMAP, which enables users to interactively edit the projection. By dragging only a few points on the canvas, the user can steer the layout of the dimensionality reduction algorithm to better reflect the expected interpretation of the underlying data. To do so, we propagate the updated position and similarity neighbourhood back to the high-dimensional representation and update the layout. We also propose a taxonomy of intents a user may have when editing the layout. Furthermore, we propose an improvement to the related ISP algorithm. Finally, we demonstrate the effectiveness and robustness of EDIMAP on a series of real world datasets and show significant improvement over related work.

5.1. Introduction

Substantial amounts of data are produced in our modern information society each day. In order to get an overview and explore large and heterogeneous datasets, suitable semantic representations are needed. Map-like visualisations of a dataset provide an intuitive way to interactively explore datasets of all kinds [78]. However, the layout of the data may sometimes not adhere to the expectations of users and domain experts. For example, assume there is a dataset of textual data, more specifically business reports. A financial expert may want to group the reports by industry sectors, whereas an environmental expert may prioritise geographical and technological aspects. Therefore, we propose EDIMAP, an algorithm that enables interactive few-shot editing of map layouts to better adhere to a user's mental model. We also introduce a taxonomy of edit intents, which describes all fundamental scenarios users may encounter during that interactive process.

There are many ways to generate a layout for the map of a dataset. For example, graph drawing algorithms [67] can be used if data contains network information. Alternatively, items from the dataset may be represented as high-dimensional vectors, for example as numeric representations of text, pixel values of images, or multi-variate data. Deep learning methods are a popular alternative, as they provide a means to embed data into semantic spaces, where each item of the dataset is represented by a high-dimensional vector. Semantically similar items reside closer to one another in this space. For the scope of this chapter, we use dimensionality reduction to project these representations into a two-dimensional space for visualisation. We call this two-dimensional projection *map layout*. On this map, each item from the dataset is rendered as a point on the two-dimensional canvas, which are positioned in such a way, that similar items are near one another. The similarity is determined using a distance metric between their respective high-dimensional vector representations. Modern dimensionality reduction algorithms such as UMAP [124] are designed to faithfully preserve the local similarities between vectors in the high-dimensional space in the learnt two-dimensional projection. Although the resulting two-dimensional visualisations help to get a coarse-grained overview of the data, the reduced dimensionality inevitably coincides with a loss of information [112]. In particular, latent global structures may not be preserved. From a semantic point of view, different dimensions in the high-dimensional space carry different information regarding similarity. Especially in embedding models of textual corpora, there are many ambiguities and overlapping word senses [10] as well as semantic and syntactic subspaces [160]. Depending on the use-case, the underlying data and its similarity may be interpreted in different ways. However, dimensionality reduction algorithms treat all dimensions equally, aiming for an unbiased, purely data-driven reduction result.

In this chapter, we aim to incorporate user feedback during the layout process. Endert et al. [54] discussed interaction patterns for semantic landscapes. In their work, they also

proposed a framework of updating a force-based layout of the data. Spathis et al. [193] use a very similar framework. They however use a neural network to first replicate a reference layout provided by an arbitrary dimensionality reduction algorithm. Edits made by a user are then used to update the model. Both these approaches are limited to handle only very small datasets. Contrary to directly editing the landscape, Johansson and Johansson [89] proposed a dimensionality reduction algorithm that can be influenced by combinations of quality metrics, while others aim to steer the dimensionality reduction algorithm during the iterative optimisation process [151]. However, their goal is to optimise visualisations of multi-variate data reduced to more than two dimensions. In either of these setups, a fundamental requirement is the interpretability of the resulting visualisation as discussed by Ding et al. [48]. Furthermore, Lespinats and Aupetit [112] raised the question, whether it is even possible to find faithful two-dimensional mappings of the originally high-dimensional data. Bian and North [19] avoid this issue by updating the input data itself. Each edit done by a user in the two-dimensional visualisation produced by multi-dimensional scaling of a pre-trained BERT model [45] is propagated back to update the model's last layer.

In most real-world scenarios with complex data, domain experts already have an idea of which dimensions of the data are more important than others and thus introduce bias by incorporating their expert knowledge. In the case of dimensionality reduction, domain experts can explore the resulting map layout and suggest edits by dragging data points to different locations to better fit their mental model of the data and its similarity. Doing so by for each point individually quickly becomes infeasible for larger datasets. To this end, we propose EDIMAP[1] to augment the interactive editing process. We assume, that behind each single edit,[2] there is a larger intent by the user that we need to capture, for example moving all similar points to a different location on the map. To this end, we propose a taxonomy of possible edit intents, for which we identified three fundamental intents: *Separate*, *Merge*, and *Arrange*, which we describe in more detail in Section 5.2. Given only a few suggested edits by the user, EDIMAP is able to update the layout according to the underlying intent while also preserving the overall arrangement where possible. This has the advantage that single edits are augmented to reduce the necessary manual effort to create a meaningful map layout for a specific use-case. Furthermore, the initial layout is not completely rearranged, which would otherwise disturb areas the user purposefully did not edit. We are able to achieve that by extending UMAP to incorporate feedback on an existing layout. To do so, we use the prior and posterior similarity neighbourhoods to update an underlying similarity graph, which is used to optimise the two-dimensional layout.

For the scope of this chapter, we assume that high-dimensional vector representations of the data are given. The initial two-dimensional layout for the map-like visualisation of a

[1] EDIMAP stands for editable UMAP, or editable manifold approximation and projection.
[2] We define an *edit* to be the action of dragging a single point on the map to a new location.

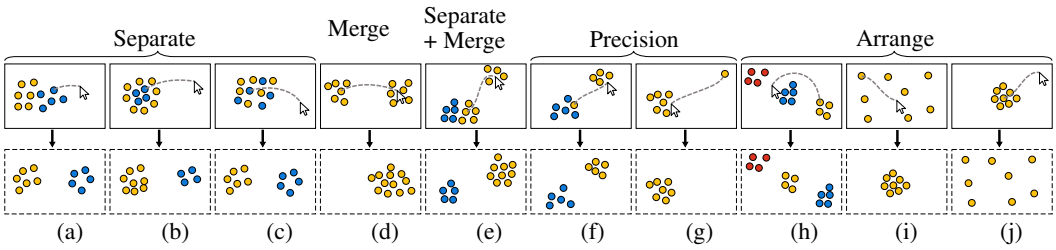


Figure 5.1.: Different scenarios from the taxonomy of edit intents as abstract illustrations.

dataset is computed using UMAP. With the help of our interface, users can suggest edits to this initial layout. Based on only a few of these edits, we update the layout according to the user's intent. This process can be described as a few-show learning algorithm to interactively transform two-dimensional representations of data [204]. We demonstrate our approach using several real-world datasets, namely the textual 20 newsgroups dataset [108], image datasets like MNIST [110], FashionMNIST [209] and MNIST-1D [68], as well as the multi-variate seeds dataset [33].

5.2. A Taxonomy of Edit Intents

Map-like visualisations are powerful tools for exploring large datasets. Typically, the layout for such a map is computed using an unsupervised representation learning approach and dimensionality reduction. After initial exploration of a newly computed layout, a user may wish to change the layout better fit their mental model of the data. In this section, we propose a taxonomy of edit intents to categorise possible objectives a user may have when editing a layout. Later in this chapter, we also introduce our prototype for a user interface that allows interactive assisted editing of map layouts.

In order to design a system for interactive dimensionality reduction, it is essential to understand the intent behind a user's edit. Saket et al. [180] have compiled a list of direct manipulation strategies users may perform to edit standard graphical encodings, such as barchart. In this section, we focus on scatterplot visualisations and propose a taxonomy of intents of edits. We define an *edit* to be the action of dragging a single point on the canvas to a new location. There are three fundamental intents, namely *Separate*, *Merge*, and *Arrange*. Figure 5.1 illustrates these three intents in different scenarios (Figure 5.1 a-d, h-j), including derived intents, such as *precision editing* (Figure 5.1 f, g) or *compound intents* (Figure 5.1 e). Very large datasets are typically not fully labelled. We use colours in the illustration only to depict latent aspects that may exist with respect to an intent. Note that the colour-coding reflects only one (among possibly many) aspects in the data and it is not known apriori that this is the relevant aspect for the user.

- *Separate (Figure 5.1 a-c).* A dimensionality reduction algorithm may create visual clusters of points based on manifolds it found in the high-dimensional space. With another (latent) aspect in mind, a user may want to separate these clusters. In the simplest case, points from different aspects are linearly separable (Figure 5.1 a). There are however also cases that are non-linear or inseparable in two dimensions (Figure 5.1 b, c). Even though points are hard to separate in the two-dimensional projection, there might be a latent subspace in the original high-dimensional data in which a linear separation corresponds to the intended edit.
- *Merge (Figure 5.1 d).* Another intent of an edit is to merge two clusters of points. Dimensionality reduction algorithms have to balance preservation of local neighbourhoods and the global structure. Based on the hyperparameter settings, it is quite common to see multiple clusters of points belonging to the same class.
- *Arrange (Figure 5.1 h-j).* As previously mentioned, preserving the global structure can be challenging. Thus, users may intend to rearrange groups of points based on their domain knowledge or a mental map they might have of the data (Figure 5.1 h). Furthermore, they may intent to make cosmetic changes to the layout to squeeze or loosen groups of points (Figure 5.1 i, j).
- *Compound intents (Figure 5.1 e).* Lastly, multiple intents might be combined in a single edit, such as separating a group of points and merging them into another group. As shown by the examples, some of the intents of this taxonomy may be very hard to identify from a single edit. Therefore, a system for interactive dimensionality reduction may need additional information. For example, a model might consider a series of edits or make suggestions on the most likely intents.
- *Precision edits (Figure 5.1 f, g).* The most trivial intent is the relocation of a single point to a different location. This intent would apply to scenarios, where a user wants surgical precision to remove single outliers from a neighbourhood.

5.3. Algorithms for Computer-assisted Layout Editing

The layout for a map-like visualisation is typically computed by reducing the dimensionality of high-dimensional vector representations of all items from the dataset. We assume this set of high-dimensional vector representations $x_i \in X \subset \mathbb{R}^n$ to be given. A dimensionality reduction algorithm, such as principle component analysis (PCA), an autoencoder, tSNE [121], or UMAP [124] is applied to compute a projection of these vectors into a two-dimensional space. These algorithms use a similarity metric, defined by a distance measure $d(x_i, x_j)$ between high-dimensional points, to faithfully preserve their pairwise similarities in the layout. For the scope of this work, we assume that the resulting *initial*

layout already exists. The aforementioned projection $P : \mathbb{R}^n \mapsto \mathbb{R}^2$ thereby maps each item x_i to its two-dimensional counterpart $y_i \in Y$.

Given the *initial layout* of the dataset, a user edits the map by dragging points from their *source location* y_k to their *target location* $\mathring{y}_k$. The proposed algorithm assists this editing process by using these changes and computing an *updated layout* of the dataset. In the remainder of this section we first define a set of objectives the updated layout should adhere to. We then describe how we adapted a baseline model proposed in related work and provide a detailed definition of our approach.

5.3.1. Objectives for Updated Layouts

During the editing of the map-like visualisation of a dataset, a user drags one or more points to their new target location. Our proposed algorithm augments these edits to assist users in their efforts to adapt the layout according to their expectations. Based on the previously introduced taxonomy of of edit intents, we derive the following set of objectives that the updated layout should fulfil.

1. An edited point should be positioned at or near the target location in the updated layout.
2. The position of points in the proximity of the target location shall not differ significantly from the initial layout.
3. Points in the proximity of the source location may also move to a different location if necessary.
4. Changes to the general layout of the map shall be minimal to preserve the user's mental map of the data. This can also be referred to as stability or robustness.

Note, that these objectives may contradict some of the previously defined intents, especially *Arrange* intents. Similarly, any algorithm for interactive dimensionality reduction may not be able to fully adhere to these objectives. To this end, we propose two approaches and compare their applicability in different scenarios.

5.3.2. Neural Embedding Baseline Model (iSP)

We derive our baseline model from related work, and train a neural network to replicate the mapping function. Sainburg et al. [177] have developed a neural network approach which replicates layouts computed by dimensionality reduction algorithms. Alternatively, instead of learning a projection function directly, a neural network can learn the projection function from an existing layout as done by Spathis et al. [193] with their so-called interactive

similarity projections (ISP). Let $\hat{F}$ be a replication of F that should be learnt by the network and N the number of items in the dataset. We adapted the objective function proposed in ISP as shown in Eq. 5.1 to train such a network and added a locality term (second line). To our knowledge, this is the only approach for editing two-dimensional maps of high-dimensional data.

$$\begin{aligned} L = & \frac{1}{2\|M\|_1} \sum_{i \neq j}^{N} [M]_{i,j} ([P]_{i,j} - [T]_{i,j})^2 + \\ & \frac{1}{\|m\|_1} \sum_{i}^{N} m_i \left(\|\hat{F}(x_i) - \mathring{F}(x_i)\|_2^2 / \sigma_d \right) \end{aligned} \tag{5.1}$$

Hereby, M is an $N \times N$ matrix used for masking and weighting pairs of items and $\|M\|_1$ is the l_1 norm. Similarly, m is a vector of length N for masking and weighting items in the dataset and $\|m\|_1$ is its l_1 norm. The first term of the loss sums the differences of the similarity matrices P and T. Entries of each matrix are defined as $\|f(x_i) - f(x_j)\|_2 / \sigma$, where f is the original mapping function F to define T or $\hat{F}$ for P respectively. The normalisation terms σ and σ_d are given by the mean of the according pairwise similarities. Since the first term only preserves similarities, we added the second term which sums the absolute displacement of points between the target layout and its imitation. During the training to replicate the reference layout, $\mathring{F} = F$ and the masks are set to $[M]_{i,j} = 0.5$ for all $i \neq j$ and $m_i = 0.5$ respectively. Once a user drags a point y_k to a target location $\mathring{y}_k$, we update the position in a cloned projection function $\mathring{F}$ of F. The mask is set to $[M]_{i,j} = 0.01$ for all pairs except for neighbours of y_k, for which they are set to one. Neighbours are sampled from the original and new location in the two-dimensional space. The similarity matrix is updated by setting $[T]_{k,l} = 0$ for all neighbours l in the target location of $\mathring{y}_k$. The mask for the displacement term m is set to zero for y_k and its neighbours in the original location. Training of the model is resumed with updated terms in the loss function. For the scope of this work, we use a neural network with a single fully connected layer with $N \times 2$ parameters.

5.3.3. Editable UMAP (ediMAP)

Our proposed algorithm assisting the editing of map layouts is based on the popular UMAP algorithm [124]. The core concept of UMAP is to construct a network of similar items from the dataset to create the two-dimensional layout of the data. It uses a distance metric, typically on high-dimensional vector representations, to calculate similarities, which are represented as a weighted edges in a network of items. A spectral embedding of this network is used to initialise the layout and is then fine-tuned with a force-directed layout algorithm.

Algorithm 1 EDIMAP Algorithm

function UPDATELAYOUT($X, P, \mathring{Y}, k, d, \xi$)
  $\mathscr{G} \leftarrow$SIMGRAPH(X, k, d)
  for all $\mathring{y}_i \in \mathring{Y}$ **do** ▷ For all edited points
    $\mathscr{N}^k_{\mathring{y}_i} \leftarrow kNN(\mathring{y}_i)$ ▷ k nearest neighbours to $\mathring{y}_i$ in $P(X)$ with distance $\|\cdot\|$
    $\overline{x} \leftarrow \sum_{x_i \in \mathscr{N}^k_{\mathring{y}_i}} \frac{x_i}{|\mathscr{N}^k_{\mathring{y}_i}|}$ ▷ Centroid of respective points in X
    $\mathscr{N}^k_{\overline{x}} \leftarrow kNN(\overline{x})$ ▷ k nearest neighbours to $\overline{x}$ in X with distance $d(\cdot)$
    for all $y_j \in \mathscr{N}^k_{\overline{x}}$ **do**
      $w^1_{i,j} \leftarrow \exp(-\max\{0, d(x_i,x_j) - \rho\}/\sigma)$ ▷ Edge weight in X; ρ and σ as in Algorithm 2
      $w^2_{i,j} \leftarrow \exp(-\max\{0, \|\mathring{y}_i - y_j\| - \rho\}/\sigma)$ ▷ Edge weight in $P(X)$; ρ and σ as in Algorithm 2
      Update weight for $(x_i,x_j) \in \mathscr{G}$ to $avg(w^1_{i,j}, w^2_{i,j})$
    $\mathscr{N}^k_{y_i} \leftarrow kNN(y_i)$ ▷ k nearest neighbours to y_i in $P(X)$ with distance $\|\cdot\|$
    for all $y_j \in \mathscr{N}^k_{y_i} \setminus \mathscr{N}^k_{\mathring{y}_i}$ **do** ▷ For all initial neighbours
      Update weight of edge (x_i,x_j) to $\xi w_{i,j}$
      for $l \leftarrow 1,..,|X|$ **do**
        if $l \neq i$ and $(x_j,x_l) \in \mathscr{G}$ **then**
          Update weight of edge (x_j,x_l) to $\xi^2 w_{j,l}$
  OPTIMISELAYOUT($n_{epoch}, \mathscr{G}, X, P$)

We adapt the basics of this concept in EDIMAP. Hereby, the edits to the layout suggested by a user are used to update the similarity network. As shown by related work, the initial placements have a siginificant impact on the resulting layout [101, 168]. Thus, when we continue to update the existing layout and only partially change the underlying similarities, we implicitly assure to comply with Objective 4, stating that the mental map of the data should be preserved.

Our EDIMAP algorithm does not make any assumptions about which dimensionality reduction approach was used to generate the initial layout. Thus, we first need to construct the normalised similarity graph as described in Algorithm 2. The similarities are based on the distance measure $d(x_i,x_j)$ between high-dimensional vectors, each representing their respective item in the dataset. Let $\mathscr{N}^k_{x_i}$ be the set of k nearest neighbours of item x_i. For each $x_j \in \mathscr{N}^k_{x_i}$, we add an edge (x_i,x_j) to the similarity graph. The edge weights are defined as

$$w_{i,j} = \exp(-\max\{0, d(x_i,x_j) - \rho\}/\sigma),$$

where ρ is the distance to the closest neighbour to x_i and σ the distance to the k-th closest neighbour to x_i. Note, that UMAP defines σ to be the smoothed k nearest neighbour dis-

tance. This similarity graph is based on the high-dimensional vector representations, but should also reflect the proximity between points in the initial layout, as most dimensionality reduction algorithms commonly aim to preserve local similarity neighbourhoods. The construction of the similarity graph is defined in Algorithm 2.

Algorithm 2 Construction of the Similarity Graph

function SIMGRAPH(X, k, d) ▷ Construct Similarity Graph
  $\mathscr{G} \leftarrow \emptyset$
  for all $x_i \in X$ **do**
    $\mathscr{N}_{x_i}^k \leftarrow kNN(x_i)$ ▷ k nearest neighbours to x_i in X with distance d
    for all $x_j \in \mathscr{N}_{x_i}^k$ **do**
      $\rho \leftarrow \min(\mathscr{N}_{x_i}^k)$ ▷ Distance to closest neighbour to x_i
      $\sigma \leftarrow \max(\mathscr{N}_{x_i}^k)$ ▷ Distance to k-th neighbour of x_i
      $w_{i,j} \leftarrow \exp(-\max\{0, d(x_i, x_j) - \rho\}/\sigma)$ ▷ Compute edge weight
      $G \leftarrow G \cup \{(x_i, x_j, w_{i,j})\}$ ▷ Add weighted edge to similarity graph
  return $\mathscr{G}$

As a user moves an item x_i from its source location y_i in the initial layout to its target location $\mathring{y}_i$, we update the similarity graph as described in Algorithm 1. First, we determine the k nearest neighbours $\mathscr{N}_{\mathring{y}_i}^k$ of the edited point at the target location $\mathring{y}_i$. We then compute the centroid $\bar{x} \in \mathbb{R}^n$ of those points to heuristically determine the target location in the high-dimensional space. This is used to determine the theoretical actual target neighbourhood $\mathscr{N}_{\bar{x}}^k$. As before, we add edges for each neighbour to the similarity graph and weigh them by their normalised distance. This time, however, we use the average of the normalised distances in the high-dimensional and two-dimensional space. Using only either one space to determine the similarity weight would either neglect the actual similarity in the semantic representation or what the user actually sees while editing the map layout. Furthermore, we update the edge weights of the original neighbourhood of the edited item x_i in the similarity graph as follows. Edges, if they exist, connecting x_i to the k nearest neighbours $x_j \in \mathscr{N}_{y_i}^k$ at the source location y_i, are reduced by the factor $\xi \in (0,1)$. All edges, apart from the aforementioned, connecting any $x_j \in \mathscr{N}_{y_i}^k$ are reduced by the factor of ξ^2. This reduction of edge weights limits the otherwise counteracting forces in the update phase of the layout. Furthermore, it could be exposed in a user interface for editing map layouts as a user defined parameter to influence, how much the points in the source neighbourhood should be moved along with the point that was edited.

