Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Visual Exploration of Topics in Multimedia News Corpora

Osaro Gabriel

Course of Study: INFOTECH

Examiner: Prof. Dr. Thomas Ertl

Supervisor: Markus John, M.Sc.,
Dipl.-Inf. Kuno Kurzhals

Commenced: February 2, 2018

Completed: May 23, 2018

CR-Classification: H.5.2, I.5.2

Abstract

As news contents grow daily, the demand for tools to help users make sense of large document corpus will continuously be on the increase. Such tools will particularly be useful for journalist and ordinary users who intend to explore large collection of news documents for various analytical tasks. When users attempt to explore documents, they are usually in search for a particular topic of interest, or to compare various topics for similarity, or to see when in time a particular topic was discussed or to explore the distribution of a topic over time or to see how frequent a particular topic was discussed in the corpus or in general to test a particular hypothesis. Existing tools fall short in providing effective and suitable interaction mechanism to enable users answer these questions in a single application framework.

In this paper we presented a framework that gives users the opportunity to easily answer questions relating to their exploratory tasks. We developed new visual elements and augment them with existing interfaces to provide users with ample options and flexibility to explore multimedia news corpus from different angles depending on their analytic tasks. Our method uses machine learning for topic extraction, clustering and word cloud generation. Our approach effectively combines both overview + detail and focus + context schemes to enrich users experience with exploring large collection of multimedia news documents. Our framework ensures synchronization of the various visual interfaces to provide immediate feedback on user's interactions. To demonstrate the effectiveness of our approach, we presented some realistic use cases from the perspective of a news analyst. And based on our observations, we identified some possible directions for future studies.

Contents

List of Figures

List of Abbreviations

CSM comma-separated-values. 34

CTM Correlated Topic Model. 21

DAG directed acyclic graph. 22

DOI Degree of Interest. 25

HLDA Hierarchical Latent Dirichlet Allocation. 22

HPAM Hierarchical Pachinko Alloation Model. 22

IDF inverse document frequency. 17

InfoVis Information Visualization. 33

LDA Latent Dirichlet Allocation. 20

MDS Multi-Dimensional Scaling. 19

NLP Natural Language Processing. 11

PAM Pachinko Alloation Model. 21

PCA Principal Component Analysis. 19

SQL Structured Query Language. 34

t-SNE t-Distributed Stochastic Neighbor Embedding. 19

TF-IDF Term Frequency - Inverse Document Frequency. 17

1 Introduction

There have been a growing number of researches in the area of visual analytics of news corpus. This is as a result of the immediate benefits derived from visualizing large text document using various visual techniques which offer insight into the nature of the document under consideration. To make sense of the visualization effort, Natural Language Processing (NLP) techniques are used to pre-process the text corpus to provide and extract meaningful structure out of the data such as feature extraction, topic segmentation, topic modeling, hierarchical clustering, text summarization, etc.

Application of visual analytics to exploration of news corpus presents lots of benefits for users as it can aid users to quickly scan through the corpus to see important events in the massive document collection. As news contents grow daily, the demand for tools to help users make sense of large document corpus will continuously be on the increase. Such tools will particularly be useful for journalist and ordinary users who intend to explore large collection of news documents for various analytical tasks. When users attempt to explore documents, they are usually in search for a particular topic of interest, or to compare various topics for similarity, or to see when in time a particular topic was discussed or to explore the distribution of a topic over time or to see how frequent a particular topic was discussed in the corpus or in general to test a particular hypothesis.

To aid users in answering these questions, there is a need to combine different visual analytic frameworks for detail exploratory analysis. This combination draws from the strengths of each framework by making use of the specific features they offer. The frameworks used in this work are carefully selected based on their suitability in the area of analyzing news corpus. To answer the question of searching for topics, one approach to assist user to achieve this task is the use of the Lens concept [19]. With the Lens, users can quickly scan through a scatterplot of the document to see the important topics discussed within a particular cluster of choice. To help users gain a quick overview of the extracted data at different aggregation levels, we used a slider visual control based on the smoothscroll framework [42]. The smoothscroll control provides a slider like interface for quick navigation of the aggregated data by allowing users to select topics of interest for further comparison. With the smoothscroll control, users can easily navigate the data at different time frames such as year, months, weeks and days. By combining the lens and slider visual elements we offer users the opportunity to explore the corpus from different visual perspectives by utilizing the advantages from both worlds.

To support further exploration of the data beyond the features offered by the Lens and slider, we designed and implemented additional frameworks. For instance, the extracted topics need to be presented in a visual context that shows the hierarchical structure of the topics and allows easy navigation of the data. Also a view to visually compare several topics and events in the dataset for relevance, similarity or both was also developed. This way, users could easily see the details of each news contents in order to confirm if their questions are answered or verify the hypothesis set out at the beginning of their exploratory task. Furthermore, our design ensured that the various views are linked and coordinated in such a way that user action in one view is automatically propagated to the rest of the views. With this, all views are updated to show the current action of the user, thus further simplifying the user's effort in exploring the document.

In this report, we discussed the implementation of our framework and the model we adopted for components synchronization. In chapter 2 we examine the theoretical background of the work covering topics such as text processing, feature generation, dimension reduction, topic modeling, and visualization techniques. In chapter 3 we review existing literatures on this subject. Chapter 4 discusses our methodology. In chapter 5 we presented a detail description of the system architecture including the concept behind the design, the process flow, data model/dataset, different views, etc. Chapter 6 discusses the design considerations adopted during the implementation. In chapter 7 we test the efficacy of our implementation by demonstrating some realistic use cases that prove the effectiveness of our design. In chapter 8 we present a discussion of our findings based on our observations from the use cases. Chapter 9 concludes the report and gives possible directions for future study.

2 Theoretical Background

2.1 Text Mining And Processing

Text mining is seen as extracting previously unknown information from text [18]. However, in the context of news corpus, text mining could be defined as retrieving relevant and meaningful information from a large collection of news data through the application of various NLP techniques.

Raw texts from documents are usually filled with unimportant information which could be seen as garbage. To achieve better results from different clustering algorithms, it is necessary to clean the text to get rid of this garbage that could adversely affect the accuracy of intended algorithms. We are most likely going to get results filled with garbage information if we feed our algorithms with texts cluttered with garbage, just like garbage in garbage out. Hence, the cleaner the text the better the results we get from our clustering algorithms, because the quality of the clustering results is dependent on the quality of the input texts.

Text processing involves series of steps performed to ensure that the raw text is transformed to a form suitable for the program to work with. Text corpora needs to be refined and cleaned before attempting to perform knowledge retrieval. This is necessary as most raw data is usually filled with words which are unimportant for the final processing. These words could potentially reduce the accuracy of the results if not eliminated. It is also important that words be reduced to their base form, a process known as stemming, for better results. Text processing and refinement involve a number of steps which includes tokenization, stemming, lemmatization, and stopwords removal.

2.1.1 Tokenization

This involves breaking sentences into individual word known as tokens. Tokenization is the first step undertaken in any NLP tasks [39]. It involves splitting the corpus into basic units or tokens which do not require further decomposition. Without this basic units or tokens it is difficult to carry out further processing on the corpus. Tokenizing texts usually involve splitting sentences by using some form of delimiters such as white space.

In our application, the use of white space suffices as a delimiter, since the language of our corpus is in German and the German language naturally uses white space to delimit words.

2.1.2 Stemming and Lemmatization

Stemming reduces a word to its base or root form. For example, the German words, *aufenthalten* and *aufenthaltes* are both reduced to the base form *aufenthalt*. Lemmatization reduces a word to its base form or lemma, but unlike stemming, lemmatization does this by considering the meaning of the word in a sentence or its morphological analysis. Stemming and lemmatization are the first steps performed towards normalizing the input tokens by reducing the size of our dictionary, creating a more compact and dense vector representation of our tokens and thus reducing the dimension of our input data. These benefits translate to immediate improvement on the results of our clustering algorithms.