Finally, we revise the map layout using a force-directed layout algorithm based on the updated similarity graph as described in Algorithm 3. For each node that was affected by updating the similarity graph, we iteratively update the location of the respective point's position y_i on the map over several epochs. In each epoch of the layout optimisation, the

location is updated to $\tilde{y}_i$ using

$$\tilde{y}_i = y_i - \eta \sum_{(y_i, y_j, w_{i,j}) \in \mathscr{G}} w_{i,j} \frac{(y_i - y_j)}{\|y_i - y_j\|},$$

where η is the learning rate, which decays with each iteration. Typically, force directed layout algorithms require an additional repelling force. However, since we only edit the locations of points affected by the update of the similarity graph and use all the remaining that are connected to these points in the graph as fixed references, we only need the attracting force defined above. Note, that as the number of edited points increases, the update has to be performed in batches. Otherwise, the overall layout would change significantly and the assumption of not needing additional forces is ill-advised.

Algorithm 3 Optimisation of the Layout

function OPTIMISELAYOUT($n_{epoch}, \eta, \mathscr{G}, X, P$)
  for all $\mathring{y}_i \in \mathring{Y}$ **do** ▷ Can also be done in batches
    for $e \leftarrow 1, .., n_{epoch}$ **do**
      for all $y_j \in \mathscr{N}_{\overline{x}}^k \cup \mathscr{N}_{y_i}^k$ **do**
        ▷ Update position of $y_i = P(x_i)$
        $\tilde{y}_i = y_i - \eta \sum_{(y_i, y_j, w_{i,j}) \in \mathscr{G}} w_{i,j} \frac{(y_i - y_j)}{\|y_i - y_j\|}$
  return Updated projection P

In summary, we presented an algorithm that assists users in their effort to edit an existing two-dimensional layout of a dataset. The initial layout is updated with a force-directed layout algorithm based on a graph of similarities, which incorporates these edits. We also described an improved version of a baseline from related work.

5.4. Evaluation

In this section, we apply our EDIMAP algorithm for interactive dimensionality reduction to several real world datasets. We simulate user interactions to measure how well our model can fulfil the expectations and objectives we defined earlier across several different setups. The resulting maps of the datasets are evaluated both quantitatively and qualitatively in a series of experimental setups.

5.4.1. Datasets

For our experiments, we use six datasets with different characteristics: real world datasets with multivariate, image, and text data, as well as an artificial dataset. This includes the

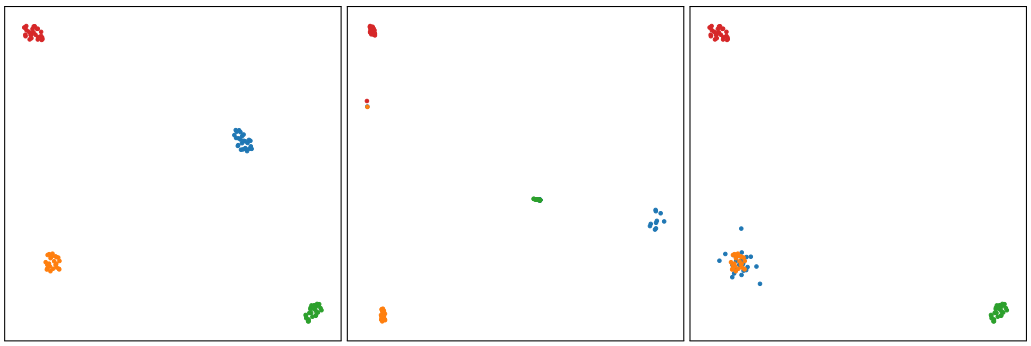
well-known *MNIST*[3] dataset of written digits [110] and the *MNIST-1D*[4] variant, which is derived from the original data but harder to separate [68]. We also use *FashionMNIST*[5], which contains greyscale images of fashion articles like shoes and sweaters across ten different categories [209]. Aside from image data, we also use the multivariate *seeds*[6] dataset [33]. It is comprised of measurements of kernels of three different wheat varieties. In contrast to the other datasets, provides intuitively interpretable dimensions with features like kernel length or weight. Furthermore, we evaluate our approach on textual data using the *20 newsgroups*[7] dataset [108]. We use a tf-idf representation of the messages as well as a doc2vec [131] embedding trained on the entire collection. Since real-world datasets often contain overlapping or ambiguous latent aspects, which makes them difficult to use for evaluation, we also generate an artificial dataset. With this *Blobs* dataset, we can control the latent aspects within the high-dimensional space. We do so by combining normally-distributed clustered points of two two-dimensional spaces. In particular we generate N points around the centroids $(1,1)$ and $(0,0)$ and another set of N points around $(1,0)$ and $(0,1)$. Respective pairs of two-dimensional vectors are concatenated to obtain a set of N four-dimensional points with two hidden subspaces.

As mentioned earlier, both UMAP and EDIMAP require a similarity measure between the semantic high-dimensional vector representations. In the following, we will use the cosine similarity for text datasets and the euclidean distance for the others. The similarities on the two-dimensional map are always defined as the euclidean distance.

5.4.2. Simulated User Interaction

Our interface for editing the map layout, allows the user to directly drag points in order to move them to their target location and thereby suggest updates. Our algorithm assists this editing process by propagating the suggested edit to similar items. For the scope of this work however, we only simulate these user interactions to be able to automatically evaluate several runs and different configurations. As mentioned earlier, the edits of a user always follow a specific intent they might have. The diverse characteristics the datasets in our evaluation require different setups and simulated intents. We simulate the edits by moving one or more points with a specific label towards the centroid of points with another label. In this way, we mimic the user intent of merging two clusters of points. This procedure is repeated for different numbers of edited points and different sets of labels. We include figures of the layouts of all six datasets in their initial layouts as computed by UMAP and the updated layouts by ISP and EDIMAP based on simulated edits as described above.

[3] `http://yann.lecun.com/exdb/mnist/`
[4] `https://github.com/greydanus/mnist1d`
[5] `https://github.com/zalandoresearch/fashion-mnist`
[6] `https://archive.ics.uci.edu/ml/datasets/seeds`
[7] `http://qwone.com/~jason/20Newsgroups/`



(a) Initial Layout (UMAP) (b) ISP (c) EDIMAP

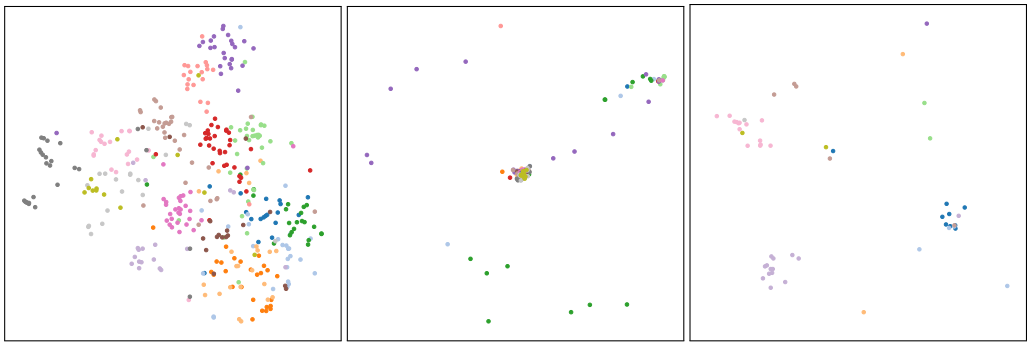
0 – 1 – 2 – 3

Figure 5.2.: Scatterplots of *Blobs* dataset before and after updating.
Edit intent: Merge clusters 0 and 1. *(best viewed in colour)*

For the artificially generated *Blobs* dataset, we simulate the intended merging of the orange and blue clusters as shown in Figure 5.2. In the initial layout, all differently labelled points are clearly separated. While EDIMAP is able to perfectly achieve the goal of merging two clusters without altering the location of any other points, ISP moves points from the unrelated green cluster and shrinks the other clusters.

For the *20 newsgroups* dataset, we simulate edits with the intent to create a map layout of four clusters of messages. These four clusters are based on the respective subtopics of *computer* (`comp.*`, 1-5), *recreational* (`rec.*`, 7-10), *science* (`sci.*`, 11-14), and *talk* (`talk.*`, 16-19). As shown in Figure 5.3, ISP forms a central cluster of points containing all labels. The updated layout by EDIMAP loosely resembles the four intended clusters: *computer* on the bottom right, *talk* on the top left, and most of *science* on the bottom left, and *recreational* on the top right. This example and other experiments we did suggest, that when moving many points at once from different areas of the map, the individual intents are hard to distinguish and many points overlap and are rendered on top of one another. To circumvent this issue, we split the edits across smaller batches and do repeated partial updates. Adding a repelling force may provide further improvement.

For the *seeds* dataset, we aim to show how well our algorithm is able to detect, that the simulated user is trying to sort the seeds by their kernel groove length. Therefore, we simulate movements, which should form two clusters of seeds with a kernel groove length below or above 5.5mm (one mostly orange cluster and one mixed cluster) as shown in Figure 5.4. Since EDIMAP is not focused on intents of edits of that kind, it struggles to update the layout accordingly, in particular because several target locations interfere with a previously occupied area of the the layout. It maps all orange points onto the same location and forms two additional clusters. ISP should be able to better solve this particular task, as



(a) Initial Layout (UMAP) (b) ISP (c) EDIMAP

comp.graphics – comp.os.ms – comp.sys.ibm – comp.sys.mac – comp.windows
rec.autos – rec.motorcycles – rec.sport.baseball – rec.sport.hockey
sci.crypt – sci.electronics – sci.med – sci.space
talk.politics.guns – talk.politics.mideast – talk.politics.misc – talk.religion.misc

Figure 5.3.: Scatterplots of *20 newsgroups* dataset before and after updating.
Edit intent: Form clusters of all `comp.*`, `rec.*`, `sci.*`, and `talk.*` topics.
Some points overlap in the updated layouts.

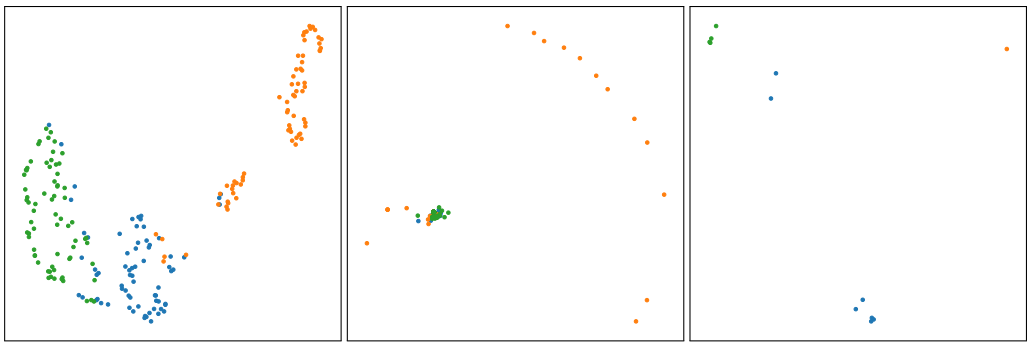
it jointly learns to update the projection function over all points. Although there is a mixed cluster, the remaining orange points are scattered across the entire map.

For the *MNIST* dataset, we simulate a user who intents better separate the 4s, 7s, and 9s as well as the 3s, 5s, and 8s, which form consecutive clusters in the initial UMAP layout as shown in Figure 5.5. As we can see, ISP significantly changes the overall layout and scatters the 4s and 9s all over the map while compressing all other digits to a small area in the middle. EDIMAP on the other hand is able to increase the margins between each cluster of digits. However, since the source location of some of the edited points is right at the separating line between two digits, some points get mixed into the wrong cluster.

Almost all dimensionality reduction algorithms have problems separating the numbers of the *MNIST-1D* dataset, which results in a layout where all numbers are mixed on the layout. To this end, we simulate a user who moves points representing the same number towards one another to clean up the layout. Both algorithms fail to achieve that task.

For the *FashionMNIST* dataset, we aim to merge clusters as shown in Figure 5.6. Of the originally more fine-grained labels, the intent is to form groups of articles, in particular *footwear* (7: sneaker, 9: boot, 5: sandal), tops (0:tshirt, 2: pullover, 6: shirt, 4: coat), and others (1: trouser, 3: dress, 8: bag). As before, ISP forms groups most points in the middle, whereas EDIMAP is able to form the three intended clusters.

Note, that in an actual interface for editing map layouts, the data may not be annotated as



(a) Initial Layout (UMAP) (b) ISP (c) EDIMAP

1 – 2 – 3

Figure 5.4.: Scatterplots of *seeds* dataset before and after updating. Edit intent: Group seeds by kernel groove length. Some points overlap in the updated layouts.

described here. We only used the label information to simulate the edits and for rendering to provide a clear definition of semantically similar groups of items in a dataset. The qualitative evaluation clearly shows the strengths and weaknesses of the approaches. As mentioned before, understanding the intent behind a user's edit is very important to algorithmically assist the editing process. As expected, the results for EDIMAP show, that it performs best on *Merge* and *Separate+Merge* intents. With appropriate tuning of hyperparameters, it could also be used for *Precision* edits. The addition of virtual nodes for target locations could further improve the performance on *Separate* and *Arrange* intents.

5.4.3. Quantitative Results

In this section, we provide a quantitative evaluation of our EDIMAP approach for editing map layouts. Corresponding to the previously defined objectives for the updated layout, we determine

1. DIST: the distance of the edited points between the target location and their location in the updated layout;
2. TARGET: the displacement of points between the initial layout and the updated layout of points near the target location;
3. DTT: the difference between pairwise distances of all points in the source and target cluster, based on the label information, before and after updating the layout; and
4. TOTAL: the total displacement of all points between the two layouts.

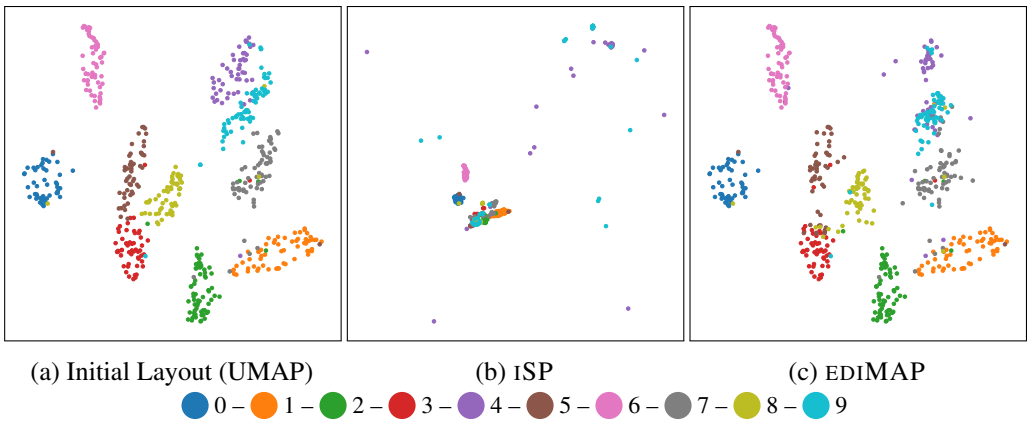


Figure 5.5.: Scatterplots of *MNIST* dataset before and after updating.
Edit intent: Increase margins between 4s, 7s, 9s and 3s, 5s, 8s.

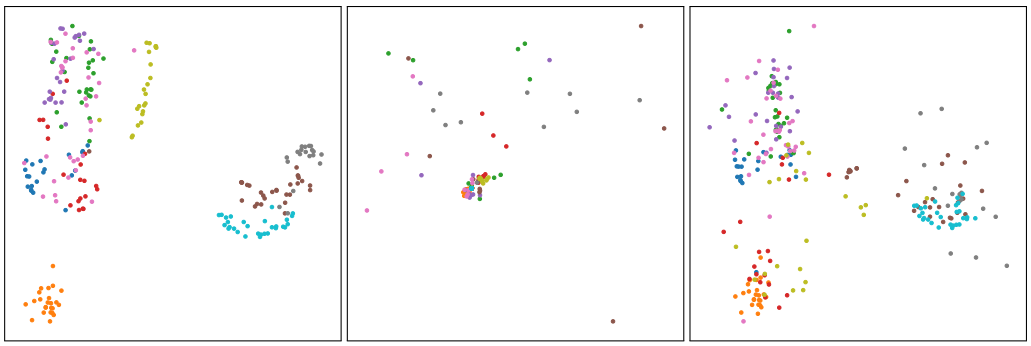
For all metrics, smaller numbers are considered better, DTT should be negative in most cases. However, since the overall purpose of our approach is to update the layout, some movement has to necessarily occur. Thus, we exclude all points from the calculations that were explicitly edited. For these experiments, we move randomly selected points that have the same label towards the centroid of points that have a different label and repeat this process for multiple pairs of labels. All results are averaged over multiple runs and normalised by the relevant total number of points and size of the respective layout. Furthermore, we calculate each metric for an increasing number of edited points to see how much user feedback is required to cause an effect.

All values are normalised to reduce the effect of different sizes of the initial layouts and the varying sizes of the datasets. We list the results for all metrics and configurations after updating the layout with ISP and EDIMAP in Table 5.1. Here we can see, that the layouts updated using EDIMAP generally cause less displacement to the overall layout, as shown by the TOTAL and TARGET metrics. Although ISP uses a mask to minimise the effect it has on the general layout, there is still more movement overall. Both algorithms show a similar performance in the DTT and DIST metrics, however EDIMAP requires less points to be edited to minise these numbers.

Furthermore, we compute metrics commonly used in related work on dimensionality reduction, namely

5. ACCURACY, which is the average score of a k-nearest neighbour classifier;

6. SILHOUETTE COEFFICIENT, which is based on the mean intra-cluster distance and the mean nearest-cluster distance, cluster assignments are given by the labels;



(a) Initial Layout (UMAP) (b) ISP (c) EDIMAP

T-Shirt – Trouser – Pullover – Dress – Coat
Sandal – Shirt – Sneaker – Bag – Boot

Figure 5.6.: Scatterplots of *FashionMNIST* dataset before and after updating. Edit intent: Form clusters of tops, footwear, and other items.

7. KULLBACK-LEIBLER DIVERGENCE (KLD), which we calculate by transforming the pairwise distances in the high-dimensional and low-dimensional space into probability distributions;

8. NORMALISED MUTUAL INFORMATION (NMI), which indicates to which extent points with the same label occur independently;

9. TRUST, which is based the number of neighbours that are in k-neighbourhood in the layout but not in the k-neighbourhood in the original high-dimensional space; and

10. CONTINUITY, which is the inverted variant of the trust score.

We use the same experimental setup as before to form a cluster of points with two different labels. Since the edit intent is to form a single cluster, we update the original labels of the source to the target label before computing the metrics. For the metrics that require a neighbourhood size, we set the k to the average number of items per label. We again move an increasing number of points with the source label towards the centroid of points with the target label as shown in Figure 5.7. In general, we observe that the number of moved points does not significantly affect the scores when updating the layout using EDIMAP. However, as the number of edited points increases, the scores decline when updating the layout with ISP. Generally, EDIMAP performs better than ISP across all datasets and metrics. We do however observe an initial decrease in all scores for both algorithms as soon as the first point was moved. This may be caused by the fact that some points are not moved along with the other points and thus contaminate the k-neighbourhoods. Although some individual points may not perfectly fit in the updated layout and negatively affect these

Table 5.1.: Experimental results of different displacement measures after updating the layout with (ISP/ EDIMAP)-based on simulated merge edits using 1%, 10%, and 30% of points in the source cluster. All values are averaged across multiple runs. Smaller values are better.

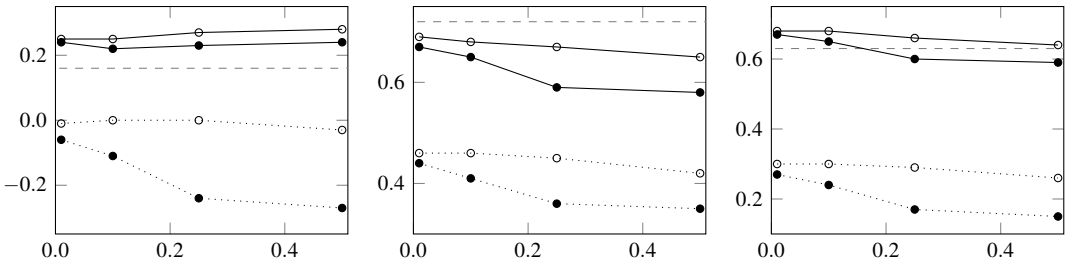
Metric	Blobs		20 news		seeds		MNIST		MNIST1D		F-MNIST	
TOTAL (1%)	0.41	0.08	0.50	0.01	0.44	0.13	0.43	0.06	0.60	0.06	0.35	0.07
TOTAL (10%)	0.41	0.08	0.50	0.01	0.45	0.13	0.43	0.06	0.57	0.06	0.36	0.07
TOTAL (30%)	0.40	0.08	0.50	0.01	0.45	0.15	0.43	0.07	0.61	0.06	0.36	0.07
TARGET (1%)	0.39	0.09	0.55	0.00	0.51	0.24	0.45	0.08	0.62	0.07	0.32	0.10
TARGET (10%)	0.39	0.09	0.55	0.00	0.51	0.25	0.45	0.08	0.59	0.08	0.33	0.10
TARGET (30%)	0.39	0.09	0.55	0.00	0.51	0.26	0.44	0.08	0.62	0.08	0.33	0.10
DIST (1%)	0.44	0.40	0.72	0.39	0.53	0.27	0.48	0.20	0.73	0.12	0.37	0.28
DIST (10%)	0.43	0.39	0.60	0.26	0.58	0.22	0.49	0.17	0.62	0.13	0.39	0.24
DIST (30%)	0.45	0.39	0.59	0.20	0.52	0.15	0.52	0.14	0.68	0.11	0.41	0.17
DTT (1%)	-0.40	-0.03	-0.27	-0.06	-0.23	-0.02	-0.30	-0.00	-0.13	-0.01	-0.33	-0.00
DTT (10%)	-0.39	-0.03	-0.27	-0.06	-0.23	-0.04	-0.29	-0.01	-0.12	-0.02	-0.31	-0.02
DTT (30%)	-0.37	-0.04	-0.26	-0.08	-0.23	-0.09	-0.25	-0.06	-0.13	-0.03	-0.29	-0.09

scores, we have demonstrated in the qualitative results, that general layout was improved with regards to the edit intent.

In conclusion, we were able to demonstrate that our proposed EDIMAP algorithm provides useful assistance for editing layouts for map-like visualisations of a dataset. In judging the performance of EDIMAP or any other editing assistance it is important to note, that identifying the intent of a user's edit is almost impossible. Using only the feedback of dragging a point to a new target location can be interpreted in many different ways. With EDIMAP, we focused on one aspect, the intent of merging clusters of points a user determined to be similar and were able to show its effectiveness to achieve that goal.

5.5. User Interface

We developed a basic prototype to provide an interactive human-machine interface for editing the map layouts. This web-interface allows users to load a pre-computed layout of a dataset, which is rendered in the centre. The screenshot in Figure 5.8 shows the interface with a map of the MNIST dataset, which consists of annotated images of hand-written digits. Users can interact with the visualisation by zooming and panning, as well as editing points by individually dragging them to a different location on the map. All edits are



(a) Silhouette Score *(dotted)* KLD *(solid)*

(b) Accuracy *(dotted)* Continuity *(solid)*

(c) NMI *(dotted)* Trust *(solid)*

Figure 5.7.: Isometrics for updated layouts using EDIMAP (⊸/○) and ISP (•/•) after moving 1%, 10%, 25%, and 50% of points from a source cluster to a target cluster. Averaged across multiple runs and all datasets. Dashed horizontal line (where applicable) indicates results for the original high-dimensional data.

recorded and can be sent to the server where an updated layout is computed based on these edits. If necessary, this process can be repeated until the resulting layout satisfies the user's expectations. In order to best assist the user in that process, only a few examples should be required to achieve that goal. As indicated by the taxonomy of edit intents, deriving the underlying goal only from single edits may almost be impossible. Thus, we expose hyper-parameter settings in the interface, such that a user can set the characteristics of the update algorithm to their needs. However, this requires some level of expertise to utilise the full potential of the algorithm. In a real-world application, these hyper-parameter settings should be replaced by a single setting and visual indicators. For example, when hovering a point to be edited, other points that would be affected by editing that point could be highlighted using a heuristic approximation. Additionally, a single slider controlling the neighbourhood size or force behind the edit could be added, which would be comparable to known interaction patters in image manipulation programs, for example the radius or intensity of a tool.

5.6. Conclusion

In this chapter, we presented an approach for editing map-like visualisations of datasets. Given an existing two-dimensional projection of the high-dimensional representations, our EDIMAP algorithm is able to assist the editing process based on only a few suggested edits by a user. We described, how EDIMAP uses a similarity graph for updating the layout. Additionally, we improved a neural network based model from related work, which we used as a baseline. In a quantitative and qualitative evaluation using several real-world datasets, we were able to demonstrate the effectiveness of our approach.

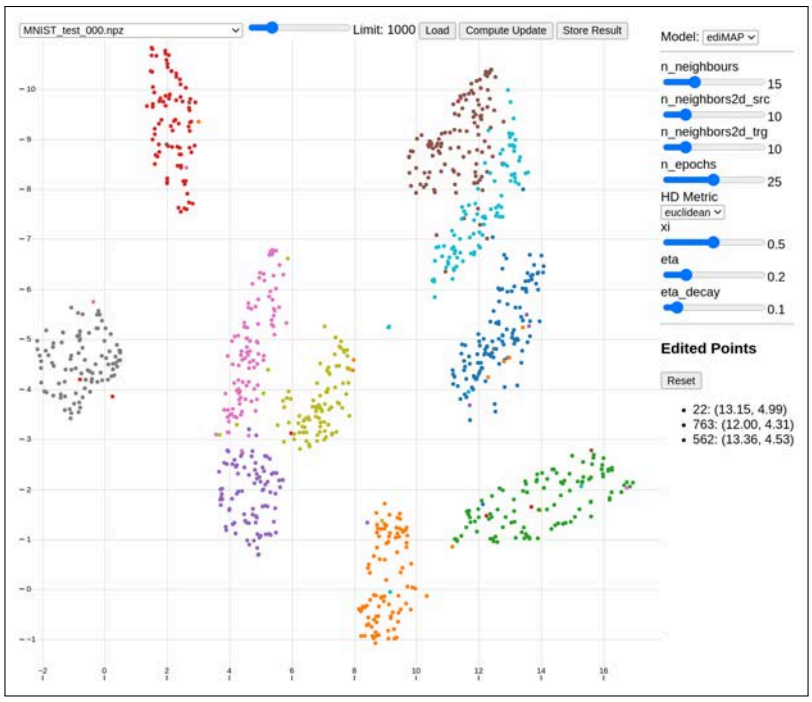


Figure 5.8.: Interface prototype for editing map layouts.

However, as discussed in the evaluation, identifying the intent of a user's edit important, yet very challenging. Conditioning models on specific intents to generate several suggested updated layouts for a given edit could be part of future work, along with an enhanced user interface. Intuitive and semantically meaningful visualisations of datasets heavily rely on expressive high-dimensional semantic representations of items in the dataset. The user feedback could be propagated back to the underlying representation model to improve the model itself, not just the two-dimensional visualisation.

Part II.

Domain-specific Corpus Exploration

Chapter 6

SEGMENTING SEMI-STRUCTURED TEXT IN EMAILS FOR EXPLORATION

Nowadays, email communication plays an integral part of everybody's life. For auditors and investigative journalists, it can be invaluable to extract and analyse communication networks to reveal interesting patterns of decision making processes within a company. Fraud detection is another application area where precise detection of communication networks is essential. In this chapter, we present an approach to untangle email threads originating from forward and reply behaviour with the help of recurrent neural networks.

Problem Statement Free-text emails contain a lot of entangled information. As emails are sent back and forth, prior messages are automatically quoted by the email program, which also inserts meta-data on who sent what when to whom. Identifying these structures ex post is a very hard problem, especially when different programs were involved, the data gets corrupted due to changing encodings, edits by the users, or varying language settings. Investigators often only have access to a limited set of raw information. Thus, extracting as many details as possible from the available data is crucial.

Contributions We developed QUAGGA, a state-of-the-art deep learning model that classifies each line of an email into different zones, namely in-line metadata, greeting and sign-off phrases, signatures, and the actual messages. This model is also at the core of our ready-to-use and open source library.[1] In order to evaluate the model, we created detailed annotations of a subset of the Enron corpus and also created a novel dataset based on user groups in the public Apache Foundation mailing archives. We show that our deep learning approach outperforms state-of-the-art systems based on traditional machine learning and hand-crafted rules. Furthermore, we developed BEACON, a fully integrated platform that can ingest raw email datasets to be explored with an innovative interface.

[1] `https://github.com/HPI-Information-Systems/Quagga`

6.1. Introduction

Emails are an important part of day to day business communication, hence their analysis inspired research from a variety of disciplines. In social network analysis, user profiling, or behaviour analysis often only information contained in the well structured email protocol headers is used. However, a lot more information remains hidden in the free text body of an email, which contains additional meta-data about a discussion in the form of quoted messages that are forwarded or replied to.

In the early days of email communication, users followed clear rules, e.g. prefixing quoted text with angle brackets (>). Nowadays, due to the diversity of email programs, formatting standards, and the freedom to edit quoted text, identifying the different parts of a message body is a surprisingly challenging task. Email programs like Outlook, Thunderbird, or even online services such as Gmail, usually group emails into conversations and attempt to hide quoted parts. To this end, they try to match preceding emails by subject and sender, which fails in case the subject or quoted text was edited.

We propose a neural network based approach for extraction of the inherent structure in email text to overcome problems of error-prone rule-based approaches. This enables downstream tasks to work with much cleaner data and additional information by focusing on specific parts. Further we show improvements in flexibility and performance over earlier work on similar tasks.

Our goal is to extract the inherent structure of free text emails containing a conversation thread composed of consecutive quoted or forwarded messages. Components of an email are referred to as *zones* similar to the definition used by Lampert et al. [106]. Figure 6.1 shows an annotated example for clarification. We assume that a conversation thread is represented as a sequence of *client header* and *body blocks*. A pair of corresponding header and body is called *conversational part* or *message*. Furthermore, we assume that each single line can be assigned to exactly one zone.

In this context, client headers are blocks of meta-data automatically inserted by an email program, usually containing information on the sender, recipient, date, and subject of the quoted email. Generally the header indicates, whether the subsequent message body was forwarded or replied to by the text above. Bodies are the actual written messages, which on reply or forward are quoted below the newer message.

Message bodies can often be further separated into a *greeting* (such as a formal or informal address of the recipient at the beginning of the message), *authored text* (the actual message), *signoff* (closing words of the message), and a *signature* (containing contact information, advertising, or legal disclaimers). As emails with inline relies are usually copied, we consider a block of quoted lines and responses as one body block.

From: Alice *Sent:* Mon, 14 May 2001 07:15 AM
To: Bob, Brian
Subject: RE: Telephone Call with Jerry Murdock

Body	Thank you for your help.
Body	
Body/Signature	ISC Hotline
Header	03/15/2001 10:32 AM
Header	
Header	Sent by: Randi Howard
Header	To: Jeff Skilling/Corp/Enron@ENRON
Header	cc:
Header	Subject: Re: My "P" Number
Body	
Body/Greeting	Mr. Skilling:
Body	
Body	Your P number is P00500599. For your convenience, you can also go to
Body	http://isc.enron.com/ under Site Highlights and reset your password or
Body	find your "P" number.
Body/Signoff	Thanks,
Body/Signoff	
Body/Signoff	Randi Howard
Body/Signature	ISC HOTLINE
Body	
Header	From: Jeff Skilling 03/15/2001 10:01 AM
Header	
Header	To: ISC Hotline/Corp/Enron@Enron
Header	cc:
Header	
Header	Subject: My "P" Number
Body	
Body	Could you please forward my "P" number. I am unable to get into the XMS
Body	system and need this ASAP.
Body	
Body/Signoff	Thanks for your help.

Figure 6.1.: Example email with zones; consecutive blank lines reduced to one

In this chapter, we present QUAGGA, a neural network based approach for email segmentation. Furthermore, we provide a brief overview of our BEACON system for ingesting and exploring large email corpora.

6.2. Related Work

Email corpora provide fascinating insights into human communication behaviour and therefore inspire research in many different areas. Datasets such as the Enron [98] or Avocado corpus [144] provide real world information about business communication and contain a mix of professional emails, personal emails, and spam. Ben Shneiderman published parts of his personal email archive for research [150]. Also popular is the 20 Newsgroups dataset [108] sampled from newsgroup postings in the early 90s, which we discard as it

contains only few conversation threads. For the work at hand, we use the Enron corpus and emails we gathered from public email archives of the Apache Software Foundation[2].

A recent survey shows the diversity of email classification tasks alone [135]. Similarly interesting is the analysis of communication networks based on meta-data like sender, recipients, and time extracted from emails [23].

Models based on the written content of emails may get confused by automatically inserted text blocks or quoted messages. Thus, working with real world data requires normalisation of data prior to the problem at hand. Rauscher et al. [159] developed an approach to detect zones inside work-related emails where relevant business knowledge may be found.

In their work towards detecting emails containing requests for action, Lampert et al. [107] observed a relative error reduction by 40% when removing quoted sections of emails. Similar observations were made more recently predicting reply behaviour within the Avocado dataset [211].

Thread Reconstruction. Another popular area of research is the reconstruction of graphs reflecting which message responds to another. Wang et al. propose baseline approaches based on temporal relationships [205]. There are also more advanced models that use sentence-level topic features to resolve a message graph using random walks [91]. Most recently, Tien et al. [201] proposed a novel convolutional neural network over a grid built by assigning roles to extracted entities. The latent graph is derived from the configuration with the highest coherence score. In our work however, we only focus on separating conversational parts within free text messages, not the actual reconstruction of the thread.

Email Zoning with Rules and Text Alignment. We identified three approaches to email zoning: rule based, text alignment, and machine learning.

The most naïve approach is to write specific rules that match commonly used patterns in email text. Talon[3] provides a sophisticated set of patterns to match most popular client header formats. The obvious downside is the lack of flexibility and that it's error-prone to changes.

Assuming a complete email corpus, a message in one user's outbox may be found in the inbox of other user(s). Likewise, quoted messages exist within the corpus as an original message from preceding communication. By finding overlapping text passages across the

[2] `http://mail-archives.apache.org/mod_mbox/`
[3] `https://github.com/mailgun/talon`

corpus, Jamison et al. managed to resolve email threads of the Enron corpus almost perfectly [86]. It has to be noted, that the claimed accuracy of almost 100% was only tested on 20 email threads.

In order to reassemble email threads, Yeh et al. considered a similar approach with a more elaborate evaluation reaching an accuracy of 98% separating email conversations into parts [212]. To do so, they rely on additional meta information in emails sent through Microsoft Outlook (thread index) and rules that match specific client headers. Thus, such an approach will not work on arbitrary emails, nor can it handle different localisation or edits by the user.

Contrary to approaches using text alignments, we don't assume a complete corpus. Our goal is to extract all information from only a single email archive or even a single email.

Machine Learning for Email Zoning. Another approach to email zoning uses machine learning with carefully designed features.

Carvalho and Cohn proposed JANGADA [29], a system to remove quoted text and signature blocks from emails in the twenty newsgroup dataset [108]. They first classify emails to find those that contain quoted text or signatures and then classify each line individually using Conditional Random Fields (CRF) and sequence-aware perceptrons. Reported accuracies range from 97% to above 99%.

Other researchers applied JANGADA to Hotmail emails and measured accuracies around 64% [56]. With some adaptation, they managed to extract five different zones (author text, signature, advertisement, quoted text and reply lines) with an average accuracy of up to 88%.

Lampert et al. developed the ZEBRA system [106] as a pre-processor to their previously mentioned work on requests for action [107]. Adversely to previous approaches, they use Support Vector Machines and therefore classify lines of an email into zones individually rather than considering a sequence of lines. For that, they describe graphic, orthographic, and lexical features to represent lines within their context reaching an average accuracy of 93% on the two-zone task and 87% on a nine-zone task. Comparing the performance by zone type, most problems are caused by signature lines (F-score around 60%), signoffs (70%) and attachments (69%). It was found, that adding contextual features didn't improve the performance [106]. Contrary to our objectives, ZEBRA only tries to identify the zones within the very last message within an email thread and rejects the rest as quoted text, whereas we aim to detect the zones across the entire email.

We compare results of our system described in Section 6.3 with JANGADA and ZEBRA. We not only aim to improve upon those results, but also provide a system that is able to detect zones along the entire conversation thread contained in an email and not only the latest part.

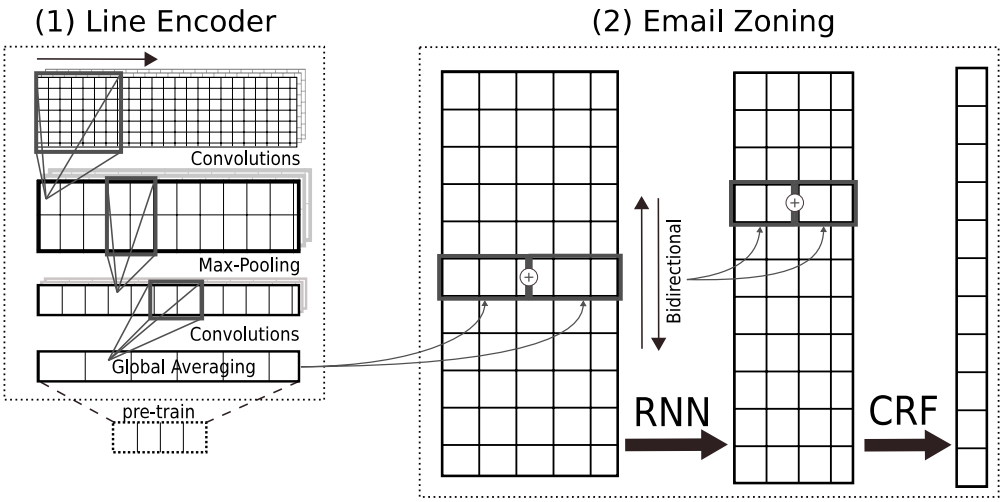


Figure 6.2.: Schematic model overview; Left side shows line embedding stage using the CNN approach, right side outlines email zoning model.

Furthermore, our system uses neural networks rendering expensive and potentially error-prone feature engineering obsolete. In this way, even very small or incomplete datasets can be utilised for downstream tasks like social network analysis, speech act recognition and other research areas using email data.

6.3. Neural Network for Detecting Document Segments

Systems for email segmentation that are discussed earlier are based on hand-crafted rules to match common structures directly or use them as features for machine learning models. Such approaches will fail when client headers are localised, formats are changed, or quoted messages are edited by users or get corrupted. Humans will most likely be able to identify the structure of an email even for languages or formats they don't know. Pre-defined rules on the other hand would either be too general and match many false-positives or grow to be very complex.

In most cases it may seem obvious to the human eye how to segment an email into client headers and quoted text even though different or corrupted formats are used. However, even a sophisticated text parsing program will fail since client headers follow no standardised format. Usually lines start with attribute keywords such as "From:" or "Subject:", however their value may span multiple lines and use varying delimiters. This even makes it hard to detect the boundaries between header and body blocks, since one can not rely on the presence of keywords or well formed, deterministic schemas.

In this chapter, we propose QUAGGA based on neural network architectures. As shown in Figure 6.2, emails are processed in two stages: the line encoding and the email zoning stage. In this section we describe he how email text is represented and how classifiers can be used as a reliable and robust preprocessor for a simple program to extract its inherent structure.