2.1.3 Stopwords removal

Stopwords are unimportant words in a text corporal whose absence do not impact the overall meaning of a sentence. They are usually very frequent in a text and their presence could reduce the quality of the output results. Stopwords includes common words such as determinant, articles, pronounces, punctuations, etc. Stopwords, if not removed from the corpus, usually litter the similarity matrix, increases the size of the resulting vector space and eventually fill the results of our clustering algorithms, thus providing no meaningful information from the algorithms.

2.2 Document Clustering

2.2.1 Types of learning

Two types of learning are prevalent in knowledge retrieval; supervised and unsupervised learning.

2.2.2 Supervised Learning

Supervised learning is the process of identifying structure within text from labeled data. The output is guided by training a set of input data and then expects the algorithms to identify similar structure from unknown data. This involves labeling the data so that the algorithms could learn and classify similar or unknown input data into the labeled category. Supervised learning involves classification [35].

2.2.3 Unsupervised Learning

In Unsupervised learning, the algorithms have no prior training on the nature of the input data. This means that the input data is not labeled and the algorithm is expected to classify them based only on textual properties of the input text [35]. Unsupervised learning involves clustering.

Clustering Algorithms: Clustering is the processing of sorting and grouping a set of objects into categories [35]. These categories are unknown from the input data and the algorithm is expected to generate them automatically. This is in contrast to classification in which the categories are known beforehand, and the algorithms in this case group the items based on the labeled tag. Text clustering is broadly classified into two categories which include, flat clustering and hierarchical clustering.

2.2.4 Flat clustering

In flat clustering, each document is in its own clusters and no relationship exists among the clusters.

K-Means: The most popular algorithm for flat clustering is the k-mean. Its goal is to partition a set of document into k different clusters such that each document belongs to the cluster with the nearest mean [17]. That is the mean of the document is closer to that of the centroid. With this, the square Euclidean distance between each document and its centroid is minimized, thus, minimizing the within cluster square error. K-mean makes a number of assumptions about the nature of the input data [17]. These assumptions must hold if the algorithm is to produce a reliable result. These assumptions include;

1. Input data should not be spherical. k-mean algorithm would fail to generate the right clusters if the input data set is spherical. In contrast, hierarchical clustering with single linkage works well for spherical data.

2. Equal distribution of cluster size. For k-mean to work well, each cluster must have roughly the same number of data points. This assumption is not required for hierarchical clustering.

2.2.5 Hierarchical clustering

In contrast to flat clustering, hierarchical clustering group document into clusters by showing the hierarchy that exists among the clusters [31]. Hierarchical clustering is of two types; Agglomerative and divisive.

Hierarchical Agglomerative Clustering In this approach, each document is in its own clusters. Two clusters are merged according to some similarity metrics to form a new cluster. This process is repeated until the desired cluster is formed [31].

Divisive Clustering Divisive approach starts by assuming that all documents are in one big cluster. This big cluster is split to form smaller clusters. The process is repeated until the desired cluster is formed [31].

Similarity metrics

Similarity metrics is used in hierarchical clustering to determine which clusters to merge in case of agglomerative or which cluster to split in case of divisive. This metrics is usually provided in the form of a dissimilarity matrix. The dissimilarity is calculated based on some distance functions such as Euclidean, Square Euclidean, or Manhattan distance [31].

Linkage criteria

The distance between sets of document is determined by the linkage criteria which specify the distance between two documents in relation to the set of document. Different linkage criteria exist for hierarchical clustering [31].

Single linkage Two documents with the shortest distance apart are merged into a new cluster. This is also known as nearest neighbor clustering [31].

Complete linkage In complete linkage clustering, two clusters with the shortest distance are merged [31]. The distance between the two clusters is determined by the distance between their farthest elements in the cluster. This method is also known as farthest neighbor clustering. Complete linkage clustering produces better hierarchies and compact clusters when compared to single linkage as it does not have the chaining problem present in single linkage cluster. The chaining phenomenon occurs when sets of points are extended over a long distance. Single linkage clusters tend to cluster these points into one cluster without regards to the overall structure of the points. This means that a few points between two clusters will cause single linkage to merge the two clusters as one [31].

Average linkage In average linkage, distance between the clusters is defined as the average of the distances between points in one cluster to every other points in another cluster [31].

A common way of representing the output of hierarchical clustering is through a *dendogram*. A dendogram is a graph that shows the hierarchy between the documents in the corpus and the distance at which they were merged.

2.3 Feature Extraction

After text documents have been cleaned, tokenized and stemmed, it is necessary to transform the various tokens into a form the algorithms can understand. These transformations involve assigning numbers to the tokens. These numbers are referred to as the features of the text. The common technique for assigning numbers to text tokens is by considering the frequency of each term in the document. This frequency is normalized by taken the whole document corpus into consideration, a process known as the inverse document frequency (IDF). This technique, known as Term Frequency - Inverse Document Frequency (TF-IDF), is intended to reflect the importance of a word in the document corpus [32].

2.3.1 TFIDF

TF-IDF is one of the most popular methods of transforming text document into feature vectors [2]. It works by assigning weight to each word in the text based on some weighing scheme described previously. The formula for TF-IDF is given below;

$$TF\left(\text{t}\right) = \frac{\text{Number of times term t appears in a document}}{\text{total number of terms in the document}} \tag{2.1}$$

$$IDF\left(\text{t}\right) = \log_e\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right) \tag{2.2}$$

$$TFIDF\left(\mathrm{t}\right)=TF\left(\mathrm{t}\right)xIDF\left(\mathrm{t}\right) \tag{2.3}$$

To quickly illustrate the TF-IDF formula, let's take an example of a document containing 100 words and where the German word 'Hund' appears 4 times. The TF of Hund is calculated as;

$$TF\left(\mathrm{Hund}\right)=\frac{4}{100}=0.04 \tag{2.4}$$

Further assume we have 10 thousand documents and the word 'Hund' appears in 500 of these, we can calculate the IDF of Hund as;

$$IDF\left(\mathrm{Hund}\right)=\log_e(\frac{4}{100})=1.30 \tag{2.5}$$

Therefore;

$$TFIDF\left(\mathrm{Hund}\right)=0.04x1.30=0.052 \tag{2.6}$$

In TFIDF calculation, the IDF tend to give less weight to words that appear too frequent across the entire corpus. For instance, words such as 'Der', 'Die', 'Das' that are likely to appear often in a single document will have high TF number, but their IDF will almost be zero since they appear too often in the entire corpus. With the above example, let's assume 'Der' appears in 10 thousands of the documents, the IDF of 'der' will be; $IDF(der)=\log_e(10000/10000)=0$, and the $TFIDF(der)=0$

2.4 Document Similarity

To produce useful correlations within text data, it is important to compute the similarities among all the documents in the corpus. With similarity measure we could separate one document from another, group each document into clusters and identify the relationship between the clusters. The document similarity measure is calculated from the TFIDF matrix generated from the previous step.

Euclidean distance and cosine similarity are the most common measure of similarity. Euclidean distance measures the distance between two vectors, while cosine similarity measures the angle between two vectors. The formula for Euclidean distance and cosine similarity are given below.

For vectors $\vec{a}$ and $\vec{b}$:

The **Euclidean distance** is given by;

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{2.7}$$

And the **cosine similarity** is calculated as;

$$cos(\theta) = \frac{\vec{a}.\vec{b}}{||\vec{a}||||\vec{b}||} \tag{2.8}$$

For vectors with large magnitude, Euclidean distance may produce relatively acceptable result since only the magnitude is considered in the distance measurement. In the case of the vectors generated from TFIDF calculation, the application of Euclidean distance will not be suitable because the magnitude of these vectors are negligible and do not contribute much to the overall similarity structure within the documents.