6.3.1. Representation of Email Data

In the initial stage of our system, the email text data is encoded into a low dimensional space to be used as input to the second stage as outlined on the left side in Figure 6.2. The smallest fragments to be considered for email zoning are the lines in the email text. Lines are delimited by the newline character (`\n`), which may not necessarily be the same as wrapped lines displayed by an email program. Analysis of the annotated data shows that this granularity is sufficient for all header, body, and signature zones as was assumed by other research on similar tasks.

Each line is encoded as a sequence of one-hot vectors representing respective characters. We distinguish one hundred different case-sensitive alpha-numeric characters and basic ASCII symbols plus an out-of-scope placeholder. This is sufficient for all email corpora we looked at, since only a negligible portion of characters exceeds this set. We presume, that this could be adapted for applications with Cyrillic, Arabic or other alphabets.

Inspired by research on character-aware language models [97], we devised a recurrent and a convolutional neural network model. The recurrent model consists of a layer with varying number of gated recurrent units (GRU), where the last unit's output serves as a fixed size embedding of the line. The convolutional model uses two convolutional layers (CNN), which scan the sequence of characters in a line and are intertwined by max-pooling and global-averaging layers finally leading into a densely connected layer, where the number of neurons corresponds to the embedding size as shown on the left in Figure 6.2.

In both models, the line representations are learnt in a supervised fashion. During training, a densely connected layer with softmax activation is appended so that a classifier can be trained to distinguish between lines of corresponding zone types. The optimal hyper-parameter settings of the topology such as the number of layers and embedding size was determined experimentally. We provide a detailed analysis of the influence of limiting the length per line on embedding accuracy later on.

6.3.2. Classification of Email Lines

We use the dense vector representations described above to classify each line into zones. However, without the context in which a line appears in, the classification performance

on ambiguous or deceptive cases won't be optimal. Thus, we do sequence-to-sequence classification using a GRU-CRF model as outlined in the right part of Figure 6.2, which takes a sequence of line encodings per email as input. Three of the five zone types only appear within message bodies, so we use two concatenated embeddings as input, where one is pre-trained using two- and the other with five-zone classification.

Best performance was achieved with a bi-directional GRU layer, which scans the lines from top to bottom and in reverse order and concatenates both hidden states of each line. In sequence-to-sequence classification, recurrent neural networks only consider the previous hidden state but neglect the actually predicted label sequence. We already observed small improvements by using a bi-directional layer over a uni-directional one, since each line's context reflects the previous and following lines. However, like in language models [83, 120], the addition of a conditional random field (CRF) to the output shows further performance gains.

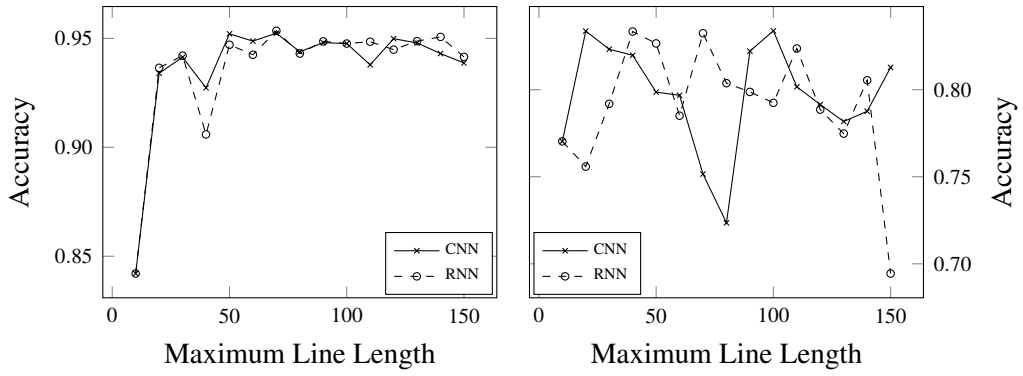
Training both parts of the system as an entire model in one pass by directly connecting the encoder output layer to the second stage model's input lead to unstable results, even after pre-training the line encoder. Therefore we train the model in the second part of our system separately from the line encoding model.

The predicted sequence of zone types can be used to extract the conversational parts of an email and also separate the message from additional content such as signatures, greetings and sign-offs. Consecutive lines with the same predicted zone type are aggregated into a block. Further processing inevitably requires making some assumptions about the general structure of emails. Based on our analysis of emails in the training dataset, we assume that a body always proceeds a header block. We do not segment the individual message bodies any further, even when they contain in-line replies and quoted parts.

Small errors in the prediction can be fixed heuristically. For example a block with a single line classified as header containing only the "Subject:" keyword likely is either a false positive or belongs to another block nearby. However, since such rules could reduce the initial robustness, we omit them in the evaluation in this chapter. We found, that using QUAGGA as a pre-processor for finding related blocks significantly improves the accuracy of parsing rules in downstream tasks, such as constructing communication graphs from header blocks compared to a purely rule-based parser without pre-processing.

6.3.3. Selection of Model Parameters

In the description of the proposed model we highlighted adjustable parameters. This includes the model for line representations in the first stage, limiting the length of each line, and finding the ideal dimension of line embeddings. We base the model's topology configuration on the analysis of related models [83, 97, 120].



(a) Two-zone Model

(b) Five-zone Model

Figure 6.3.: Accuracy for increasing max line length using 32-dimensional embeddings

The ideal configuration for the line encoder model is determined through grid search across hyper-parameters. We record the accuracy of the convolutional and recurrent approach for line encoding in Figure 6.3. Note, that the convolutional model assumes a fixed size input, so shorter lines are zero-padded. When evaluating the reported accuracy, one has to consider two things. First, this metric may not project down to the later stage of our system, and second, the label distribution biases range of values.

We did not observe significant differences between embedding dimensions above 32, so we choose this dimensionality in favour of a less complex model. Most errors are caused by blank lines, which are usually classified as “Body”. Results when training using two-zone classification are mostly stable for both models. The majority of lines in our training data are between forty and fifty characters long.

Both approaches for line representations seem to have their strengths and weaknesses. Since there is no clear winner, we continue only using the convolutional model in this work and fix the input length to 100 characters per line. We do so based on the argument, that one may want to process large corpora and prefer a faster system.

6.4. Evaluation

In this section we present an overview of the email datasets we used and discuss the sampling of emails to create an unbiased evaluation set. Further, we describe competing approaches that are used as baselines for comparison of our results. We also analyse model parameters and its robustness to changes in email text.

6.4.1. Datasets

We evaluate our proposed approach on the Enron corpus [98] and emails gathered from public mail archives of the Apache Software Foundation[4] (ASF). Estival et al. [56] and Lampert et al. [106] discussed shortcomings in working with Usenet-style emails, leading us to refrain from using the 20-newsgroup dataset [108] as was done for the JANGADA system. We found that more recent email threads from the ASF archives, especially those on mailing lists for users of different software projects, offer diverse formatting patterns.

Each dataset is divided into three subsets for training, testing, and a final evaluation. Emails are sampled at random from their respective original dataset and put into one of those subsets. To ensure representative results that are not biased by author or domain, sampling per subset is restricted to distinct mailboxes (Enron) or mailing lists (ASF). The ASF dataset was compiled by randomly selecting emails from the *flink-user*, *spark-user*, and *lucene-solr-user* mailing list archives.[5]

We manually annotated all lines in both datasets as being part of the in-line meta-data (*Header*) or the written message (*Body*), which corresponds to the two-zone classification. For a more fine-grained segmentation of the message, the *Body* is furthermore separated into *Greeting* (block of introductory words), *Signoff* (block of closing words), and *Signature* (automatically inserted email signature block) for the five-zone classification. In order to enable future work on parsing emails in even more detail, we also annotated around 400 emails from the Enron corpus on a character level. Hereby, all meta-data from the header, such as names, dates, email addresses, and subject line were marked. Furthermore, these names were also annotated in the body of the email, including references to attached files and all information from the signature, such as phone numbers, position, postal address, and organisation. Where possible, all information is linked, for example to specify which phone number or address belongs to which name mentioned.

Table 6.1 provides statistics for both datasets including the expected number of messages to be extracted. Prior heuristic analysis of the Enron corpus estimated 60% of emails to contain conversation threads [98], which is close to our annotated data. On average an email has two parts with 20 lines per message. Only a few messages contain a signature, which on average are six lines long.

6.4.2. Competing Approaches

We compare our proposed model for extracting zones from emails against several other approaches. Most notably, JANGADA [29] and ZEBRA [106] are reimplemented with slight

[4] `http://mail-archives.apache.org/mod_mbox/`

[5] Annotated datasets and code can be found at `https://github.com/TimRepke/Quagga`

Table 6.1.: Annotated Datasets in Numbers

	Enron			ASF		
	Train	Test	Eval	Train	Test	Eval
Emails	500	200	100	350	100	50
Individual messages	1,048	474	233	934	226	108
Average length of threads	3.5	3.6	3.5	3.5	3.7	3.1
Number of signatures	103	58	26	76	13	5
Number of lines	23,467	8,531	4,248	24,630	6,082	2,919
Number of *Header* lines	3,373	1,451	740	1,146	264	129
Number of *Body* lines (2 zones)	20,088	7,080	3,508	23,484	5,818	2,790
Number of *Body* lines (5 zones)	18,887	6,484	3,256	20,989	5,333	2,461
Number of *Greeting* lines	78	55	26	537	146	89
Number of *Signoff* lines	531	235	119	1,561	272	216
Number of *Signature* lines	598	306	107	397	67	24

modifications to fit the more refined problem statement. Both systems originally are intended to distinguish lines within an email, which are not part of the latest message of that thread. Clearly that deviates from our goal to extract *all* individual parts and detect zones with additional detail within those. Since the systems are supposed to detect zones within the first part of the email, their features and underlying models should in principle also work on our task.

The source code for JANGADA is freely available on the author's web page,[6] which we used as a basis for our implementation in Python, incorporating the originally used model for sequence labelling, which is part of the MinorThird Library.[7] For extracting signatures, JANGADA originally only considers the last ten lines of an email. In our implementation, the perceptron performs a multi-class classification along all lines of the email. The model is trained with window-size 5 for 40 epochs.

The ZEBRA project web page[8] does only provide annotated data, but not the system's source code. Gossen et al. implemented[9] it for their work on classification of action items in emails [184]. We used that as a guideline for our adapted Python implementation. The SVM is trained for a maximum of 200 iterations in a one-versus-rest fashion for multi-class classification using RBF kernels.

[6] `http://www.cs.cmu.edu/~vitor/software/jangada/`
[7] `http://minorthird.sourceforge.net/`
[8] `http://zebra.thoughtlets.org/zoning.php`
[9] `https://github.com/gerhardgossen/soZebra`

Table 6.2.: Precision, recall, and accuracy of classifying email lines into zones

		Enron			ASF		
Approach	Zones	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
JANGADA [29]	2	0.89	0.88	0.88	0.97	0.97	0.97
ZEBRA [106]	2	0.66	0.25	0.25	0.88	0.18	0.18
FeatureRNN	2	0.98	0.98	0.97	0.97	0.95	0.94
QUAGGA	2	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**
JANGADA [29]	5	0.82	0.85	0.85	0.90	0.92	0.91
ZEBRA [106]	5	0.60	0.25	0.24	0.81	0.20	0.20
FeatureRNN	5	0.92	0.75	0.75	0.90	0.60	0.60
QUAGGA	5	**0.93**	**0.93**	**0.93**	**0.95**	**0.95**	**0.95**

In order to compare how more modern machine learning models perform using the features proposed in JANGADA and ZEBRA, we use those as input for a recurrent neural network with two GRU layers [37], which we will refer to as *FeatureRNN*. The above models are baselines for the comparison to our proposed QUAGGA system using a convolutional model as line encoder with fixed input sizes of 100 characters per line.

6.4.3. Results

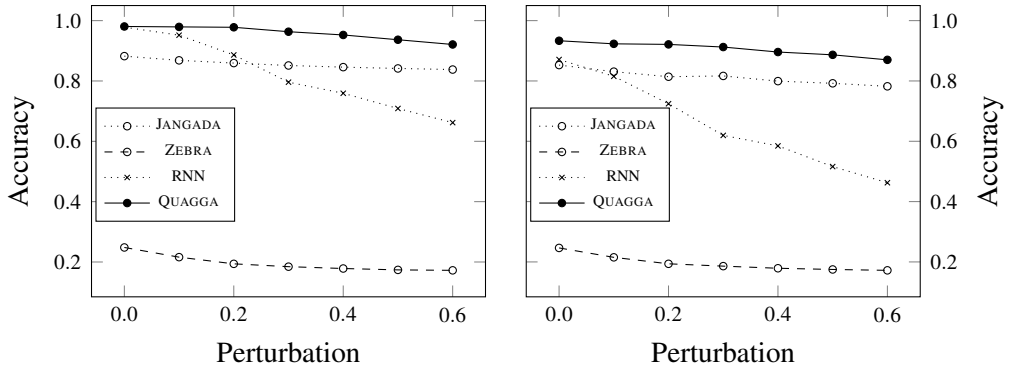
In this section we compare QUAGGA to similar systems found in related work. To get a good understanding of the versatility, we not only look at the results shown in Table 6.2, but also consider the robustness against noise or otherwise changing data as well as how many training samples are required to get good results.

We were not able to reproduce reported accuracies of ZEBRA [106] in our widened scope to full emails. This was expected, given the nature of the features and context-free classification of lines. JANGADA uses more general features and looks at a sliding window of lines and we got close to reported accuracies, especially for the ASF dataset, which is closer to the twenty newsgroup data the authors used [106]. Overall, our system shows very good performances and seamlessly adapts to other datasets without problems.

Number of Training Samples Complex neural network based machine learning models require lots of training samples to reliably proficiently solve a given task. We limited the number of Enron emails shown to the network during training down to 10% of the Enron training set, corresponding to 50 emails and then measured the performance whilst continuing to add training samples up to all 500 emails. The model trained with the least

Table 6.3.: Precision, recall, and accuracy of classifying email lines from the respectively opposing dataset the model was trained on.

		Enron → ASF			ASF → Enron		
Split	Zones	Prec.	Rec.	Acc.	Prec.	Rec.	Acc.
Test	2	0.94	0.94	0.94	0.88	0.88	0.88
Eval	2	0.97	0.97	0.97	0.88	0.88	0.88
Test	5	0.89	0.89	0.89	0.83	0.78	0.78
Eval	5	0.89	0.89	0.89	0.88	0.81	0.81



(a) Two-zone Model

(b) Five-zone Model

Figure 6.4.: Robustness against perturbation for the Enron test set

data in this scenario only lags behind around 1% in accuracy compared to a model trained on all data in both the two- and five-zone task.

Cross-corpus Compatibility Ideally, a system like QUAGGA would be trained once and work well on arbitrary emails. Since the style of emails in our two datasets varies, we are able to simulate how well our model can abstract to unseen data beyond the training and testing data of only one corpus. To this end, we trained QUAGGA on Enron and tested it on ASF emails and vice versa. The results are and shown in Table 6.3. We observe, that by training on ASF emails, QUAGGA does not generalise as well to emails from he other corpus (F1: 0.86 or 0.78, for two- or five-zones) as the other way around (F1: 0.94 or 0.87). Compared to the results when training and testing using emails from the same dataset, this leads to a decrease of performance of around 4-10%.

Robustness to Noise Our hypothesis is, that a model which learns meaningful features itself is more robust towards changes to the email text as hard coded rules responding to specific keywords or patterns. To show the robustness of our model, we introduce the notion of a perturbation threshold $\rho \in [0,1)$. Before passing an email to a model, a function iterates over each character and with probability ρ edits, removes, or duplicates it. Training was performed on uncorrupted data only.

The robustness of each model against increasing perturbation is shown in Figure 6.4. The drop in performance is the same the for both, the two- and five-zone task, although at different absolute accuracies. QUAGGA doesn't seem to be affected up to $\rho = 0.2$ and keeps producing reliable results even at higher perturbation thresholds. Surprisingly, also JANGADA is not influenced significantly by the introduction of perturbation. As opposed to ZEBRA and FeatureRNN, it is using more features related to small patterns or proportions of types of symbols, whereas the others depend on more complex patterns which are more error prone to change.

6.5. User Interface

Corpora of day-to-day emails contain the history of many relevant discussion points that drive a business. Often this data is analysed retrospectively by internal auditors in companies, law enforcement agencies, or journalists who received leaked documents. Sifting through massive amounts of emails to identify fraudulent or immoral behaviour is a tedious task, which sometimes hundreds of experts up to a year to fully analyse. Traditional tools for this task, such as the NUIX Engine[10], usually only provide keyword or pattern search and access to a database with raw extracted data. Even tools like NetLens, which is specifically tailored to explore the Enron corpus, are comparably rudimentary in their overall capabilities [92]. However, as internal auditors of a large German bank reported, investigations without prior knowledge of what to look for exactly may result in expert users having to read lots of potentially irrelevant emails.

To this end, we propose BEACON, a system specialised for the exploration of large scale email corpora during an investigation. By combining communication meta-data and integrating additional information using advanced text mining methods and social network analysis, our system goes beyond traditional approaches. The objectives are to provide a data-driven *overview* of the dataset to determine initial leads without knowing anything about the data. The system also supports extensive filters and rankings of available information to *focus* on relevant aspects and finding supporting emails. At each point, the interface components are updated to provide the relevant *context* in which a certain information snippet is embedded.

[10]NUIX Analytics extracts and indexes knowledge from unstructured data (nuix.com)

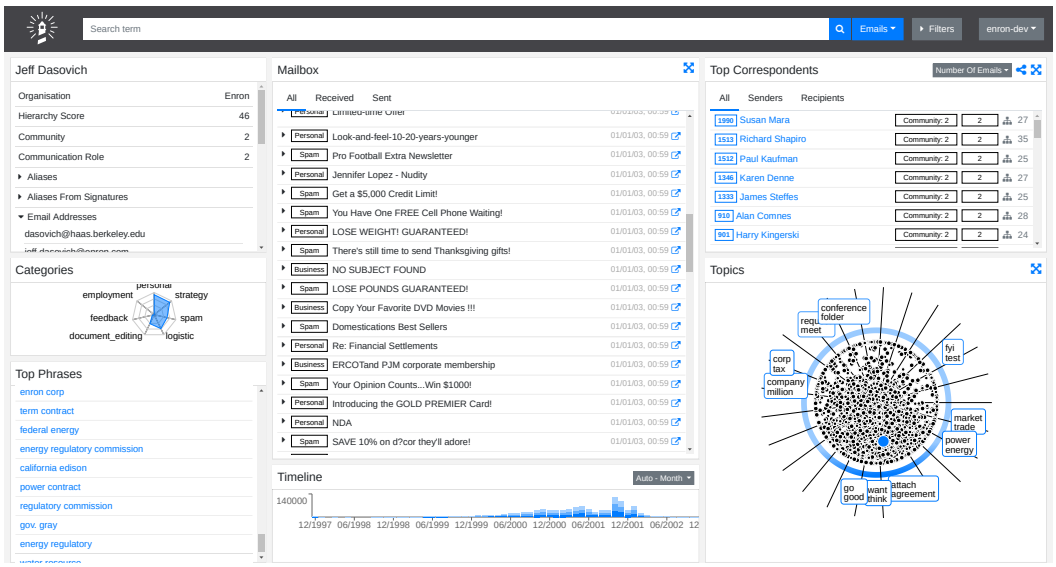


Figure 6.5.: Correspondent View showing one mailbox, top phrases, extracted details, correspondents, topic distribution, etc.

The raw dataset is processed by a distributed ingestion pipeline written in Apache Spark.[11] This pipeline first cleans the emails as described earlier in this chapter, removes duplicate messages and merges aliases of correspondents (e.g. when different email addresses or name abbreviations are used). We use topic models trained on the cleaned messages to find salient semantic structures, classify the emails to separate personal emails, newsletters, or spam from business-related emails [214], and cluster emails [189]. Furthermore, we analyse the correspondence graph to reveal communities and roles [35], detect hierarchies [40], and rank correspondents [47].

The BEACON system's user interface (see Figure 6.5) provides powerful search capabilities and visualisations for all unstructured text and structured information extracted by the processing pipeline. It is organised into two views, the search view and correspondent view. Components of these views handle specific aspects, e.g. the network graph, and are tightly integrated with one another. The interface concept is tailored to the needs of expert users during an investigation. It is designed to approach a corpus with a text-driven or network-driven strategy, while keeping the option to switch context at each point. We preserve data provenance, so that the original data can be accessed from any point. The interface also provides innovative visualisations of the data, such as Topic Spheres [53] and interactive communication matrices.

[11] `https://spark.apache.org`

We evaluated the performance of the system using real world datasets of different size, notably emails of the U.S. Democratic National Committee (DNC) leaked in 2016 and the well known Enron Corpus [98]. A prototype was also employed in an active internal investigation of a large German bank.

6.6. Conclusion

In this chapter, we presented a reliable and flexible system for finding the inherent structure of an email message chains containing multiple conversational parts. In the first stage, the system uses a convolutional neural network to encode lines of an email which are used by a GRU-CRF to predict a sequence of zone types per line reaching accuracies of 98%. Compared to similar models, we show significant improvement, especially a seamless adaptation to other datasets as well as robustness against corrupted data. Research based on email data can largely benefit from this system by pre-processing the text and focus downstream algorithms on relevant parts of an email like client headers or the actual text cleaned of irrelevant parts.

In addition to the QUAGGA system, we provide a detailed annotation of a subset of the Enron corpus that can directly be used for building communication networks without further parsing including linked person aliases and, if present, contact details from email signatures as well as the new ASF corpus.

The Future of the Zebroid Zoo.

Zebra *(M. Katz, 2015, CC0 1.0, flickr)*; Quagga *(F. York, 1870, CC BY-NC-SA 3.0 [84])*; Chipmunk *(G. Gonthier, 2007, CC BY 2.0, flickr)*; Okapi *(E. Kilby, 2013, CC BY-SA 2.0, flickr)*; Zorse *(Fährtenleser, 2019, CC BY-SA 4.0, Wikimedia)*

This chapter is based on our QUAGGA paper [163]. Since its publication, others have used our approach in their research. For example, to improve the detection of phishing and spam emails [58, 182]. Aside from making the code for QUAGGA available as an easy-to-use library, we also parsed the entire Enron corpus and provide GraphML and JSON exports. Bevendorff et al. [18] extended our approach in CHIPMUNK by adding language models and segmenting emails into 15 domain-specific zones and parsed 153 million emails. We were able to show that QUAGGA has good cross-language capabilities on the ASF dataset. Jardim et al. [87] proposed OKAPI, which adds a pre-trained language

model to the QUAGGA line encoder to further improve the language independence. This summary shows, that basic automated email segmentation line-by-line can be achieved with very high accuracies. However, parsing full-text email data on a character level has not been attempted yet, which would allow the extraction of additional metadata. Linking names and email addresses into groups of aliases, for example, could enhance tools for data-driven investigations.

Chapter 7

EXPLORATION OF ONLINE NEWS COMMENTS

The comment sections of online news platforms have shaped the way in which people express their opinion online. In some cases, there are tens of thousands of comments on a single news article. Massive amounts of comments like that will likely never be read in full by other visitors. Even when articles only receive a few hundred comments, it becomes challenging to get an overview of all the different thoughts users shared.

Problem Statement These overwhelming number of comments, which regularly also include irrelevant and toxic comments, often prevent in-depth discussions to emerge. Summarising the prior discussion in such a way, that readers joining the conversation can get an overview of the topics discussed and the different arguments expressed could significantly foster the exchange. It would also offer journalists and other stakeholders to engage in the discussion, draw feedback, or to gauge the public opinion on certain topics. This summary not only has to reflect the contents, but also other aspects, like the sentiments and distribution of different opinions. Furthermore, tracing back aggregates to individual comments needs to be possible in order to respond accordingly if required or to verify the algorithms results. Since arguments and topics can sometimes have many tightly linked aspects, the summary needs to be explorable.

Contributions To foster more interactive and engaging discussions, we propose our COMEX interface for the exploration of reader comments on online news platforms. Potential discussion participants can get a quick overview and are not discouraged by an abundance of comments. It is our goal to represent the discussion in a graph of comments that can be used in an interactive user interface for exploration. To this end, a processing pipeline fetches comments from several different platforms and adds edges in the graph based on topical similarity or meta-data and ranks nodes on metrics such as controversy or toxicity. By interacting with the graph, users can explore and react to single comments or entire threads they are interested in.

7.1. Introduction

In the past, newspaper readers could only interact with and express their opinion on an article by writing a letter to the editor. The editor could then decide to publish and/or to reply to the letter in the next issue of the newspaper. The considerable effort of writing and mailing such letters, was a natural limiting factor for the number of interactions. Nowadays, users of online news platforms can easily post comments and discuss article topics with others. Expressing ones opinion takes less effort than ever, it comes free of costs, and thanks to mobile devices it is available from anywhere at anytime. The simplicity and ubiquity of expressing one's opinion online was therefore termed as *democratisation of opinion*. On the downside, readers can be overwhelmed by the volume of comments. Repeated arguments, troll comments, or attention-seeking unrelated opinions hinder the emergence of meaningful discussions. Long discussions across multiple pages may discourage readers from scrolling through more than the top ten comments.

We envision a platform that focuses on providing a space for discussions where people listen to and refer to each other's comments. To this end, we part from a traditional "linear" list to a two-dimensional canvas that groups comments using different features for a better overview. This allows for new interaction paradigms that could inspire readers of news comments to engage in an already ongoing discussion.

In this chapter, we present ComEx, a platform for visualizing of and interacting with online discussions. It features a novel concept of stipulating engagement through improved information visualisation. In this regard, we identified three components that are crucial to reach this goal:

1. *More engagement:* more users who were passive in the past should become active contributors in discussions.
2. *More in-depth:* more comments should refer to one another and more dialogues should emerge.
3. *More insights:* users should read more relevant and less redundant or off-topic comments.

Besides these user-centred aspects, technical aspects are also currently preventing a better user experience. Online discussion spaces are fragmented across various individual platforms even though the topics discussed are typically similar, e.g. daily news events. To our knowledge, we are the first to introduce the idea of a common, shared discussion platform with the goal of increasing engagement of discussion participants and facilitating interaction. To this end, we present a novel interface for exploring large amounts of reader comments across different news platforms. The core of the visualisation is based on a graph representation of comments, where nodes are sentences and edges describe how

they relate to one another. This graph allows us to incorporate several views on the data and enrich the comments with syntactic and semantic features, such as topical similarity or temporal proximity. It is our goal, to find a graph representation that captures arguments and the evolution of the discourse. By clustering, filtering, and merging, the interface enables users to reduce the complexity by exploring the comments at different levels or granularity.

The following sections provide an overview of the system architecture of ComEx and describe the different visualisation and interaction paradigms behind it. Furthermore, we discuss our early work towards a meaningful graph representation and showcase initial results applied in a case study on reader comments about bushfires in Australia.

7.2. Related Work

The goal for our interactive visualisation is to form visual clusters of comments that make it easy to comprehend the intrinsic semantic structure of a large set of comments. To achieve this goal, the underlying layout model not only reflects the structural information, e.g., sentences belonging to the same comment, but also the key topics and arguments made. Appropriately mapping the nuances of a discussion, the size of the dataset, and the text lengths pose as hard problems for language models. Attempts using topic models have been made to visualise political speeches [53] as moving particles or text collections as glyphs symbolising topic distributions [171]. Others use document embeddings and dimensionality reduction to create a partial map of Wikipedia articles [188] or scatterplots of forum posts [149]. Both examples are not applicable here, as they rely on a large, manually labelled dataset, so we propose to use pre-trained sentence embeddings. Based on those, we can cluster the comments of one story into key discussion points in a data-driven fashion. The layout within each cluster is done using attracting and repelling forces between particles based on sentiment or keywords. Thereby, we benefit from sentence embeddings to get a global layout and achieve a nuanced local layout by using mined meta-data (clusters, keywords, sentiment, etc.).

Related work in the area of text mining forms clusters of comments mentioning the same entities [155] or and visualises discussions with pie charts [65] or topic-model-based graphs [3]. Zhang et al. [216] focus on summarising social media posts to provide aggregates of all reposts and replies in a conversation. They form pseudo-documents as context used in an encoder of a recurrent neural network from which the summary is generated. Leveraging sentiment analysis and stance detection, there is also related work on allowing users to search for diverse perspectives on the same topic [73, 105]. In their analysis of millions of comments, Ambroselli et al. [6] identified three main causes for increased user engagement: reactions to personal stories, hate speech, or comments by the article's author. Our

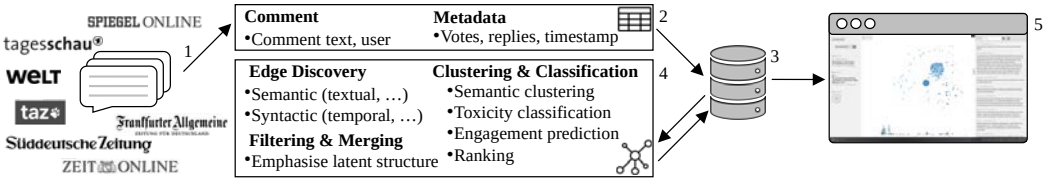


Figure 7.1.: The COMEX system translates a set of news comments into a graph structure and allows their exploration through a web interface.

system allows integrating such additional functionality in the form of data processing modules. As an example, we incorporate a comment classification approach that identifies main causes for increased user engagement based on comment texts [173]. Thereby, COMEX can highlight engaging points of a discussion that are likely to trigger many user reactions. More distantly related is work that supports exploratory search through scientific articles [138]. A comprehensive overview of the characteristics of exploratory search has been published by Palagi et al. [145].

7.3. System Overview and Paradigms

The COMEX system implements a novel concept of interacting with and getting an overview of the growing number of reader comments in online news discussions. Figure 7.1 shows the system architecture. The data ingestion pipeline scrapes comments from different news platforms (1). The comment texts and their metadata, such as upvotes, references to other comments, or timestamps are then stored in a relational format (2), which is cached (3) to reduce the load on the news platforms. This data is transformed into a graph structure, where the nodes (sentences of comments) and edges (relations between sentences) are processed with text mining and graph analysis algorithms, such as semantic clustering, PageRank, and toxic comment classification (4). The results are sent back to the cache. A web-based user interface allows exploring the enriched graph at different levels of detail (5). The architecture of the system is designed in such a way, that text and graph processing modules are interchangeable and can easily be configured. More details on that are highlighted in the following section. The representation model and pipeline can be used programmatically for experiments or other applications. In the scope of our system, the data is accessed through a highly customizable API by our interactive frontend.

We enrich our graph representation through text mining and graph analysis beyond the comment meta-data. By aggregating the data, users are enabled to actually comprehend the entire dataset in one compact view and gain information. This aggregation comes with loss of details, which has to be balanced with the benefits of a better overview. Although

we refer to the comments on online news platforms as a discussion or discourse, many statements and arguments are frequently repeated by different users without referring to an already existing comment. Our graph representation helps to identify and visualise these inherent semantic clusters of comments. In the most simplified view, the graph representation is used to draw particles on a two-dimensional canvas. Hereby, the comment positions reflect semantic similarity and cluster affiliation. To convey more additional information, particles can be drawn as glyphs or vary in size or colour. The canvas can be enriched by overlays of cluster contours, heat maps, and explanatory keyphrases.

We embed comment threads originating from multiple news platforms in the same space, thus merging topically related discussions from various articles. In this way, we provide a global view and increase the diversity of represented opinions following our three key goals. The summarising visualisation aims for *more engagement*, reducing potential bias in discussions by having a broader group of contributors and novel playful ways of interaction, such as reacting to clusters of comments.

We anticipate a larger number of replies in general and deeper threads, which means *more in-depth* replies. Receiving a reply to one's comment acts as an acknowledgment for the user and demonstrates that the comment is relevant to others. Readers should be able to easily retrieve those comments that are most relevant to them, as reading every single comment becomes infeasible for popular articles. Our interactive visualisation condenses long discussions into groups of similar comments for *more insights*. In this way, we are still able to show all contributed comments, while users can make an informed decision on which subset of comments to actually read if a particular aspect caught their attention. This overview could also give information on different viewpoints, such as how many commentators share a particular point of view. By retaining data provenance information, users are able to switch back and forth between the generalised overview and the underlying data for more details. Further, the system includes a full-text search and time or lasso selection in the interface to filter the list of comments. Afterwards, users can jump back to the original platform to comment, or react to a selection of comments directly in the visualisation.

7.4. Graph Representation of Reader Comments

The processing pipeline transforms the comment data retrieved by the scrapers into a graph and further enriches it. Each comment may contain more than one main semantic aspect, such as different arguments or responses to other comments. Thus, we heuristically assume sentences to be the smallest "atomic" semantic unit of a comment. Each sentence is added as a node in the graph representation of a discussion. By adding edges between sentences of the same comment, we are able to maintain data provenance along with meta-data about

the original content. In this section, we discuss possible data mining methods to enrich the graph representation.

The first step is the discovery of relations between sentences and adding edges to represent these relations. Second, we assign class labels and scores to the nodes and edges. This allows us to rank and cluster them based on these assignments. Finally, the number of nodes and edges is reduced by filtering or merging them to provide a comprehensible entry point. Note, that edges of the graph are only the basis to internally represent reader comments and the layout. By default, edges are not shown in the interface to reduce visual clutter.

Edge Discovery. Our approach builds on ideas by Barker and Gaizauskas [15], who represent arguments in comments (assertions or viewpoints) in a graph. Given this graph, they generate textual summaries of an entire discussion. In contrast to their laborious process of manually constructing the nodes and edges, we generate them automatically and present them in an interactive visualisation. To this end, we construct a network of sentences as nodes adding edges if their pairwise semantic similarity is above a certain threshold. Therefore we use the cosine similarity of the sentence embedding vectors from fastText [22]. Syntactic edges are added between all sentences that belong to the same comment and also to its replies. Thereby, structural information of the comments and the discourse is incorporated. The resulting network is drawn using a force-based layout algorithm. As edges are hidden, the comment landscape only shows clusters of sentences depicted as dots. Since we include both semantic and syntactic edges, the layout can provide an overview of the key topics of the discussion, while preserving its overall structure.

Clustering and Classification. On the node level, the TextRank algorithm [129] ranks sentences and assigns weights to identify key statements, which we assume to be strongly connected and to form similarity communities. The clustering progressively removes edges and thereby conforms to our idea of reducing the discussion to its most essential statements for a comprehensible overview. Further, a neural network model detects *toxic* comments, such as insults or threats, which are comments that make other users leave a discussion [172]. Another neural network model from related work [173] detects *engaging* comments, such as questions or factual statements, which are likely to receive many reactions by other users.

Filtering and Merging. The class labels generated by the two neural networks are used to put more visual emphasis on the engaging comments than on the toxic comments. Nodes with a small number of edges represent sentences that are only loosely connected. In a simplified view, these nodes are either filtered completely or merged with neighbouring nodes. Nodes for sentences with almost similar embeddings are grouped together.

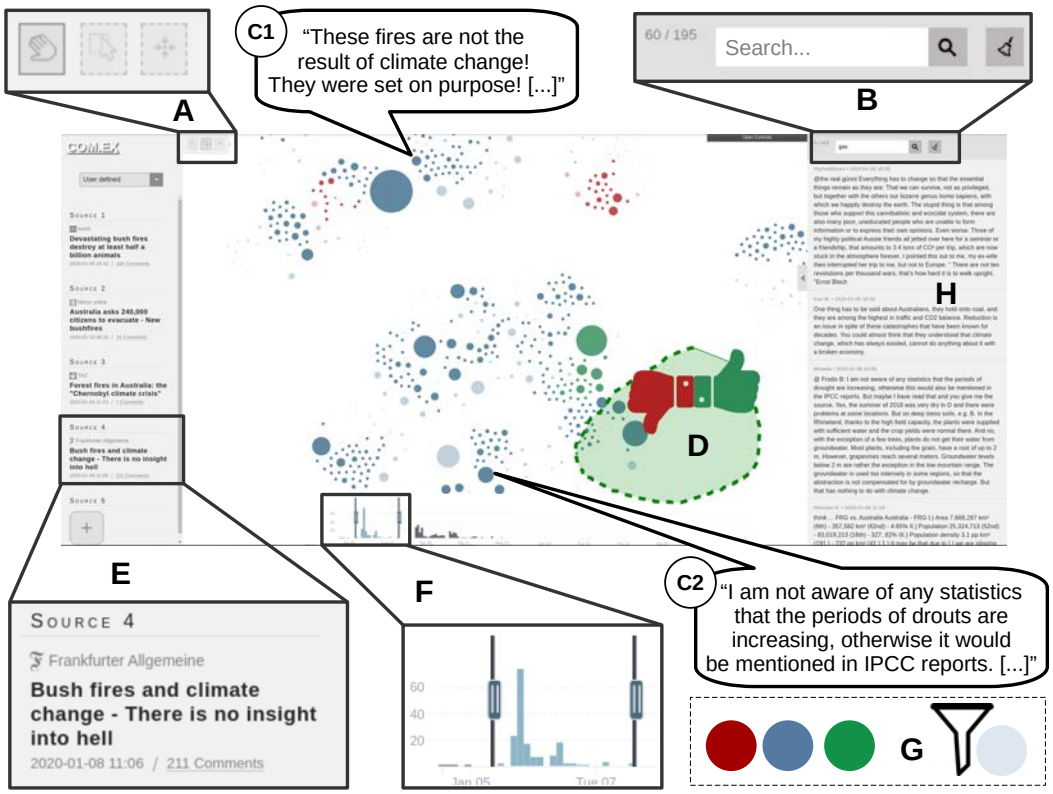


Figure 7.2.: COMEX user interface; Users select news articles (E), comments are visualised in the centre and can be filtered by time (F) or search terms (B); different modes are available (A), such as exploration by zooming, panning and reading a selected comment (H) by clicking nodes, or users can lasso-select and express their sentiment on groups of comments (D), nodes are coloured based on average sentiment and greyed out when filtered (G). Two example comments (C1, C2) are shown.

7.5. User Interface

The COMEX user interface is structured into three main components: the news outlet selection, the interactive graph, and the detailed comment view (Figure 7.2).

News Outlet Selection. The panel on the left-hand side of the interface allows selecting a set of reader discussions on articles from different news outlets. There are presets of news stories that were covered by many platforms but users are free to select (the comments of) any news article that is published on one of the seven German news platforms

currently supported. Users can add an article by simply pasting its URL. Comments on this article are then retrieved by our server and merged with previously selected comments to construct a graph representation. Mckay et al. [126] suggested to build systems that support users in reflecting on their own view by comparing it with diverse views of others. By incorporating comments from many different news outlets, we implement this design idea in the context of online discussions.

Interactive Graph. In the centre of the interface is the graph layout of all the comments. Note that the edges of the underlying graph are used only by the force-layout and are not shown for simplicity. This visualisation enables users to interact with single keyphrases of longer comments or with multiple comments at once — instead of only appending a reply to an existing list.[1] Additionally, they can rearrange or filter the nodes and navigate the canvas by zooming and panning. When using a lasso to select nodes, comments on the right panel are automatically filtered. By selecting an interval on the time histogram at the bottom, additional filters are applied. As stated before, nodes in the graph are individual sentences of comments. By clicking a node, all other nodes belonging to the comment are highlighted and the comment is shown in the right panel. Once a lasso selection is active, users can vote up or down on multiple comments to signal their agreement or disagreement. The fill colour of the nodes is updated to convey areas of predominantly positive or negative sentiment. The size of nodes can be determined by multiple factors. In Figure 7.2 the TextRank score is used, but the votes or number of replies on the originating platform has similar effects. Sliding a selection window over the time histogram shows how the discussion evolves over time. For example, it reveals which topics came up early in the course of the discussion.

Detailed Comment View. The panel on the right-hand side lists the comment texts where the text of the currently selected comment is highlighted. With a search bar, users can quickly find comments that mention keywords they are interested in. At the top of the panel are also parameter controls to adjust the number of nodes and edges displayed. This view is also updated by filters applied to the interactive graph.

Additional Possibilities. In this section, we described features of the interface we thought to be essential for exploring the comment landscape. All these features are implemented in a prototype system. Further features could be added to enable users to analyse the data in more depth. The underlying graph representation of reader comments provides the basis for additional capabilities. As the graph implicitly maintains data provenance,

[1] In the context of our demo, the effects of voting on or replying to one or multiple comments are not transmitted back to the news platforms.

Table 7.1.: Overview of available interaction features in comment sections on online news platforms *(as of October 2020)*, which are limited to upvotes, downvotes, and replies, as well as ranking comments by time or popularity with regard to the number of received upvotes or replies.