The angular measurement of two vectors gives the direction or sentiment of the vectors which cosine similarity captures in its calculation. This makes cosine similarity the preferred similarity measurement for text data [15, 21]. With cosine similarity, two documents are exactly the same if the cosine of the angel between them is 1, and the documents are totally different if the angle is 0.

2.5 Dimension Reduction

The matrix generated from TFIDF is usually a multi-dimensional matrix. This multi-dimensional matrix is hard to work with and hence needs to be reduced to a two dimensional form for easy visualization on a scatter plot or for further processing as inputs to the various clustering algorithms. Given a multi-dimensional input vector, our goal with dimension reduction is to reduce the vector to 2 or 3 dimensions such that the relationship within the multi-dimensional vector is preserved. This reduced dimension can easily be visualized on a plot for instance. Various methods for dimension reduction exist which includes; Multi-Dimensional Scaling (MDS), Principal Component Analysis (PCA), and t-Distributed Stochastic Neighbor Embedding (t-SNE).

MDS projects high dimensional data into a lower dimension by preserving the Euclidean distances between the points in the high dimensional space [9].

PCA projects the high dimensional data such that the variance of the data is maximized [9]. PCA aims at preserving large pairwise distances in the data. But this large pairwise distance preservation does not work well for all data type. For example, PCA will not produce accurate representation for spherical data [41].

t-SNE measures pairwise local similarities between points in the high dimensional space [26]. t-SNE attempts to preserve the local similarity structure of the data. It aim is to model the local structure within the data by trying to bring similar points in the high dimensional space close together. Every run of t-SNE is likely to produce different results unlike PCA. This is due to the fact that the objective function t-SNE try to minimize is randomly initiated [26]. A constant known as perplexity is used by t-SNE to help fine tune its results. This perplexity is a guess on the number of close neighbors each point has. For proper functioning of the algorithm, it is recommended that the perplexity be less than the number of points. Ideal number for perplexity is within the range of 5 – 50 [26].

2.6 Topic Modeling

It is often useful and important for data analysts and researchers to be able to quickly scan large volume of documents to discover relevant topics of interests. This quick scan can reveal to users if the volume of documents under consideration require further processing. One way to help users achieve this task is to present them with a list of the most likely topics from the sets of documents. This way users know at a glance if the documents contain what they need or not. This approach of reducing sets of documents to lists of relevant topics is known as topic modeling. Topic models are algorithms used to discover the hidden theme in a set of documents. Several techniques to model topics have been proposed such as the Latent Dirichlet Allocation [4], Hierarchical LDA [16], Pachinko Allocation Model, PAM [24], and several variations of these algorithms.

2.6.1 Flat Topic Modeling

In flat topic modeling, the relationship between the modeled topics is not considered and each resulting topic is treated as independent from the rest of the topics. Popular algorithms for flat topic modeling include LDA, CTM and PAM.

LDA

Latent Dirichlet Allocation (LDA) [4] is a popular probabilistic topic model commonly applied to textual document. LDA models each document as a mixture of topics in which each topic is seen as a distribution over a set of words or vocabulary [4]. LDA algorithm works by a generative process in which a distribution over topic is chosen from a Dirichlet distribution, for each documents. Then, for each word in the document a topic is randomly chosen from the distribution. Finally, each word is chosen from the distribution over the topic. LDA assumes that each document in the sets share the same set of topics and that each topic is present in the documents in different proportion.

Practical implementation of LDA uses Gibbs sampling theory as described in [10].This implementation is given in Algorithm 2.1

Algorithmus 2.1 Gibbs sampling algorithm for LDA. Source [10]

```
Input: words $w \in$ document $d$
Output: topic assignment $z$ and counts $n_{d,k}$, $n_{d,w}$ and $n_k$
begin
randomly initialize $z$ and increment counters
for all $iteration \in N$ do
    for all $i = 0 \to N - 1$ do
        $word \leftarrow w[i]$
        $topic \leftarrow z[i]$
        $n_{d,topic} = 1; n_{work,topic} = 1; n_{topic} = 1$
        for all $k = 0 \to K - 1$ do
            $\mathrm{P}(z = k|.) = (n_{d,k} + \alpha)\frac{n_{k,w}+\beta_w}{n_k+\beta xW}$
        end for
        $topic \leftarrow samplefromp(z|.)$
        $z[i] \leftarrow topic$
        $n_{d,topic}+ = 1; n_{work,topic}+ = 1; n_{topic}+ = 1$
    end for
end for
return $z, n_{d,k}, n_{d,w}, n_k$
end
```

Topic modeling with LDA is very common and has many variances [3]. However, LDA suffers from the limitation that it only captures the relationship among words while the correlations among topics are not considered [3]. This limitation means that it would be difficult for LDA to model documents where some topics have the tendency to coexist more frequently than others. For example, a document about manufacturing is more likely to also be about economy and productivity than about sport and genetics. This limitation stem from the use of a single Dirichlet distribution to model the topic variability [4] with the assumption that one topic is not correlated with another. With this short coming, it means that LDA would not be able to discover large set of highly coherent and fine grained topics from a data corpus.

Correlated Topic Models (CTM)

Several improvement on the limitation of LDA have been proposed [3, 24], such as the Correlated Topic Model (CTM), Pachinko Alloation Model (PAM), etc. The CTM uses the logistic normal distribution [3] to model the topic proportions. This is a more flexible distribution than the Dirichlet distribution as it allows for covariance structure among the components. The CTM gives a better and realistic model of topics where one topic may be correlated with another [3].

PAM

The Pachinko Allocation Model also addresses the inability of LDA to form topic correlation due to its use of a single Dirichlet distribution. The PAM uses a directed acyclic graph (DAG) to represent nested topic correlation [24]]. Each node in the DAG graph is a Dirichlet distribution. Unlike LDA where topics are model as distribution over words, PAM extends this concept to cover topic distribution over words and also over other topics. Experimental results show that PAM produces a more coherent and consistent topic correlation than LDA [24].

2.6.2 Hierarchical Topic Modeling

Hierarchical topic modeling tries to extract the latent hierarchy that exists within the topics. In this case, each topic is assumed to be related to other topics and the algorithm is applied to discover that latent structure. Extension of the flat topic modeling algorithms produces hierarchical topic modeling algorithms such as Hierarchical Latent Dirichlet Allocation (HLDA), Hierarchical Pachinko Alloation Model (HPAM), etc.

Hierarchical topic model captures the hierarchy that exist within a set of topics from a document corpus. While topic modeling reveals the thematic structure in a document corpus, hierarchical topic modeling captures the hierarchy within this thematic structure. Hierarchical topic modeling gives a better description of the topic organization in a corpus than a flat topic model.

Hierarchical topic modeling produces a better representation of the latent topics in a corpus and helps to predict the likelihood of words in an unlabeled corpus. For instance, if we know that tennis and long jump are in a general category such as sports, we can guess the words that may be contained in a previously unknown data.

hLDA

Hierarchical LDA is an extension of LDA where topic hierarchies are generated from a set of documents by representing the topics as a tree [16]. In HLDA, hierarchical topic distribution is represented by the nested Chinese restaurant process which is an extension of the Chinese restaurant process [16]. HLDA produces a tree with certain specified depth, where a topic is associated with each level of the tree. For an L level tree, HLDA generate a set of topics by first choosing a path from the root of the tree to the leaf. Secondly, a vector of topic proportions is drawn from an L-dimensional Dirichlet. Finally, a mixing proportion of theta is used to generate the words from the topics following the path from the root to the leaf node [16].

hPAM

Hierarchical PAM is an extension of the PAM model to represent hierarchical topics. In hierarchical PAM, each node is associated with a distribution over the vocabulary [30]. This is unlike the basic PAM in which only the nodes at the lowest level are associated with a distribution of the vocabulary. In hPAM, the advantages of hLDA topic hierarchy representation is combined with the ability of PAM to mix multiple leaves of the topic hierarchy thus producing a better overall hierarchical topic representation of the document corpus.