				Ranking by		
Platform	Upvotes	Downvotes	Replies	Time	Votes	Replies
Frankfurter Allg. Zeitung	✓	–	✓	✓	✓	–
Spiegel Online	✓	✓	✓	✓	✓	✓
Süddeutsche Zeitung	✓	✓	✓	✓	✓	–
Tagesschau	–	–	✓	✓	–	–
Die Welt	✓	–	✓	✓	✓	–
Die Tageszeitung	–	–	✓	✓	–	–
Zeit Online	✓	–	✓	✓	✓	–

tools for filtering comments based on meta-data is possible at all times. Furthermore, the information could also be used to control the shape, colour, or size of the visualised nodes. For example, a user might want to colour all nodes based on the news outlet the respective comments were extracted from.

7.6. Case Study

A meaningful, thorough evaluation of the proposed concepts and platform requires many active users and a sophisticated experimental setup. Such an evaluation is beyond the scope of this chapter and deferred to future work. Nevertheless, we conducted a small-scale case study to validate the presented ideas.

The system described in this chapter was designed for the purpose of visualising comments from different platforms on a single topic. We therefore use the notion of a *news story*, which is covered by several *news articles* on the same emerging news event.

Our initial findings are based on hand-selected news stories of seven different German news platforms: faz.net, tagesschau.de, spiegel.de, sz.de, taz.de, welt.de, and zeit.de. Each day between November 2019 and February 2020, we manually selected the most prevalent news stories. For each story, the annotators manually collected respective articles from the previously mentioned news platforms. The resulting dataset comprises 150 news stories and 1,350 news articles. Only 570 of these articles have publicly available reader comments, which we retrieved programmatically. In total, we retrieved 111,000 comments and the average comment length is 45 tokens. To give an example, one of the most discussed

stories in this dataset contains 4,696 comments and is covered by 4 news platforms. It is about the UN Climate Change Conference held in Madrid, 2019.

The interaction features of the seven popular German-language news outlets we selected are limited to comment replies, upvotes, and downvotes as well as ranking by time or number of votes or replies (summarised in Table 7.1). COMEX, on the other hand, could drastically change how users interact with online comments. It provides a feature-rich exploration interface for a global overview of comments from across multiple online news platforms. Going through the processing pipeline by hand, the annotators printed all comments of two exemplary stories and collaboratively assigned semantic groups similar to the argument graph described by Barker and Gaizauskas [15]. Our manual results in general confirmed the graph layout automatically generated by COMEX.

Figure 7.2 shows the interface for four articles on Australian wildfires in 2020 with 413 reader comments. If a user wants to include additional news articles, she can add a URL on the left pane (E). The COMEX system will then extract comments from the website in the background, update the comment graph and the visualisation in the centre pane. In the displayed use case, we opted to visualise topical similarity leading to clusters of topically similar comments. The cluster at the top contains comments (C1) discussing climate change, while another cluster of comments (C2) on the bottom primarily concerns droughts. The timeline at the bottom indicates the date the comments were published. By selecting a time-window (F), comments can be filtered. Comment outside the selected window are greyed out in the visualisation and removed from the comment pane (H). Users may also filter comments using full-text search (B). There are two modes (A) in which the user can interact with the data displayed in the centre pane. The first mode supports *exploration*, including zooming and panning the visualisation. Furthermore, clicking a node in the visualisation highlights the corresponding comment in the comment pane (H) and vice versa. The second mode supports *engagement*, by providing a lasso tool to select several comments at once. Users can then express their sentiment by voting up or down. We store this information and use colour (G) to indicate the average sentiment of all votes from negative (red), neutral (blue), to positive (green).

With this case study we have shown the novel way our COMEX system enables users to interact with reader comments. More evaluation is necessary to validate the user-centred aspects of being *more engaging* and *more in-depth*, and providing *more insights*.

7.7. Conclusion

To improve the way people exchange ideas online and to foster in-depth discussions, we studied the novel task of comment exploration for users of online news platforms. Previous work on conversation or discourse exploration developed analytics tools for experts.

In contrast, we focused on letting comment readers and comment writers interact with the exploration tool. To this end, we presented COMEX, a comment exploration system that implements different methodologies for the interactive analysis and visualisation of comments in online discussions.

A promising path for future work is to study the impact of novel visualisation and exploration methods on online discussions. One exemplary research question in this context would be how visualisations could help to establish a higher conversion rate of comment readers into comment writers. Potential next steps are to conduct user studies to evaluate our prototype and identify interaction patterns. The presented system is not limited to the news comments use case, but can be employed in all kinds of scenarios where user-generated content can be linked to each other.

Although some examples we examined have initially shown promising results, we see room for improvement and potential for future work in the construction and filtering of the underlying graph representation. Our modular architecture for node enrichment and edge generation and filtering allows for a simple configuration of the pipeline. One of the major challenges is to limit the number of generated edges, e.g., by introducing locality-sensitive thresholds for similarity-based edges. The visualisation itself uses a force-based layout algorithm. We experimented with several approaches to incorporate weighted aggregates of edge weights produced from different sources, i.e., for combining cluster assignment, embedding similarity, temporal proximity, and reply structure. Finding a robust and ideally self-adjusting approach remains a task for future work. Furthermore, we found, that a keyword overlay to briefly describe the “meaning” of a visual neighbourhood could be a useful addition to the interface.

Chapter 8

CONCLUSION

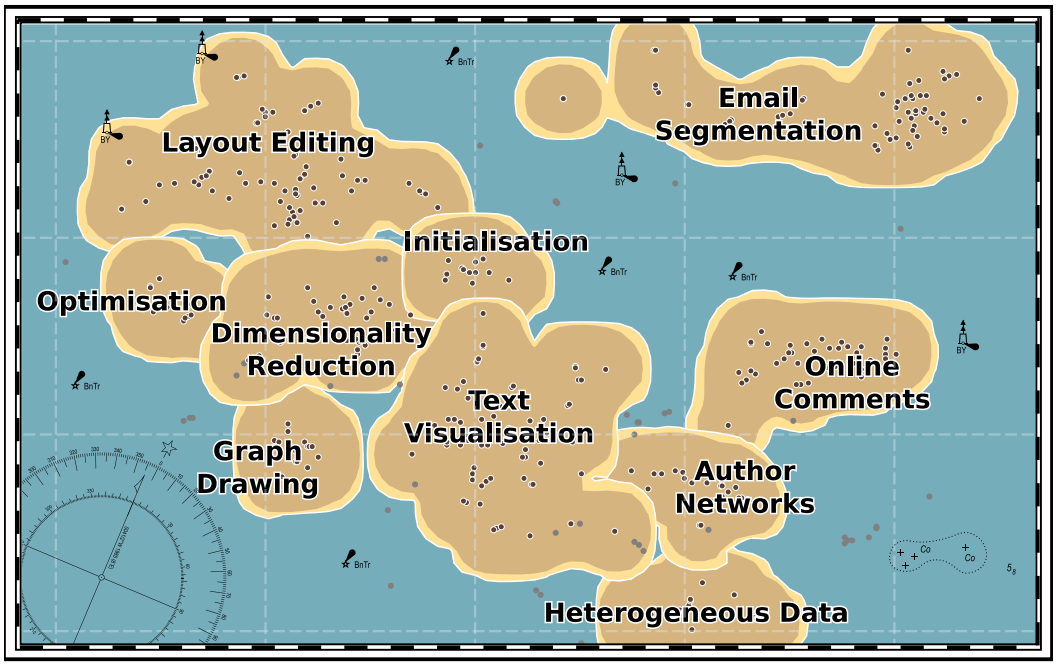


Figure 8.1.: Visualisation of this thesis, where text snippets are drawn as a dot on the map and clusters thereof form islands. Layout, formation of islands, and generation of keywords are fully data-driven. The map was later stylised by hand.

8.1. Summary

Data plays a significant role in our modern society. We use emails and social media for communication and rely on constant access to the worlds knowledge by searching online encyclopedias, digital books, and scientific or news articles. However, the vast amount of

information can often be overwhelming, especially in exploratory use-cases with little to no prior knowledge about the domain or the dataset at hand. Journalists and auditors have to sift though massive heterogeneous datasets containing emails and business documents to gain insights into potentially fraudulent or morally problematic behaviour. Although data mining technologies have proven to be powerful tools to assist the information seeking process, individually retrieved pieces of information are missing their context. Map-like visualisations of large datasets can alleviate this issue by providing an intuitive overview of the entirety of a dataset. Innovative interaction patterns allow users to explore the landscape of the data to get a better understanding of semantic relationships and salient patterns. For example, Figure 8.1 provides an artistically themed overview of the content of this thesis. Analogous to the visualisation in Section 2.7, the small dots represent chunks of 600 characters of this thesis' text that are laid out in two dimensions by preserving their pairwise semantic similarities. Clusters of these chunks form islands that are described by their most prevalent keyphrases.

In this thesis, we use machine learning to generate the layout for such map-like visualisations of textual data. While there already is related work on high-dimensional semantic text representation and dimensionality reduction to project the high-dimensional space to two dimensions to be displayed, most of them lack the ability to incorporate additional information. To this end, we proposed novel algorithms to integrate three kinds of auxiliary information, namely inherent networks, temporal data, and user feedback on existing layouts. Furthermore, we introduce domain-specific approaches for data pre-processing and representing and exploring online news comments.

Many real-world datasets contain inherent network information. Email corpora, for example, contain information about who sent what to whom and when. However, the meta-data is embedded in the semi-structured text itself and is mixed with the quoted messages as correspondents reply and forward. In order to reveal the salient structure and extract the different parts of an email, we proposed our novel neural-network-based QUAGGA algorithm. We were able to show that QUAGGA is able to identify email zones with high accuracy and is very robust noise that is commonly caused by different encodings, language settings, and email clients. For the evaluation, we introduced new annotated datasets and evaluation tasks, that have already been adapted in recent related work. We developed the BEACON platform for processing and exploring large email corpora. The ingestion pipeline automatically extracts and integrates information. To explore this integrated database, the user interface provides multiple connected views to get high-level aggregated information for the current context, which are specifically tailored to reveal patterns in the communication network or to gain insights about topic distributions. We also we proposed our multi-objective dimensionality reduction algorithm MODIR to visualise email corpora and other datasets with inherent network information such as research articles by jointly visualising the network and the semantic information in a single two-dimensional space.

We demonstrated the effectiveness of our approach on a number of real-world datasets and developed a user interface prototype to explore this novel type of visualisation.

Datasets like email corpora and other document collection grow and evolve over time. The temporal aspect can be visualised as an animation of multiple versions of the layout that gets populated with new items as they arrive. In use-cases where the visualisation has to be continuously updated, recomputing positions of the documents from scratch could result in a significantly different layout. To this end, we analysed different initialisation strategies for several state-of-the-art dimensionality reduction algorithms and introduced novel quality measures in order to evaluate the robustness and stability over time. We found, that algorithm-specific strategies usually perform better than a naive baseline.

There may be different domain-specific ways to semantically structure a dataset. Traditional dimensionality reduction algorithms however only aim at preserving pairwise similarities in the two-dimensional projection. Domain experts may want the ability to directly manipulate the layout and curate it in such a way to fit their expectation. Relocating individual items by hand quickly becomes infeasible for larger datasets. To this end, we proposed EDIMAP, an algorithm that assists the process of editing existing layouts of a dataset. Identifying the larger intent of a user from a single edit can be very challenging, which we describe in our novel taxonomy of edit intents. We were able to significantly improve the results over an approach taken from related work and introduced new metrics to measure how well the updated layout conforms with an intent.

Lastly, we introduced COMEX, a platform to explore the sometimes overwhelming amount of user comments on online news articles. Hereby, the comments are represented as nodes in a graph structure. In our case study, we described several types of edges to represent semantic and syntactic relations between comments and showcased how to visualise and interact with the graph through our user interface.

8.2. Outlook

In this thesis, we covered several novel dimensionality reduction algorithms project high-dimensional vector representations of text into a two-dimensional space for visualisation. We proposed methods that can incorporate additional information in the layout process, such as inherent networks, temporal information, and user feedback and covered domain-specific challenges in data pre-processing. Although we presented user interface prototypes, this thesis mostly considered how to place items from a dataset on a map-like visualisation. In the following, we describe perspectives for future work in this area that we identified based on our findings.

Full-text Email Parsing. With our QUAGGA algorithm for recovering the structure of full-text emails, we improved prior work in this area by considering the entire email, not just the last message in a communication chain. We also integrated our model in an open-source parsing library. Since then, others improved the cross-language capabilities and added more fine-grained email zones [18, 87]. However, parsing the in-line meta-data still remains an unsolved challenge. For the purpose of extracting the information about senders, time, and recipients, we used regular expressions in the ingestion pipeline of BEACON. The formatting in real-world datasets are often corrupted, which commonly breaks these extraction rules. Furthermore, correspondents may be represented by many different aliases as they used different email addresses or when the email client only inserts the contact name with different spellings. In order to support efforts in future work, we annotated a few hundred emails with a high level of detail, highlighting exact time and date information as well as email addresses and names including how they can be linked. Current approaches for email parsing focus on classifying lines, a future avenue to resolve the challenges described could be to develop character-level models to classify sequences of the raw text. Additionally, graph mining could be used to identify which co-occurring patterns can be used to match aliases to correspondents.

Self-explanatory Maps. In order to utilise the full potential of map-like data visualisations, it is crucial to provide concise summarising keywords to describe the different regions of a map. This enables users to get a quick overview of the different topics of the corpus and where they might find the information they are looking for. In our prototypes, we used clusters or a grid to identify which keywords or phrases are most distinctive for each region, which is done in a similar manner in related work Klouche et al. [99]. Although this is a good baseline, there are still many challenges remaining. For example, the number of displayed descriptors has to be very limited to not overwhelm users with information. Furthermore, the keywords should not overlap one another or cover the underlying map more than necessary. Lastly, they have to positioned in such a way, that they are not too far away from the documents they actually refer to. There also has to be an efficient data structure that can enable real-time updates when users zoom or pan the map to reveal keywords for the viewed context or for search results and other potential interactions. Apart from descriptive keywords, it is also important to provide visual indicators how a dataset can be clustered into topical regions. Prior work has used categorical meta-data [188] or density-based heatmaps or contours [76]. However, these are very basic approaches and rely on an additional data-source or classifier.

Utilising User Feedback. In Chapter 5, we described our novel EDIMAP algorithm for assisting users in editing existing two-dimensional layouts of a dataset. As we have

discussed there, identifying the user intent from a single edit is sheer impossible. A possible direction for future work is to develop a small selection of intuitive editing tools that users can work with and indirectly provide cues to the underlying model on how to interpret the edit. For example, prior work used pairwise constraints that users have to define ahead of time [128] to indicate similar or dissimilar pairs. Deriving these constraints from point-wise edits to the existing layouts could be one of these tools. Linear dependencies in the dimensions of word embedding vectors can be used to identify polysemy [10]. When using these embeddings, the edits in the two-dimensional space could be used to identify a subspace that would best reflect the updated neighbourhoods to recompute the layout. In this way, the layout may change significantly, but could coincide with a different overall way to interpret a dataset. Lastly, the user edits in the two-dimensional space could be propagated back into the original embedding space to update the embedding model itself and thus potentially improving its performance in other related downstream tasks.

Visualising Dynamic Datasets. There are two ways of visualising the temporal aspect of a dataset: with time itself by animation or statically using glyphs or other symbols. In traditional information-visualisation, this might be as trivial as using a line-plot instead of a point that moves up and down to show the development of stock prices over time. However, map-like visualisations already are two-dimensional, thus leaving no obvious way to directly display the trends over time. An interesting direction for future work could be the adaptation of visualisation techniques used in weather charts or scientific visualisation of flow dynamics. Although these might not be applicable for the depiction of changes over a long time period, it might be useful for differential views across multiple versions of an evolving landscape. For example, pathlines are typically used to indicate the flow of particles in physics simulations. Tracing the movement of items from the dataset would leave short traces for stable areas and longer traces for areas that are rapidly changing. A further abstraction of these dynamics would be to use symbols that are common in visualising weather systems. Stable and evolving areas could be interpreted as low- and high-pressure systems and can be drawn using known paradigms.

References

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 420–434. Springer-Verlag, 2001. doi: 10.1007/3-540-44503-X_27.

[2] A. Agrawal, A. Ali, and S. Boys. Minimum-distortion embedding. *Foundations and Trends in Machine Learning*, 14(3), 2021. doi: 10.1561/2200000090.

[3] A. Aker, E. Kurtic, A. Balamurali, M. Paramita, E. Barker, M. Hepple, and R. Gaizauskas. A graph-based approach to topic clustering for online comments to news. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 15–29. Springer-Verlag, 2016. doi: 10.1007/978-3-319-30671-1_2.

[4] N. Alder, T. Bleifuß, L. Bornemann, F. Naumann, and T. Repke. Ein Data Engineering Kurs für 10.000 Teilnehmer. *Datenbank-Spektrum*, 21(1):5–9, 2021. doi: 10.1007/s13222-020-00354-8.

[5] H. Ambavi, K. Vaishnaw, U. Vyas, A. Tiwari, and M. Singh. Covidexplorer: A multi-faceted ai-based search and visualization engine for COVID-19 information. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 3365–3368. ACM Press, 2020. doi: 10.1145/3340531.3417428. URL `https://doi.org/10.1145/3340531.3417428`.

[6] C. Ambroselli, J. Risch, R. Krestel, and A. Loos. Prediction for the newsroom: Which articles will get the most comments? In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 193–199. ACL, 2018. doi: 10.18653/v1/n18-3024.

[7] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, and O. Etzioni. Construction of the literature graph in semantic scholar. In *NAACL-HLT*, pages 84–91. ACL, 2018. doi: 10.18653/v1/n18-3011.

[8] S. An, S. Hong, and J. Sun. Viva: Semi-supervised visualization via variational autoencoders. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 22–31. IEEE, 2020. doi: 10.1109/ICDM50108.2020.00011.

[9] D. Archambault and H. C. Purchase. The "map" in the mental map: Experimental results in dynamic graph drawing. *International Journal of Human-Computer Studies*, 71(11):1044–1055, 2013. doi: 10.1016/j.ijhcs.2013.08.004.

[10] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics (TACL)*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034.

[11] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035. ACM Press, 2007. doi: 10.5555/1283383.1283494.

[12] B. Bach, N. H. Riche, C. Hurter, K. Marriott, and T. Dwyer. Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. *Transactions on Visualization and Computer Graphics (TVCG)*, 23(1):541–550, 2017. doi: 10.1109/TVCG.2016.2598958.

[13] R. Bamler and S. Mandt. Dynamic word embeddings. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 380–389. ML Research Press, 2017.

[14] D. Baranchuk, A. Babenko, and Y. Malkov. Revisiting the inverted indices for billion-scale approximate nearest neighbors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 202–216. Springer-Verlag, 2018. doi: 10.1007/978-3-030-01258-8_13.

[15] E. Barker and R. J. Gaizauskas. Summarizing multi-party argumentative conversations in reader comment on news. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12–20. ACL, 2016. doi: 10.18653/v1/w16-2802.

[16] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019. doi: 10.1038/nbt.4314.

[17] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017. doi: 10.1111/cgf.12791.

[18] J. Bevendorff, K. A. Khatib, M. Potthast, and B. Stein. Crawling and preprocessing mailing lists at scale for dialog analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1151–1158. ACL, 2020. doi: 10.18653/v1/2020.acl-main.108.

[19] Y. Bian and C. North. Deepsi: Interactive deep learning for semantic interaction. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 197–207, Geneva, Switzerland, 2021. ACM Press. doi: 10.1145/3397481.3450670.

[20] N. Bikakis and T. Sellis. Exploration and visualization in the web of big linked data: A survey of the state of the art. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 1–8. CEUR-WS.org, 2016.

[21] S. Biswal, H. Sun, B. Goparaju, M. B. Westover, J. Sun, and M. T. Bianchi. Expert-level sleep scoring with deep neural networks. *Journal of the American Medical Informatics Association*, 25(12):1643–1650, 2018. doi: 10.1093/jamia/ocy131.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146, 2017.

[23] F. Bonchi, C. Castillo, A. Gionis, and A. Jaimes. Social network analysis and mining for business applications. *Transactions on Intelligent Systems and Technology (TIST)*, 2(3):22:1–22:37, 2011. doi: 10.1145/1961189.1961194.