3 Related Work

No doubt, the increasing amount of data and the demand to find efficient means of navigating and exploring these data have attracted the attention of researchers. As a result, lots of tools to support users in data exploration have been proposed [13, 22, 33, 34, 42]. These tools can be broadly categorized into overview + detail schemes and focus + context schemes [8]. The overview + detail concept presents users with the details of interested part of the data while also showing an overview of the entire dataset. Tools in this category operate in such a way that a user action in the detail section is synchronized with the overview section which gives the user a quick mental picture of where they are in the entire dataset. Common tools in this category include the slider, smoothscroll, scrollbars, etc. With the focus + context visual category, an item of interest is focused at, and this focused item is displayed within the surrounding context [8]. Tools in this category include the magic lens such as the document lens, table lens, fisheye view, DocuCompass, etc.

In [22], icicle plot was proposed as a way to visualize hierarchical data and to explore possible relationships within the data through zooming and automatic clustering. The drawback of this approach is that it is not well suited for visualizing large dataset as most of the clusters will not be visible due to space constraint.

The fisheye view proposed in [13], offer focus + context concept by showing a single detail region of the dataset at a time while also displaying the entire dataset on the screen which gives the user a glimpse of the whole structure of the data. This detail region is determined by a Degree of Interest (DOI) function which is used to specify which part of the display belongs to the detail area.

Unlike the icicle plot, the fisheye view provides navigation which makes it more suitable to explore large data collection. However, navigating through the fisheye view can be confusing for users as items which are in the detail view can suddenly disappear into the region outside the DOI making it difficult for users to keep track of the navigated data. This visual distortion has also been observed in similar focus + context techniques such as the Document lens [34] and Table Lens [33].

The smoothscroll control was recently proposed as an advanced slider like interface for navigating through one dimensional data [42]. The control is based on the overview

+ detail concept where user can see the details of interested part of the data, while also displaying a global picture of the navigated data. With the smoothscroll, the entire dataset is displayed at once but at different aggregated level going from coarse to fine grain as user navigate through the dataset.

Unlike the fisheye view, the smoothcrol control is smooth and consistent thus providing a natural means for users to navigate through large dataset. Due to its nature, the smoothscroll is particularly suitable for navigating through large collection of data that shows hierarchical nature, by grouping them into different levels of aggregation. We find the smoothscroll particularly suited for our need as it provides a calendar like interface for navigating through time dependent data.

Users are already familiar with the calendar application for time dependent data navigation, and we feel the smoothscroll is a natural fit in this regard. With the smoothscroll control, users can easily navigate the data at different time frames such as year, months, weeks and days while also providing a global view of the entire dataset. We extended the features of smoothscroll to further supports our data exploratory tasks.

Our approach combines and at the same time extends the core concepts from these different tools to provide users with a rich set of visual interfaces for navigating and exploring large and unstructured data collection.

Besides the emergence of these visual interfaces, tools that apply machine learning for text clustering, summarization and documents visualization have also emerged of recent.

In [38], a combination of NLP with calendar based visualization for the analysis of time series data was presented. In the approach, clusters were automatically detected from time series data and the corresponding patterns visualized with different colors on a calendar. However, the approach is limited to analyzing only single variable at a time. Comparing multiple variables or topics for correlation is not supported.

Text mining could also be combined with visualization to provide visual clues that complements the topic of news articles. One such area of application is the auto generation of geographical maps with relevant annotations based on locations mentioned in an article [14, 25]. In NewsView [14], NLP is used to extract key features and locations discussed in the news article, and these informations are automatically annotated on a map to provide additional context and strengthens users perception of the topic discussed in the article. These additional information extracted from the text, in NewsView are implied in the article. Users cannot search for topic of interest, or compare multiple topics for similarity or interestingness. A similar system by Lu et al. [25] depicts how events changes in the corpus by adding time series view to the geographical map. While this system is most similar to our approach, the system differs from our framework in that it focuses on the change in the distribution of frames in the dataset, and this change in frames is used to provide clue to move to a secondary dataset. Also, the approach requires that frame categories and classes are known beforehand. We do not make such assumptions in our approach, rather, our framework allows users to freely explore the data to discover this categories for themselves.

Stasko et al. [36] presents a visual analytic system, known as Jigsaw, to depicts the relationship between entities in a collection of documents, by graphically linking the entities together. Such system could assists analyst in the sense-making process by helping them to form a mental model of the relationships among the underlying entities within the document collections. These visual linking of entities or topics is similar to the method we used for topic comparison. However, the approach does not provide overview of the thematic structure of the document, which is important to help analyst kick start their exploratory tasks. Also, the visual presentation of the approach requires multiple and high-resolution monitors to effectively navigate through the different views in the application as shown in figure 3.1. Our approach presents all views concisely in a single application interface such that users can easily switch between different views without any need for extra monitor screen.

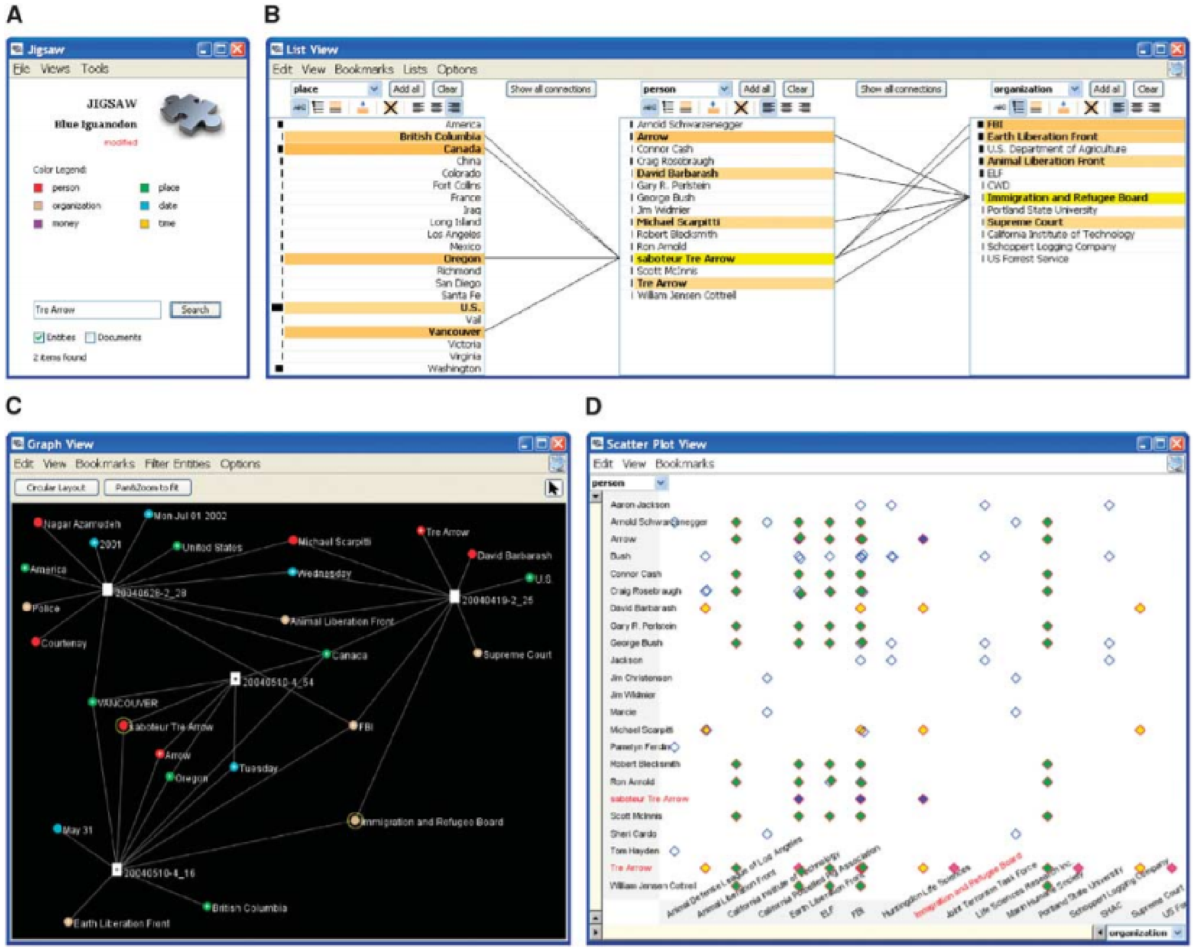


Figure 3.1: The Jigsaw system showing the various views and interfaces. Source [36]