[24] A. Boytsov, F. Fouquet, T. Hartmann, and Y. L. Traon. Visualizing and exploring dynamic high-dimensional datasets with LION-tSNE. *CoRR*, abs/1708.04983, 2017.

[25] M. Callaghan, J. Minx, and P. Forster. A topography of climate change research. *Nature Climate Change*, 10:118–123, 2020. doi: 10.1038/s41558-019-0684-5.

[26] M. Callaghan, C.-F. Schleussner, S. Nath, Q. Lejeune, T. R. Knutson, M. Reichstein, G. Hansen, E. Theokritoff, M. Andrijevic, R. J. Brecha, et al. Machine-learning-based evidence and attribution mapping of 100,000 climate impact studies. *Nature Climate Change*, pages 966–972, 2021. doi: 10.1038/s41558-021-01168-6.

[27] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):5:1–5:51, 2015. doi: 10.1145/2733381.

[28] N. Cao, J. Sun, Y. Lin, D. Gotz, S. Liu, and H. Qu. Facetatlas: Multifaceted visualization for rich text corpora. *Transactions on Visualization and Computer Graphics (TVCG)*, 16(6):1172–1181, 2010. doi: 10.1109/TVCG.2010.154.

[29] V. Carvalho and W. Cohen. Learning to extract signature and reply lines from email. In *Proceedings of the Conference on Email and Anti-Spam (CAES)*, pages 1–8, 2004.

[30] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E. Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 377–386. ACM Press, 2017. doi: 10.1145/3132847.3132925.

[31] M.-A. Chabin. Panama papers: a case study for records management? *Brazilian Journal of Information Science: Research Trends*, 11(4):10–13, 2017. doi: 10.36311/1981-1640.2017.v11n4.03.p10.

[32] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 119–128. ACM Press, 2015. doi: 10.1145/2783258.2783296.

[33] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak. Complete gradient clustering algorithm for features analysis of x-ray images. In *Information Technologies in Biomedicine*, pages 15–24. Springer-Verlag, 2010. doi: 10.1007/978-3-642-13105-9_2.

[34] H. Chen, U. Soni, Y. Lu, V. Huroyan, R. Maciejewski, and S. G. Kobourov. Same stats, different graphs: Exploring the space of graphs in terms of graph properties. *Transactions on Visualization and Computer Graphics (TVCG)*, 27(3):2056–2072, 2021. doi: 10.1109/TVCG.2019.2946558.

[35] T. Chen, L.-A. Tang, Y. Sun, Z. Chen, H. Chen, and G. Jiang. Integrating community and role detection in information networks. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 72–80. SIAM Publications, 2016. doi: 10.1137/1.9781611974348.9.

[36] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus. *Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):1161–1168, 2009. doi: 10.1109/TVCG.2009.140.

[37] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the EMNLP Workshop on Syntax, Semantics and Structure in Statistical Translatio (SSST)*, pages 103–111. ACL, 2014. doi: 10.3115/v1/W14-4012.

[38] K. Clark, M. Luong, Q. V. Le, and C. D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–18. OpenReview.net, 2020.

[39] M. Coddington. Clarifying journalism's quantitative turn: A typology for evaluating data journalism, computational journalism, and computer-assisted reporting. *Digital Journalism*, 3(3):331–348, 2015. doi: 10.1080/21670811.2014.976400.

[40] G. Creamer, R. Rowe, S. Hershkop, and S. J. Stolfo. Segmentation and automated social hierarchy detection through email network analysis. In *SIGKDD Workshop on Web Mining and Web Usage Analysis (WebKDD)*, pages 40–58. Springer-Verlag, 2009. doi: 10.1007/978-3-642-00528-2_3.

[41] V. Crow, K. Pennock, M. Pottier, A. Schur, J. Thomas, J. Wise, D. Lantrip, T. Fiegel, C. Struble, and J. York. Multidimensional visualization and browsing for intelligence analysis. In *Proceedings of Graphics and Visualization Conference*, pages 1–9. Pacific Northwest Laboratory, 1994.

[42] W. Cui, S. Liu, Z. Wu, and H. Wei. How hierarchical topics evolve in large text corpora. *Transactions on Visualization and Computer Graphics (TVCG)*, 20(12): 2281–2290, 2014. doi: 10.1109/TVCG.2014.2346433.

[43] T. Dang and A. Forbes. Cactustree: A tree drawing approach for hierarchical edge bundling. In *Proceedings of the IEEE Pacific Visualization Conference (PacVis)*, pages 210–214. IEEE, 2017. doi: 10.1109/PACIFICVIS.2017.8031596.

[44] A. de Andrade Lopes, R. Minghim, V. V. de Melo, and F. V. Paulovich. Mapping texts through dimensionality reduction and visualization techniques for interactive exploration of document collections. In *Visualization and Data Analysis*, volume 6060 of *SPIE Proceedings*, page 60600T. SPIE, 2006. doi: 10.1117/12.650899.

[45] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186. ACL, 2019. doi: 10.18653/v1/n19-1423.

[46] B. Dharamsotu, K. S. Rani, S. A. Moiz, and C. R. Rao. k-NN sampling for visualization of dynamic data using LION-tSNE. In *Proceedings of the IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC)*, pages 63–72. IEEE, 2019. doi: 10.1109/HiPC.2019.00019.

[47] J. Diesner and K. M. Carley. Exploration of communication networks from the enron email corpus. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*. SIAM Publications, 2005.

[48] J. Ding, A. Condon, and S. P. Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature communications*, 9(1): 1–13, 2018. doi: 10.1038/s41467-018-04368-5.

[49] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 93–102. IEEE, 2012. doi: 10.1109/VAST.2012.6400485.

[50] P. Eades, Q. H. Nguyen, and S. Hong. Drawing big graphs using spectral sparsification. In *Proceedings of the International Symposium on Graph Drawing (GD)*, pages 272–286. Springer-Verlag, 2017. doi: 10.1007/978-3-319-73915-1_22.

[51] A. Efrat, Y. Hu, S. G. Kobourov, and S. Pupyrev. Mapsets: Visualizing embedded and clustered graphs. *Journal of Graph Algorithms and Applications (JGAA)*, 19 (2):571–593, 2015. doi: 10.7155/jgaa.00364.

[52] J. Ehmüller, L. Kohlmeyer, H. McKee, D. Paeschke, T. Repke, R. Krestel, and F. Naumann. Sense tree: Discovery of new word senses with graph-based scoring. In *Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen" (LWDA)*, volume 2738 of *CEUR Workshop Proceedings*, pages 246–257. CEUR-WS.org, 2020.

[53] M. El-Assady, V. Gold, C. Acevedo, C. Collins, and D. Keim. Contovi: Multi-party conversation exploration using topic-space views. *Computer Graphics Forum*, 35 (3):431–440, 2016. doi: 10.1111/cgf.12919.

[54] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI)*, pages 473–482. ACM Press, 2012. doi: 10.1145/2207676.2207741.

[55] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017. doi: 10.1038/nature21056.

[56] D. Estival, T. Gaustad, S. Pham, W. Radford, and B. Hutchinson. Author profiling for english emails. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACL)*, pages 263–272. ACL, 2007.

[57] E. Faerman, F. Borutta, K. Fountoulakis, and M. W. Mahoney. LASAGNE: Locality and structure aware graph node embedding. In *Proceedings of the International Conference on Web Intelligence (WI)*, WIC, pages 246–253. IEEE, 2018. doi: 10.1109/WI.2018.00-83.

[58] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access*, 7:56329–56340, 2019. doi: 10.1109/ACCESS.2019.2913705.

[59] W. Felger, M. Frühauf, M. Göbel, R. Gnatz, and G. Hofmann. Towards a reference model for scientific visualization systems. In *Visualization in Scientific Computing*, pages 63–74. Springer-Verlag, 1990. doi: 10.1007/978-3-642-77902-2_7.

[60] B. Fortuna, M. Grobelnik, and D. Mladenic. Visualization of text document corpus. *Informatica*, 29(4):497–502, 2005.

[61] K. Franke and S. N. Srihari. Computational forensics: Towards hybrid-intelligent crime investigation. In *Proceedings of the International Symposium on Information Assurance and Security (IAS)*, pages 383–386. IEEE, 2007. doi: 10.1109/IAS.2007.84.

[62] D. Fried and S. G. Kobourov. Maps of computer science. In *Proceedings of the IEEE Pacific Visualization Conference (PacVis)*, pages 113–120. IEEE, 2014. doi: 10.1109/PacificVis.2014.47.

[63] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. doi: 10.1002/spe.4380211102.

[64] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009. doi: 10.1111/j.1467-8659.2009.01429.x.

[65] A. Funk, A. Aker, E. Barker, M. L. Paramita, M. Hepple, and R. Gaizauskas. The sensei overview of newspaper readers' comments. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 758–761. Springer-Verlag, 2017. doi: 10.1007/978-3-319-56608-5_77.

[66] E. R. Gansner, Y. Hu, S. G. Kobourov, and C. Volinsky. Putting recommendations on the map: visualizing clusters and relations. In *Proceedings of the Conference on Recommender Systems (RecSys)*, pages 345–348. ACM Press, 2009. doi: 10.1145/1639714.1639784.

[67] H. Gibson, J. Faith, and P. Vickers. A survey of two-dimensional graph layout techniques for information visualisation. *Proceedings of the International Conference on Information Visualisation (IV)*, 12(3-4):324–357, 2013. doi: 10.1177/1473871612455749.

[68] S. Greydanus. Scaling *down* deep learning. *CoRR*, abs/2011.14439, 2020.

[69] M. Gronemann and M. Jünger. Drawing clustered graphs as topographic maps. In *Proceedings of the International Symposium on Graph Drawing (GD)*, pages 426–438. Springer-Verlag, 2012. doi: 10.1007/978-3-642-36763-2_38.

[70] L. Hajderanj, I. Weheliye, and D. Chen. A new supervised t-sne with dissimilarity measure for effective data visualization and classification. In *Proceedings of the International Conference on Software and Information Engineering (ICSIE)*, pages 232–236. ACM Press, 2019. doi: 10.1145/3328833.3328853.

[71] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin (DEB)*, 40(3):52–74, 2017.

[72] C. D. Hansen and C. R. Johnson, editors. *The Visualization Handbook*. Academic Press / Elsevier, 2005. ISBN 978-0-12-387582-2.

[73] C. Harris. Searching for diverse perspectives in news articles: Using an lstm network to classify sentiment. In *Proceedings of the Workshop on Exploratory Search and Interactive Data Analytics (ESIDA@IUI)*, 2018.

[74] Z. He, J. Liu, Y. Zeng, L. Wei, and Y. Huang. Content to node: Self-translation network embedding. *Transactions on Knowledge and Data Engineering (TKDE)*, 33(2):431–443, 2021. doi: 10.1109/TKDE.2019.2932388.

[75] G. M. H. Hilasaca and F. V. Paulovich. User-guided dimensionality reduction ensembles. In *Proceedings of the International Conference on Information Visualisation (IV)*, pages 228–233. IEEE, 2019. doi: 10.1109/IV.2019.00046.

[76] J. Hildenbrand, A. Nocaj, and U. Brandes. Flexible level-of-detail rendering for large graphs. In *Proceedings of the International Symposium on Graph Drawing (GD)*, pages 625–627. Springer-Verlag, 2016. doi: 10.1007/978-3-319-50106-2.

[77] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 833–840. MIT Press, 2003.

[78] M. Hogräfer, M. Heitzler, and H.-J. Schulz. The state of the art in map-like visualization. *Computer Graphics Forum*, 39(3):647–674, 2020. doi: 10.1111/cgf.14031.

[79] S. Hong, P. Eades, M. Torkel, W. Huang, and C. Cifuentes. Dynamic graph map animation. In *Proceedings of the IEEE Pacific Visualization Conference (PacVis)*, pages 1–6. IEEE, 2020. doi: 10.1109/PacificVis48177.2020.1042.

[80] C.-S. Hoo. Impacts of patent information on clustering in derwent innovation's themescape map. *World Patent Information*, 63:1–7, 2020. doi: 10.1016/j.wpi.2020.102001.

[81] M. E. Houle. Dimensionality, discriminability, density and distance distributions. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 468–473. IEEE, 2013. doi: 10.1109/ICDMW.2013.139.

[82] Y. Hu, S. G. Kobourov, and S. Veeramoni. Embedding, clustering and coloring for dynamic maps. *Journal of Graph Algorithms and Applications*, 18(1):77–109, 2014. doi: 10.7155/jgaa.00315.

[83] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[84] W. Huber. Dokumentation der fünf bekannten Lebendaufnahmen vom Quagga. *SPIXIANA: Zeitschrift für Zoologie*, 17(1):193–199, 1994. ISSN 0341-8391.

[85] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLoS ONE*, 9(6):1–12, 2014. doi: 10.1371/journal.pone.0098679.

[86] E. Jamison and I. Gurevych. Headerless, quoteless, but not hopeless? using pairwise email classification to disentangle email threads. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 327–335. ACL, 2013.

[87] B. Jardim, R. Rei, and M. S. C. Almeida. Multilingual email zoning. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 88–95. ACL, 2021. doi: 10.18653/v1/2021.eacl-srw.13.

[88] X. Ji, H. Shen, A. Ritter, R. Machiraju, and P. Yen. Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. *Transactions on Visualization and Computer Graphics (TVCG)*, 25(6):2181–2192, 2019. doi: 10.1109/TVCG.2019.2903946.

[89] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *Transactions on Visualization and Computer Graphics (TVCG)*, 15(6):993–1000, 2009. doi: 10.1109/TVCG.2009.153.

[90] M. W. Johnson, M. Eagle, and T. Barnes. Invis: An interactive visualization tool for exploring interaction networks. In *Proceedings of the International Conference on Educational Data Mining (EDM)*, pages 82–89. International Educational Data Mining Society, 2013.

[91] S. Joty, G. Carenini, and R. T. Ng. Topic segmentation and labeling in asynchronous conversations. *Journal of Artificial Intelligence Research*, 47:521–573, 2013. doi: 10.1613/jair.3940.

[92] H. Kang, C. Plaisant, T. Elsayed, and D. W. Oard. Making sense of archived e-mail: Exploring the enron collection with netlens. *Journal of the Association for Information Science and Technology (JASIST)*, 61(4):723–744, 2010. doi: 10.1002/asi.21275.

[93] M. Karthik, M. Marikkannan, and A. Kannan. An intelligent system for semantic information retrieval information from textual web documents. In *International Workshop on Computational Forensics (IWCF)*, pages 135–146. Springer-Verlag, 2008. doi: 10.1007/978-3-540-85303-9_13.

[94] L. Kay, A. L. Porter, J. Youtie, N. Newman, and I. Ràfols. Visual analysis of patent data through global maps and overlays. In *Current Challenges in Patent Information Retrieval*, pages 281–295. Springer-Verlag, 2017. doi: 10.1007/978-3-662-53817-3_10.

[95] T. Kellermeier, T. Repke, and R. Krestel. Mining business relationships from stocks and news. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 70–84. Springer-Verlag, 2019. doi: 10.1007/978-3-030-37720-5_6.

[96] J. S. Kessler. Scattertext: a browser-based tool for visualizing how corpora differ. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 85–90. ACL, 2017. doi: 10.18653/v1/P17-4015.

[97] Y. Kim, Y. Jernite, D. Sontag, and A. Rush. Character-aware neural language models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 2741–2749. AAAI Press, 2015.

[98] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 217–226. Springer-Verlag, 2004. doi: 10.1007/978-3-540-30115-8_22.

[99] K. Klouche, T. Ruotsalo, L. Micallef, S. Andolina, and G. Jacucci. Visual re-ranking for multi-aspect information retrieval. In *Proceedings of the Conference for Human Information Interaction and Retrieval (CHIIR)*, pages 57–66. ACM Press, 2017. doi: 10.1145/3020165.3020174.

[100] D. Kobak and G. C. Linderman. Umap does not preserve global structure any better than t-sne when using the same initialization. *bioRxiv*, 2019. doi: 10.1101/2019.12.19.877522.

[101] D. Kobak and G. C. Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, pages 156–157, 2021. doi: 10.1038/nbt.4314.

[102] D. Kobak, G. Linderman, S. Steinerberger, Y. Kluger, and P. Berens. Heavy-tailed kernels reveal a finer cluster structure in t-SNE visualisations. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 124–139. Springer-Verlag, 2019. doi: 10.1007/978-3-030-46150-8_8.

[103] L. Kohlmeyer, T. Repke, and R. Krestel. Novel views on novels: Embedding multiple facets of long texts. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 1–6. ACM Press, 2021. doi: 10.1145/3486622.3494006.

[104] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982. doi: 10.1007/BF00337288.

[105] K. Kucher, R. M. Martins, C. Paradis, and A. Kerren. Stancevis prime: visual analysis of sentiment and stance in social media texts. *Journal of Visualization*, 23 (6):1015–1034, 2020. doi: 10.1007/s12650-020-00684-5.

[106] A. Lampert, R. Dale, and C. Paris. Segmenting email message text into zones. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 919–928. ACL, 2009.

[107] A. Lampert, R. Dale, and C. Paris. Detecting emails containing requests for action. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics on Human Language Technologies (NAACL-HLT)*, pages 984–992. ACL, 2010.

[108] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 331–339. Morgan Kaufman, 1995. doi: 10.1016/b978-1-55860-377-6.50048-7.

[109] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1188–1196. JMLR Inc. and Microtome Publishing, 2014.

[110] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[111] B. Lee, N. H. Riche, A. K. Karlson, and S. Carpendale. Sparkclouds: Visualizing trends in tag clouds. *Transactions on Visualization and Computer Graphics (TVCG)*, 16(6):1182–1189, 2010. doi: 10.1109/TVCG.2010.194.

[112] S. Lespinats and M. Aupetit. CheckViz: Sanity check and topological clues for linear and non-linear mappings. *Computer Graphics Forum*, 30(1):113–125, 2011. doi: 10.1111/j.1467-8659.2010.01835.x.

[113] A. Lhuillier, C. Hurter, and A. C. Telea. State of the art in edge and trail bundling techniques. *Computer Graphics Forum*, 36(3):619–645, 2017. doi: 10.1111/cgf.13213.

[114] S. Z. Li, Z. Zhang, and L. Wu. Markov-lipschitz deep learning. *CoRR*, abs/2006.08256:1–18, 2020.

[115] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell rna-seq data. *Nature methods*, 16(3):243–245, 2019. doi: 10.1038/s41592-018-0308-4.

[116] S. Liu, X. Wang, C. Collins, W. Dou, F. Ou-Yang, M. El-Assady, L. Jiang, and D. A. Keim. Bridging text visualization and mining: A task-driven survey. *Transactions on Visualization and Computer Graphics (TVCG)*, 25(7):2482–2504, 2019. doi: 10.1109/TVCG.2018.2834341.

[117] S. Lohmann, J. Ziegler, and L. Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*, pages 392–404. Springer-Verlag, 2009. doi: 10.1007/978-3-642-03655-2_43.

[118] M. Loster, T. Repke, R. Krestel, F. Naumann, J. Ehmueller, B. Feldmann, and O. Maspfuhl. The challenges of creating, maintaining and exploring graphs of financial entities. In *Proceedings of the International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets, DSMM@SIGMOD*, pages 6:1–6:2. ACM Press, 2018. doi: 10.1145/3220547.3220553.

[119] J. Ma, R. Wang, W. Ji, J. Zhao, M. Zong, and A. Gilman. Robust multi-view continuous subspace clustering. *Pattern Recognition Letters*, 150:306–312, 2021. doi: 10.1016/j.patrec.2018.12.004.

[120] X. Ma and E. H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1064–1074. ACL, 2016. doi: 10.18653/v1/p16-1101.

[121] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9:2579–2605, 2008.

[122] D. Mashima, S. G. Kobourov, and Y. Hu. Visualizing dynamic data with maps. *Transactions on Visualization and Computer Graphics (TVCG)*, 18(9):1424–1437, 2012. doi: 10.1109/TVCG.2011.288.

[123] L. McInnes and J. Healy. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018.

[124] L. McInnes, J. Healy, N. Saul, and L. Großberger. UMAP: uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861.

[125] S. J. McIntosh and D. Bainbridge. An integrated interactive and persistent map-based digital library interface. In *Proceedings of the International Conference on Asian Digital Libraries (ICADL)*, pages 321–330. Springer-Verlag, 2011. doi: 10.1007/978-3-642-24826-9_40.

[126] D. Mckay, S. Makri, M. Gutierrez-Lopez, A. MacFarlane, S. Missaoui, C. Porlezza, and G. Cooper. We are the change that we seek: Information interactions during a change of viewpoint. In *Proceedings of the Conference for Human Information Interaction and Retrieval (CHIIR)*, page 173–182. ACM Press, 2020. doi: 10.1145/3343413.3377975.

[127] A. Mead. Review of the development of multidimensional scaling methods. *The Statistician*, 41(1):27–39, 1992. doi: 10.2307/2348634.

[128] M. Meng, J. Wei, J. Wang, Q. Ma, and X. Wang. Adaptive semi-supervised dimensionality reduction based on pairwise constraints weighting and graph optimizing. *International Journal of Machine Learning and Cybernetics*, 8(3):793–805, 2017. doi: 10.1007/s13042-015-0380-3.

[129] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 404–411, 2004.

[130] K. Ø. Mikalsen, C. Soguero-Ruíz, F. M. Bianchi, and R. Jenssen. Noisy multi-label semi-supervised dimensionality reduction. *Pattern Recognition*, 90:257–270, 2019. doi: 10.1016/j.patcog.2019.01.033.

[131] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119. NIPS Foundation, Inc., 2013.

[132] R. Minghim, F. V. Paulovich, and A. de Andrade Lopes. Content-based text mapping using multi-dimensional projections for exploration of document collections. In *Visualization and Data Analysis*, volume 6060 of *SPIE Proceedings*, page 60600S. SPIE, 2006. doi: 10.1117/12.650880.

[133] B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer. Visualizing the quality of dimensionality reduction. *Neurocomputing*, 112:109–123, 2013. doi: 10.1016/j.neucom.2012.11.046.

[134] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7045–7054. JMLR Inc. and Microtome Publishing, 2020.

[135] G. Mujtaba, L. Shuib, R. Raj, N. Majeed, and M. Al-Garadi. Email classification research trends: Review and open issues. *IEEE Access*, 5:9044–9064, 2017. doi: 10.1109/ACCESS.2017.2702187.

[136] A. Narayan, B. Berger, and H. Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 39(6):765–774, 2021. doi: 10.1038/s41587-020-00801-7.

[137] Y. Nedumov, A. Babichev, I. Mashonsky, and N. Semina. Scinoon: Exploratory search system for scientific groups. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*. CEUR-WS.org, 2019.

[138] Y. Nedumov, A. Babichev, I. Mashonsky, and N. Semina. Scinoon: Exploratory search system for scientific groups. In *Proceedings of the Workshop on Exploratory Search and Interactive Data Analytics (ESIDA@IUI)*, pages 1–6. CEUR-WS.org, 2019.

[139] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 849–856. MIT Press, 2001.

[140] N. V. T. Nguyen, V. T. Nguyen, V. Pham, and T. Dang. Finanviz: Visualizing emerging topics in financial news. In *Proceedings of the International Conference on Big Data (BigData)*, pages 4698–4704. IEEE, 2018. doi: 10.1109/BigData.2018.8622097.

[141] Q. H. Nguyen, P. Eades, and S. Hong. Towards faithful graph visualizations. *CoRR*, abs/1701.00921, 2017.

[142] A. Noack. *Unified quality measures for clusterings, layouts, and orderings of graphs, and their application as software design criteria*. PhD thesis, Brandenburg University of Technology, Cottbus-Senftenberg, Germany, 2007. Chapter 6, pp97.

[143] M. Noichl. Modeling the structure of recent philosophy. *Synthese*, 198(6):5089–5100, 2021. doi: 10.1007/s11229-019-02390-8.

[144] D. Oard, W. Webber, D. Kirsch, and S. Golitsynskiy. Avocado research email collection. *Linguistic Data Consortium*, 2015. doi: 10.35111/wqt6-jg60.

[145] E. Palagi, F. Gandon, A. Giboin, and R. Troncy. A survey of definitions and models of exploratory search. In *Proceedings of the Workshop on Exploratory Search and Interactive Data Analytics (ESIDA@IUI)*, pages 3–8. ACM Press, 2017. doi: 10.1145/3038462.3038465.

[146] P. C.-I. Pang, R. P. Biuk-Aghai, M. Yang, and B. Pang. Creating realistic map-like visualisations: Results from user studies. *Journal of Visual Languages and Computing (JVLC)*, 43:60–70, 2017. doi: 10.1016/j.jvlc.2017.09.002.

[147] D. Paurat and T. Gärtner. Invis: A tool for interactive visual data analysis. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 672–676. Springer-Verlag, 2013. doi: 10.1007/978-3-642-40994-3_52.

[148] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720.

[149] J. Peltonen, Z. Lin, K. Järvelin, and J. Nummenmaa. Pihvi: Online forum posting analysis with interactive hierarchical visualization. In *Proceedings of the Workshop on Exploratory Search and Interactive Data Analytics (ESIDA@IUI)*, pages 1–8. CEUR-WS.org, 2018.

[150] A. Perer and B. Shneiderman. Beyond threads: Identifying discussions in email archives. Technical report, University of Maryland, 2005.

[151] N. Pezzotti, B. P. Lelieveldt, L. van der Maaten, T. Höllt, E. Eisemann, and A. Vilanova. Approximated and user steerable tSNE for progressive visual analytics. *Transactions on Visualization and Computer Graphics (TVCG)*, 23(7):1739–1752, 2017. doi: 10.1109/TVCG.2016.2570755.

[152] R. Pienta, J. Abello, M. Kahng, and D. H. Chau. Scalable graph exploration and visualization: Sensemaking challenges and opportunities. In *Proceedings of the International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 271–278. IEEE, 2015. doi: 10.1109/35021BIGCOMP.2015.7072812.

[153] P. G. Poličar, M. Strazar, and B. Zupan. Embedding to reference t-SNE space addresses batch effects in single-cell classification. In *Discovery Science - International Conference DS*, pages 246–260. Springer-Verlag, 2019. doi: 10.1007/978-3-030-33778-0_20.

[154] P. G. Poličar, M. Stražar, and B. Zupan. openTSNE: a modular python library for t-SNE dimensionality reduction and embedding. *bioRxiv*, 2019. doi: 10.1101/731877.

[155] R. E. Prasojo, M. Kacimi, and W. Nutt. Entity and aspect extraction for organizing news comments. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 233–242. ACM Press, 2015. doi: 10.1145/2806416.2806576.

[156] H. C. Purchase. Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing (JVLC)*, 13(5):501–516, 2002. doi: 10.1006/jvlc.2002.0232.

[157] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):1–24, 2019.

[158] P. E. Rauber, A. X. Falcão, and A. C. Telea. Visualizing time-dependent data using dynamic t-SNE. In *Proceedings of the Conference on Visualization (EuroVis)*, pages 73–77. Eurographics Association, 2016. doi: 10.2312/eurovisshort.20161164.

[159] F. Rauscher, N. Matta, and H. Atifi. Context aware knowledge zoning: Traceability and business emails. In *Proceedings of the International Workshop on Artificial Intelligence for Knowledge Management (AI4KM)*, pages 66–79. Springer-Verlag, 2015. doi: 10.1007/978-3-319-55970-4_5.

[160] E. Reif, A. Yuan, M. Wattenberg, F. B. Viégas, A. Coenen, A. Pearce, and B. Kim. Visualizing and measuring the geometry of BERT. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 8592–8600. NIPS Foundation, Inc., 2019.

[161] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3980–3990. ACL, 2019. doi: 10.18653/v1/D19-1410.

[162] T. Repke and R. Krestel. Topic-aware network visualisation to explore large email corpora. In *International Workshop on Big Data Visual Exploration and Analytics (BigVis)*, Proceedings of the International Conference on Extending Database Technology (EDBT), pages 104–107. CEUR-WS.org, 2018.

[163] T. Repke and R. Krestel. Bringing back structure to free text email conversations with recurrent neural networks. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 114–126. Springer-Verlag, 2018. doi: 10.1007/978-3-319-76941-7_9.

[164] T. Repke and R. Krestel. Exploration interface for jointly visualised text and graph data. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 73–74. ACM Press, 2020. doi: 10.1145/3379336.3381470.

[165] T. Repke and R. Krestel. Visualising large document collections by jointly modeling text and network structure. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, pages 279–288. ACM Press, 2020. doi: 10.1145/3383583.3398524.

[166] T. Repke and R. Krestel. Extraction and representation of financial entities from text. In S. Consoli, D. Reforiato, and M. Saisana, editors, *Data Science for Economics and Finance – Methodologies and Applications*, chapter 1, pages 1–24. Springer-Verlag, 2020.

[167] T. Repke and R. Krestel. Interactive curation of semantic representations in digital libraries. In *Proceedings of the International Conference on Asian Digital Libraries (ICADL)*, Lecture Notes in Computer Science (LNCS), pages 1–10. Springer-Verlag, 2021. doi: 10.1007/978-3-030-91669-5_18.

[168] T. Repke and R. Krestel. Robust visualisation of dynamic text collections: Measuring and comparing dimensionality reduction algorithms. In *Proceedings of the Conference for Human Information Interaction and Retrieval (CHIIR)*, pages 255–259. ACM Press, 2021. doi: 10.1145/3406522.3446034.

[169] T. Repke, M. Loster, and R. Krestel. Comparing features for ranking relationships between financial entities based on text. In *Proceedings of the International Workshop on Data Science for Macro-Modeling with Financial and Economic Datasets (DSMM@SIGMOD)*, pages 12:1–12:2. ACM Press, 2017. doi: 10.1145/3077240.3077252.

[170] T. Repke, R. Krestel, J. Edding, M. Hartmann, J. Hering, D. Kipping, H. Schmidt, N. Scordialo, and A. Zenner. Beacon in the dark: A system for interactive exploration of large email corpora. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 1871–1874. ACM Press, 2018. doi: 10.1145/3269206.3269231.

[171] P. Riehmann, D. Kiesel, M. Kohlhaas, and B. Froehlich. Visualizing a thinker's life. *Transactions on Visualization and Computer Graphics (TVCG)*, 25(4):1803–1816, 2018. doi: 10.1109/TVCG.2018.2824822.

[172] J. Risch and R. Krestel. Bagging BERT models for robust aggression identification. In *Proceedings of the Workshop on Trolling, Aggression and Cyberbullying (TRAC@LREC)*, pages 55–61. European Language Resources Association (ELRA), 2020.

[173] J. Risch and R. Krestel. Top comment or flop comment? predicting and explaining user engagement in online news discussions. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*, pages 579–589. AAAI Press, 2020.

[174] J. Risch, T. Repke, L. Kohlmeyer, and R. Krestel. ComEx: Comment exploration on online news platforms. In *Joint Proceedings of the Workshops co-located with Conference on Intelligent User Interfaces (IUI)*, pages 1–7. CEUR-WS.org, 2021.

[175] R. C. Roberts and R. S. Laramee. Visualising business data: A survey. *Information*, 9(11):285, 2018. doi: 10.3390/info9110285.

[176] A. Rule, J.-P. Cointet, and P. S. Bearman. Lexical shifts, substantive changes, and continuity in state of the union discourse, 1790–2014. *Proceedings of the National*

Academy of Sciences (PNAS), 112(35):10837–10844, 2015. doi: 10.1073/pnas.1512221112.

[177] T. Sainburg, L. McInnes, and T. Q. Gentner. Parametric UMAP: learning embeddings with deep neural networks for representation and semi-supervised learning. *CoRR*, abs/2009.12981, 2020.

[178] T. Sainburg, M. Thielk, and T. Q. Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10):1–48, 2020. doi: 10.1371/journal.pcbi.1008228.

[179] B. Saket, A. Endert, and T. Rhyne. Demonstrational interaction for data visualization. *IEEE Computer Graphics and Applications*, 39(3):67–72, 2019. doi: 10.1109/MCG.2019.2903711.

[180] B. Saket, S. Huron, C. Perin, and A. Endert. Investigating direct manipulation of graphical encodings as a method for user interaction. *Transactions on Visualization and Computer Graphics (TVCG)*, 26(1):482–491, 2020. doi: 10.1109/TVCG.2019.2934534.

[181] A. Sallaberry, Y.-c. Fu, H.-C. Ho, and K.-L. Ma. Contact trees: Network visualization beyond nodes and edges. *PLoS ONE*, 11(2):e0146368, 2016. doi: 10.1371/journal.pone.0146368.

[182] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan. Phishing email detection using natural language processing techniques: A literature survey. In *Proceedings of the International Conference on AI in Computational Linguistics (ACLing)*, pages 19–28. Elsevier, 2021. doi: 10.1016/j.procs.2021.05.077.

[183] J. W. Sammon. A nonlinear mapping for data structure analysis. *Transactions on Computers*, 18(5):401–409, 1969. doi: 10.1109/T-C.1969.222678.

[184] S. Scerri, G. Gossen, B. Davis, and S. Handschuh. Classifying action items for semantic email. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3324–3330. European Language Resources Association, 2010. ISBN 2-9517408-6-7.

[185] J. Schlötterer, C. Seifers, and M. Granitzer. On joint representation learning of network structure and document content. In *International IFIP Cross Domain Conference for Machine Learning & Knowledge Extraction (CD-MAKE)*, pages 237–251. Springer-Verlag, 2017. doi: 10.1007/978-3-319-66808-6_16.

[186] B. Schmidt. Stable random projection: Minimal, universal dimensionality reduction for library-scale data. In *International Conference of the Alliance of Digital Humanities Organizations (DH)*, pages 1–3. Alliance of Digital Humanities Organizations (ADHO), 2017.

[187] R. Schwanhold, T. Repke, and R. Krestel. Modeling the evolution of word senses with force-directed layouts of co-occurrence networks. In *Proceedings of the International Workshop on Computational Approaches to Historical Language Change (LChange@ACL)*, pages 58–63. ACL, 2021. doi: 10.18653/v1/2021.lchange-1.8.

[188] S. Sen, A. B. Swoap, Q. Li, B. Boatman, I. Dippenaar, R. Gold, M. Ngo, S. Pujol, B. Jackson, and B. Hecht. Cartograph: Unlocking spatial visualization through semantic enhancement. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 179–190. ACM Press, 2017. doi: 10.1145/3025171.3025233.

[189] E. Sherkat, S. Nourashrafeddin, E. E. Milios, and R. Minghim. Interactive document clustering revisited: A visual analytics approach. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pages 281–292. ACM Press, 2018. doi: 10.1145/3172944.3172964.

[190] X. Shi and P. S. Yu. Dimensionality reduction on heterogeneous feature space. In *Proceedings of the International Conference on Data Mining (ICDM)*, pages 635–644. IEEE, 2012. doi: 10.1109/ICDM.2012.30.

[191] R. M. Shiffrin and K. Börner. Mapping knowledge domains. *Proceedings of the National Academy of Sciences (PNAS)*, 101:5183–5185, 2004. doi: 10.1073/pnas.0307852100.

[192] A. Sinitsin, V. Plokhotnyuk, D. Pyrkin, S. Popov, and A. Babenko. Editable neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12. OpenReview.net, 2020.

[193] D. Spathis, N. Passalis, and A. Tefas. Interactive dimensionality reduction using similarity projections. *Knowledge-Based Systems*, 165:77–91, 2019. doi: 10.1016/j.knosys.2018.11.015.

[194] D. A. Szafir, D. Stuffer, Y. Sohail, and M. Gleicher. Textdna: Visualizing word usage with configurable colorfields. *Computer Graphics Forum*, 35(3):421–430, 2016. doi: 10.1111/cgf.12918.

[195] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 990–998. ACM Press, 2008. doi: 10.1145/1401890.1402008.

[196] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 1067–1077. ACM Press, 2015. doi: 10.1145/2736277.2741093.

[197] J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 287–297. ACM Press, 2016. doi: 10.1145/2872427.2883041.

[198] J. Tao, J. Xu, C. Wang, and N. V. Chawla. Honvis: Visualizing and exploring higher-order networks. In *Proceedings of the IEEE Pacific Visualization Conference (PacVis)*, pages 1–10. IEEE, 2017. doi: 10.1109/PACIFICVIS.2017.8031572.

[199] W. Tao, X. Liu, Y. Wang, L. Battle, Ç. Demiralp, R. Chang, and M. Stonebraker. Kyrix: Interactive pan/zoom visualizations at scale. *Computer Graphics Forum*, 38 (3):529–540, 2019. doi: 10.1111/cgf.13708.

[200] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. doi: 10.1126/science.290.5500.2319.

[201] D. Tien Nguyen, S. Joty, B. El Amel Boussaha, and M. de Rijke. Thread reconstruction in conversational data using neural coherence models. In *Proceedings of the Workshp on Neural Information Retrieval (Neu-IR)*, pages 1–5. HAL, 2018.

[202] N. J. Van Eck and L. Waltman. Visualizing bibliometric networks. In *Measuring Scholarly Impact*, pages 285–320. Springer-Verlag, 2014. doi: 10.1007/978-3-319-10377-8_13.

[203] Y. Wang, Z. Jin, Q. Wang, W. Cui, T. Ma, and H. Qu. DeepDrawing: A deep learning approach to graph drawing. *Transactions on Visualization and Computer Graphics (TVCG)*, 26(1):676–686, 2020. doi: 10.1109/TVCG.2019.2934798.

[204] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):63:1–63:34, 2020. doi: 10.1145/3386252.

[205] Y.-C. Wang, M. Joshi, W. W. Cohen, and C. P. Rosé. Recovering implicit thread structure in newsgroup style conversations. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*, pages 152–160. AAAI Press, 2008.

[206] J. Wawrzinek, S. A. R. Hussaini, O. Wiehr, J. M. G. Pinto, and W. Balke. Explainable word-embeddings for medical digital libraries – A context-aware approach. In *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, pages 299–308. ACM Press, 2020. doi: 10.1145/3383583.3398522.

[207] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. TIARA: a visual exploratory text analytic system. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 153–162. ACM Press, 2010. doi: 10.1145/1835804.1835827.

[208] J. A. Wise. The ecological approach to text visualization. *Journal of the American Society for Information Science (JASIST)*, 50(13):1224–1233, 1999. doi: 10.1002/(SICI)1097-4571(1999)50:13<1224::AID-ASI8>3.0.CO;2-4.

[209] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[210] J. Yang and J. Leskovec. Overlapping communities explain core–periphery publisher of networks. *Proceedings of the IEEE*, 102(12):1892–1902, 2014. doi: 10.1109/JPROC.2014.2364018.

[211] L. Yang, S. Dumais, P. Bennett, and A. Awadallah. Characterizing and predicting enterprise email reply behavior. In *Proceedings of the ACM SIGIR Conference on Information Retrieval (SIGIR)*, pages 235–244. ACM Press, 2017. doi: 10.1145/3077136.3080782.

[212] J. Yeh and A. Hamly. Email thread reassembly using similary matching. In *Proceedings of the Conference on Email and Anti-Spam (CAES)*, pages 1–8, 2006. doi: 10.7916/D8J3921Q.

[213] V. Yoghourdjian, T. Dwyer, K. Klein, K. Marriott, and M. Wybrow. Graph thumbnails: Identifying and comparing multiple graphs at a glance. *Transactions on Visualization and Computer Graphics (TVCG)*, 24(12):3081–3095, 2018. doi: 10.1109/TVCG.2018.2790961.

[214] A. Zhang, L. Garcia-Pueyo, J. B. Wendt, M. Najork, and A. Broder. Email category prediction. In *Proceedings of the International World Wide Web Conference (WWW)*, pages 495–503. ACM Press, 2017. doi: 10.1145/3041021.3055166.

[215] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *Transactions on Big Data (TBD)*, 6(1):3–28, 2020. doi: 10.1109/TBDATA.2018.2850013.

[216] Y. Zhang, J. Li, Y. Song, and C. Zhang. Encoding conversation context for neural keyphrase extraction from microblog posts. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1676–1686. ACL, 2018. doi: 10.18653/v1/n18-1151.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift selbständig und ohne unerlaubte Hilfe angefertigt sowie nur die angegebene Literatur verwendet habe,
- die Dissertation keiner anderen Hochschule in gleicher oder ähnlicher Form vorgelegt wurde,
- mir die Promotionsordnung der Digital Engineering Fakultät der Universität Potsdam vom 27. November 2019 bekannt ist.

Tim Repke – 25. November 2021