Recent approach by Dou et al. [11] provides visualization that enables exploration of events in a dataset based on attributes such as who, what, when and where. Events characterization and visualization in the datasets are based on these attributes. Figure 3.2 shows the overview of the system. Topic modeling is used for detecting events in the corpus, while name entity recognition is used to characterize the attributes of events. However, access to the contents of each documents to confirm findings is not supported. Furthermore, the approach assumes that each event in the dataset is associated with one topic. Exploring scenarios where one event could be the result of multiple topics is not supported by the approach. This scenario is one aspect our approach differs from this tool.

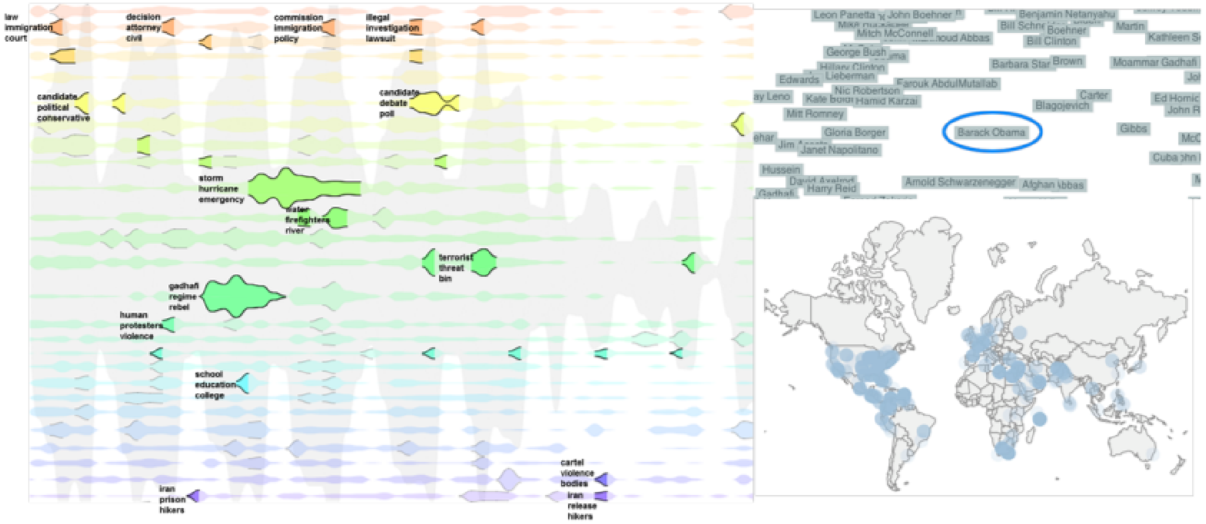


Figure 3.2: The Leadline system showing the various views supporting document exploration. Source [11]

In [37], Stoffel et al. discuss an approach to augment user's news context by querying additional contents from social media sources such as reddit using user defined search terms. NLP is used to extracts sentiment from the additional sources and the results presented visually to enrich and broadens user's perspective on the subject. This tool focus on providing users with additional information outside the content of the articles users are exploring. This differs from our approach in that we provide avenue for users to discover the hidden thematic structures and important events in a news corpus.

4 Methodology

This thesis aim at developing an interactive visualization that allows users to get an overview of important events in a news dataset by exposing to users various visual interfaces to enable them explore the data at various levels of details (year, months, weeks), with custom features that enable them to search specifically for topics. The framework behind the implementation of this thesis is designed to provide suitable interaction between the source data and the visualization.

This chapter focuses on the methodology behind the system architecture and how the framework was implemented to achieve the design goal of the thesis. The system overview showing the flow of data in the system from the raw input data, its transformation, intermediate representation, and its output as visual elements closely mirror the data state model proposed by Ed Chi [7]. This model was extensively discussed by Card et al [6] in their book where they dubbed it the *Information Visualization Reference model*. In this book, they showed that the model is applicable to wide varieties of visual applications.

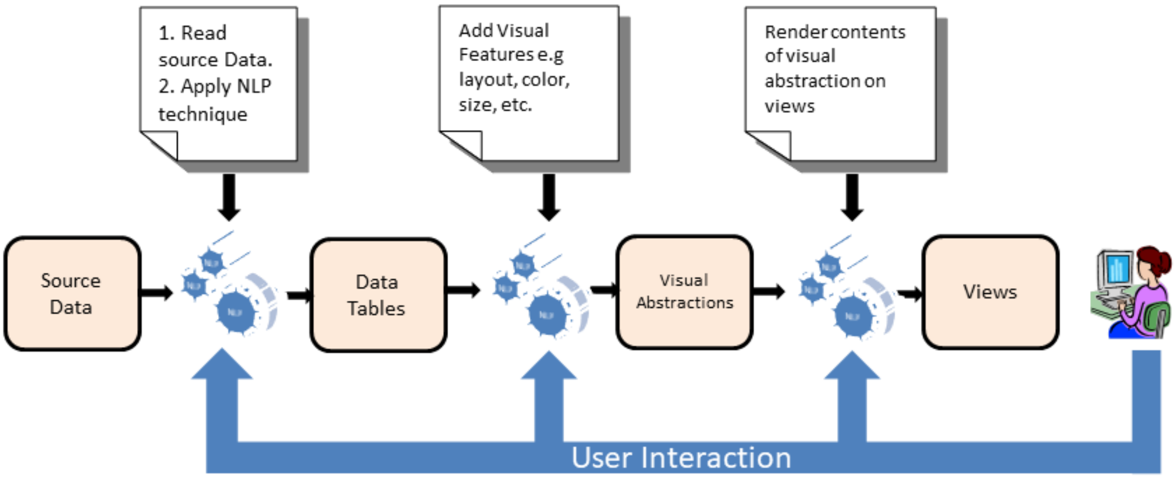


Figure 4.1: Custom rendition of the Information Visualization Reference Model depicting the major stages in the Visualization channel. The flexibility of the model allows user's interaction to potentially modify any stage of the visualization pipeline

A custom rendition of this model as applicable to our framework is shown in Figure 4.1. In this model, the raw data (Tagesschau dataset) is read from a Tab Delimited File. The data undergoes different transformation mechanism along the way to put it in a state suitable for further processing. First, the data is cleaned to get rid of empty contents. The tagesschau subtitles are then parsed and tokenized to generate the basic unit that need no further decomposition. The tokenized data is further run against a regular expression matcher to remove unimportant words that may litter the dataset and add no information to the visualization effort. Lemmatization is carried out as described in section 2.1.2. Once the data is cleaned and processed, it is transformed to internal representation such as data tables, trees, arrays etc. From the data tables and trees, various visual abstractions are created to map the data to visual representation by adding visual properties such as colors, size of display, orientation, position, etc.

4.1 Tools and Libraries

The implementation of our framework took advantages of some existing tools and libraries available in the software community. Such tools were indispensable to our project as they allow us to focus on important part of our program without the need to re-write existing free libraries from scratch. However, the tools were extended and customized to fit the specific need of our framework. In the architecture and implementation we would discuss aspects of these tools that were extended and the modifications we made to make it suitable for our application.

4.1.1 Text processing, tokenization, Lemmatization, Stemming

Processing of the raw data is crucial to ensure that quality input data is made available to the rest of the program, so that quality results could be expected from the various unsupervised learning algorithms used in the application.

For tokenization, lemmatization, stemming, part of speech tagging, we used the machine learning tool kits from Stanford University, known as Stanford CoreNLP [27].

For stopword removal, we developed our own inbuilt function to remove unwanted words that could dilute the efficiency of our algorithms as described in section 2.1.3. We used a custom and comprehensive list of German stopwords as input to our function from which we filtered out words found in this list.

Feature extraction was done by implementing the popular TFIDF weighing scheme as described in section 2.3.1. We implemented the cosine similarity described in section 2.4, from which the similarity values were calculated from the TFIDF vectors. We calculated the dissimilarity from the cosine similarity values and used this as input to the dimension reduction mechanism.

4.1.2 Dimension Reduction

Visualizing multidimensional vectors on a 2D plane is made possible through the different techniques available in dimension reduction. For our application, we used the t-distributed stochastic neighbor embedding. The java version of t-sne we found useful was implemented by Leif Jonsson[1]. The reason we chose t-sne instead of other dimension reduction techniques such as MDS, PCA, etc was highlighted in section 2.5.

4.1.3 Topic Modeling

For the unsupervised learning part of our application, we used a machine learning library known as MALLET [28]. MALLET is an open source machine learning toolkit which is available under the Common Public License. The toolkit readily implements lots of algorithms for topic modeling, document classification, clustering, etc [28]. MALLET toolkits include implementation for popular topic modeling algorithms such as LDA, PAM, hPAM, and hLDA, which we found very relevant for our application. We made use of all the topic modeling algorithms from MALLET and we finally settled for hPAM because of the reasons we described in section 2.6.2.

4.1.4 UI and Visualization

The DocuCompass, which is one of the visual elements we use and extended in our work was built on the prefuse visualization framework. We also made used of the prefuse framework extensively in some part of our application, for instance in the hierarchical display of the results of the topic modeling generated from the input corpus.

Prefuse[2] is a UI toolkit which was designed to closely mirrored the information visualization reference model described and depicted in Figure 4.1. It is released under the BSD license and it is targeted toward creating interactive and dynamic UI visualization applications. The framework is highly customizable and it is equipped with sets of packages and classes that provide the foundation for any visualization application. The

[1] https://github.com/lejon/T-SNE-Java

[2] http://prefuse.org

prefuse packages and how it relates to the information visualization reference model is depicted in Figure 4.2.

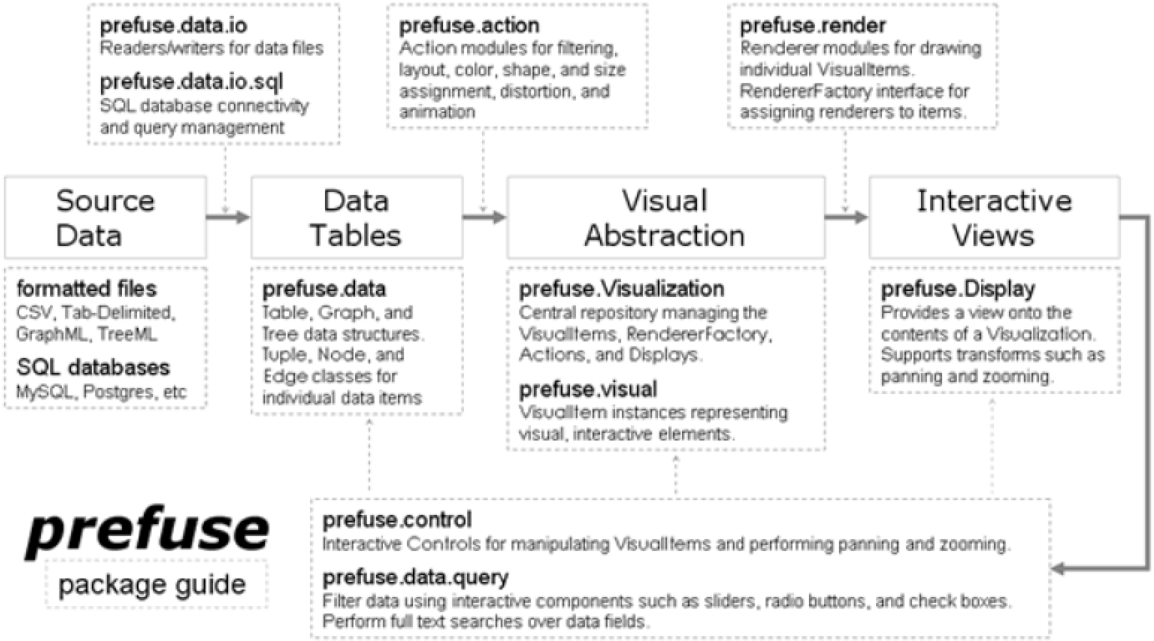

prefuse.data.io
Readers/writers for data files
prefuse.data.io.sql
SQL database connectivity and query management
prefuse.action
Action modules for filtering, layout, color, shape, and size assignment, distortion, and animation
prefuse.render
Renderer modules for drawing individual VisualItems. RendererFactory interface for assigning renderers to items.
Source Data
Data Tables
Visual Abstraction
Interactive Views
formatted files
CSV, Tab-Delimited, GraphML, TreeML
SQL databases
MySQL, Postgres, etc
prefuse.data
Table, Graph, and Tree data structures. Tuple, Node, and Edge classes for individual data items
prefuse.Visualization
Central repository managing the VisualItems, RendererFactory, Actions, and Displays.
prefuse.visual
VisualItem instances representing visual, interactive elements.
prefuse.Display
Provides a view onto the contents of a Visualization. Supports transforms such as panning and zooming.
prefuse
package guide
prefuse.control
Interactive Controls for manipulating VisualItems and performing panning and zooming.
prefuse.data.query
Filter data using interactive components such as sliders, radio buttons, and check boxes. Perform full text searches over data fields.

Figure 4.2: Packages and classes in the Prefuse tookit arranged in accordance with the Information Visualization Reference Model. This arrangement reveals how Prefuse toolkit implements each components of the Information Visualization (InfoVis) Reference Model. Source: http://prefuse.org/doc/manual/introduction/structure/

Prefuse provides support for reading data from different sources such as comma-separated-values (CSM), tab-delimited text, Structured Query Language (SQL) databases, XML, etc. It automatically transformed these input data sources into its internal data representation such as Table, Tree or Graph. The rows in the internal Table structure are stored in the Tuple class. Classes for Nodes and Edges are used to store each members of Tree and Graph. Data returned from an SQL database can be loaded directly into the internal Table, Tree or Graph structures.

Prefuse provides a powerful expression language for querying and modifying the data in its internal data structure. This language is useful for creating customized columns derived from existing data columns or for automatically adjusting the values in the data to create desired visual representation.

A visual abstraction of the data is created by adding visual attributes such as the coordinates (x,y), size, color, etc. to the data. The prefuse visualization class creates and manages these visual abstractions. These visual attributes, together with the corresponding internal data is accessible through the VisualItem class.

The characteristics and visual behaviors of an item is defined by the Action modules. These include characteristics such as color, shape, size and behaviors such as animation, layout, visual encoding and distortions of an item. The way an item is drawn on the screen is determined by the renderer class.

Interacting with the visualization is made possible through the display class which draws all the items on its view. The display class provides access to the underlying visualization and support user interaction such as zooming, panning and rotation. The display support any number of user interaction with the display or VisualItem such as keyboard or mouse actions through the Control class. The visual Item on the display can be dynamically queried through the column of the internal data structure and a predicate/query to create new columns, search for items, or filter items based user input action or need.

4.1.5 Smoothscroll

Smoothscroll is a flexible framework for navigating and aggregating hierarchical data [42]. It is written in the Java programming language. It offers customizable features which make it easily extendable and applicable to variety of data types. Its application to text data has been tested and evaluated in Verifocal Reader [20].

We implemented and extended the framework to provide timeline navigation for the need of our application. The framework supports the ability to integrate glyphs and highlights which makes it even more flexible and customizable as different visual cues such as chats, pictograms, etc can be integrated into the display to provide additional and useful information to the user.

The default orientation of the smoothscroll slider is horizontal. We had to rotate it 90 degrees to make it align with the interface of our application. We extended the smoothscroll data type to support data values for displaying glyphs, bar charts, and pictogramss. We also re-wrote part of the smoothscroll paint method to visually display these additional visual cues such as the bar charts and pictograms.

We extended the mouse event function to capture user click events and retrieve the timestamp value corresponding to the location on the slider where the user clicked. The smoothscroll was also adapted to propagate user's action on the slider to other parts of the application to ensure synchronization between the slider and the rest of the application.

4.1.6 DocuCompass

DocuCompass provides free analysis and exploration of 2D spatializations of documents corpus in which there is no definition or assumption on the information need of the user [19]. This technique is suitable as it can be used during the initial foraging stage of a set of documents to help users solidify their information need as they explore the corpus. The approach provides the focus + context concept where the item of focus is displayed within the surrounding context.

The docucompass uses the magic lens approach to provide insight into the nature of the document under consideration. The lens implementation is very flexible and has different configuration parameters which are applicable to a wide variety of user's document exploration requirements. With the lens, a subset of the data can be interactively mined and dynamically filtered to reveal documents of interest to the user.

To use the docucompass in our application, we had to make some adaptations to reflect the requirements of our application. For instance, the docucompass assumes the information need of the user is unknown. This is not entirely true in our application, as we provide an overview of the important topics discussed in the corpus to the user. This means that the first point of exploration of the corpus by the user will most likely be the topic overview interface. We use the lens to provide a summary of the important terms in the cluster of documents that contains the topic of choice the user selected from the topic overview interface. Therefore, the lens provides further exploratory analysis of the topic which helps the user to drill down on the relevant documents that discussed the topic in details. Further areas of extension of the docucompass application will be explored during the discussion of the implementation of our application.

5 Architecture

In this chapter, we discussed in details the architecture of our proposed framework for exploration of multimedia news sources. The dataset for our application is the daily Tagesschau reports which is collected over a period of three years. The recorded videos have previously been explored and the corresponding subtitles extracted by Strohm et al [12]. Our focus is to provide a suitable interactive framework to explore the news corpus in details and support users in data exploration, discovery and sense-making. Our program works directly with the subtitles already extracted from the daily news videos. The number of subtitles for the year 2015, for instance, is 3171.

5.1 System Architecture

The overall architecture of our framework is depicted in figure 5.1. At the heart of the architecture is the framework shown in the figure. This framework manages the coordination and synchronization of the various parts of the application. The framework is flexible and extensible allowing new visual interfaces to be plugged into the existing visual elements, for instance to add a new visual control or slider implementation. The main components of the architecture includes the DocuCompass, 2D spatialization, smoothscroll, topic view, picture view, content view, and data pre-processing component. The data pre-processing components transform the raw data from a file into a form that the rest of the application could work with.

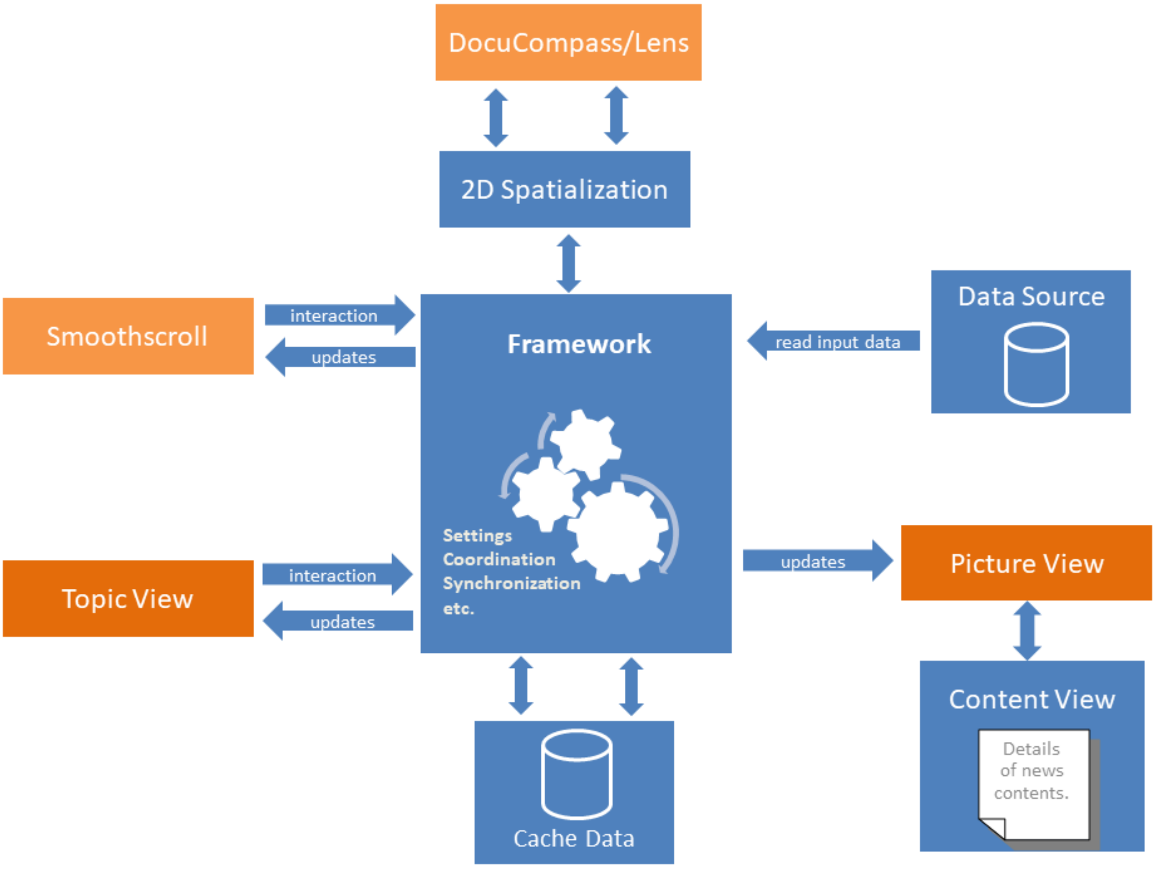


Figure 5.1: System architecture showing the different components and interfaces along with the framework that connect each system elements together. The framework is the heart of the application performing actions such as system settings, coordination, synchronization, etc.

DocuCompass: The DocuCompass is tighly integrated into the 2D spatialization view of the documents. It is separated in the architecture to emphasis the fact that the DocuCompass is not the only elements interfacing with the 2D spatialization. For instance, users actions in other views are reflected on the 2D spatialization plot by changing the size and colors of the glyphs on the plot. The DocuCompass provides the Lens components for exploring the documents on the scatterplot by showing the top terms in a cluster of document collection.

2D spatialization: the main item of focus on the interface is probably the scatterplot of the documents because it gives users an overview of the similarity plots of the documents as they are grouped into clusters. This benefit becomes more prominent as the scatterplot allows users to interactively explore the documents with other visual elements such as the Lens, smoothscroll, and the topic view.

Smoothscroll: smoothscroll provides an interactive slider for navigating the data at different time frames such year, months, weeks, and days [42]. With the smoothscroll, documents that falls within a particular time period can be selected easily and explore further. For instance, the user can select documents for a certain day or months for exploration.

Topic View: the results of the hierarchical topic modeling are displayed in the topic view. This hierarchical data is displayed using a tree UI element from the prefuse framework. This tree element from the prefuse framework was extended to suit the need of our application. We will discuss the extension we made during the implementation.

Picture View: the picture view displays the actual picture of the cover story corresponding to the documents the user is exploring. For instance, when the user click on a particular day on the slider, the picture cover for all the documents on that day are displayed on the picture view. From the picture view the users can quickly verify whether the results are in line with their expectation from their exploratory tasks. The user can also further explore the data beyond the picture view, for instance, by clicking on the picture to read the details of the news.

Content View: the content view provides an opportunity for the user to read the full content of a document under exploration. The content view is designed such that the user can view the content of multiple documents at the same time. This provides an avenue for the user to visually compare several topics and events in the dataset for relevance, similarity or both. Thus users could verify if a particular document fulfill their information need or not. The content view is perhaps the last stage in the exploration of the documents. When users get to this stage it is probably assumed that the user already has a particular information need defined by using previous visual interfaces, and that the user is now interested in seeing and reading the details of a particular or several documents to strengthen their information need or to confirm their hypothesis.

5.2 Process Flow

The process flow diagram of our application model is depicted in figure 5.2. From this model, the dataset is read from a file containing the information about the extracted video contents; chapter name, subtitle, year, month, day, filename, video start time, and video end time. These contents are read and stored in an internal tagesschaus data object. The subtitles are then preprocessed by applying tokenization, Lemmatization and finally removing stop words. The tokenization and lemmatization are done using the

StanfordCoreNLP library. Stopwords are removed using our own in built function with a comprehensive list of German stopwords. The output from the preprocessing stage is feed into the feature extraction module and the hierarchical topic modeling module. The feature extraction module uses the TFIDF weighing scheme to generate feature vectors from the tokens. These feature vectors are used as input to calculate the cosine similarity of all the documents which gives the similarity matrix. This is a multi-dimensional matrix whose dimension is in the order of the number of terms/tokens present in the longest subtitle documents. For easy visualization, the similarity matrix is reduced to a 2D projection using tSNE as the dimension reduction technique. Our choice of tSNE is elaborated in section 2.5. The 2D projection is stored as table in Prefuse internal data structure. The process that transformed the prefuse table to visualization as 2D glyphs (Scatterplot) is similar to the infoVis reference model described in Figure 4.1.

During user interaction with the scatterplot, documents stored in the prefuse table need to be retrieve quickly to create a responsive interface that dynamically process and updates user's action without any visible delay or lag that could interrupt user's engagement with the program. This could be a problem with a large document corpus. To avoid this scalability issue we used the quadtree data structure to provide optimized indexing of the VisualItems for fast and efficient document retrieval.

The hierarchical topic module generates topic hierarchies from the preprocessed input corpus. This gives an overview of the important theme or points discussed in the document corpus. The topic hierarchies are stored in xml data format which serves as input to the prefuse internal Tree data structure. Similarly, the process from the prefuse tree data structure to visualization as tree graph is similar to the infoVis reference model described in Figure 4.1. The smoothscroll takes as input the start and end time of the input data. The start time indicates the smallest timestamp in the document corpus, while the end time is the largest timestamp in the document collection. This gives the range of timelines from which the documents could be navigated on the slider.

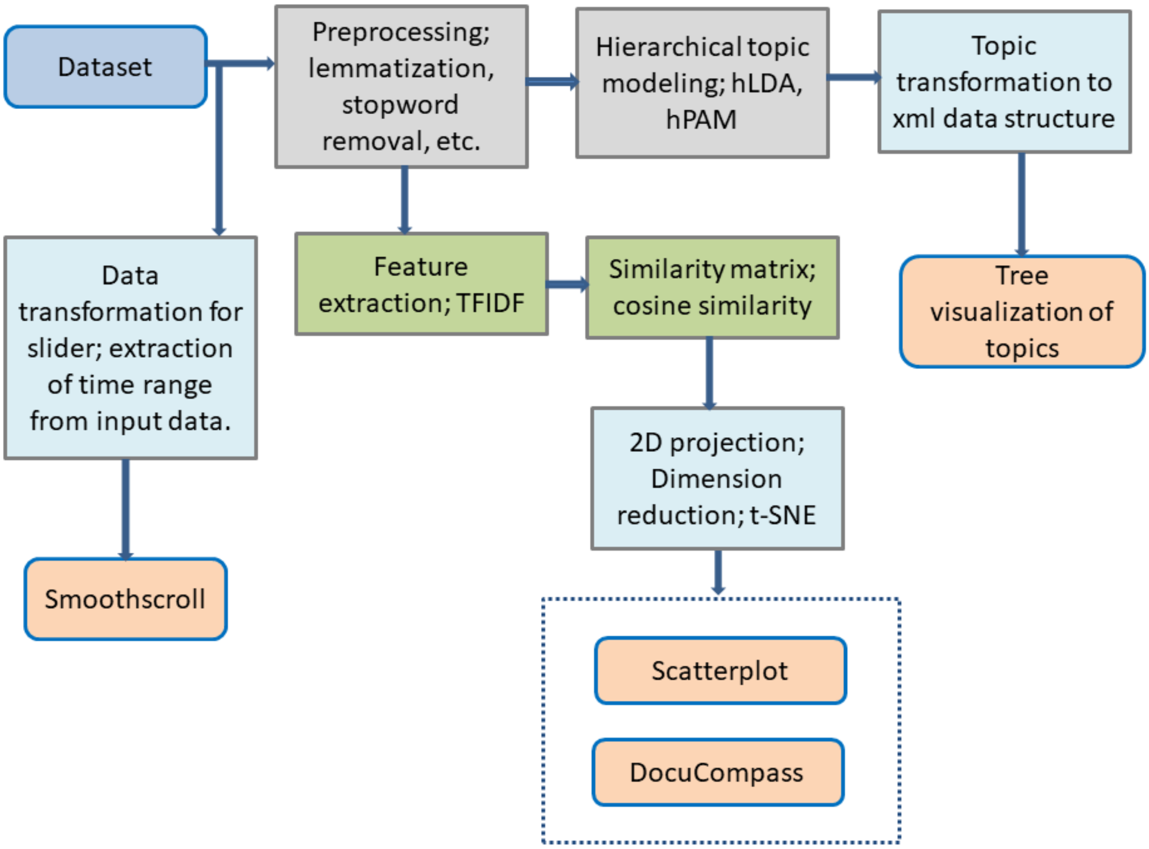

Dataset
Preprocessing; lemmatization, stopword removal, etc.
Hierarchical topic modeling; hLDA, hPAM
Topic transformation to xml data structure
Data transformation for slider; extraction of time range from input data.
Feature extraction; TFIDF
Similarity matrix; cosine similarity
Tree visualization of topics
2D projection; Dimension reduction; t-SNE
Smoothscroll
Scatterplot
DocuCompass

Figure 5.2: System process flow showing the major steps involved from data transformation to data visualizations on all primary visual interfaces.

6 Implementation

In this chapter, we discussed in details the design choices and considerations we made during the implementation of the system architecture described in the previous chapter. The major interfaces from the system architecture are listed below;

I. Smoothscroll
II. DocuCompass
III. Topic view
IV. Picture view
V. Content View

These interfaces are connected together through the framework we developed. We will discuss the details of the framework in this chapter.

The smoothscroll and DocuCompass are existing visual elements described in [42] and [19] respectively. We used these visual elements in our application and made some adjustments to make them suitable for the goal of our application. We also extended part of their implementation as we deem fit for our specific use. Details of these adjustments and additions will also be discussed in this chapter.

6.1 Smoothscroll

A basic implementation of smoothscroll for navigating timelines of pictures was presented by Wörner and Ertl in [42]. This implementation is shown in Figure 6.1.

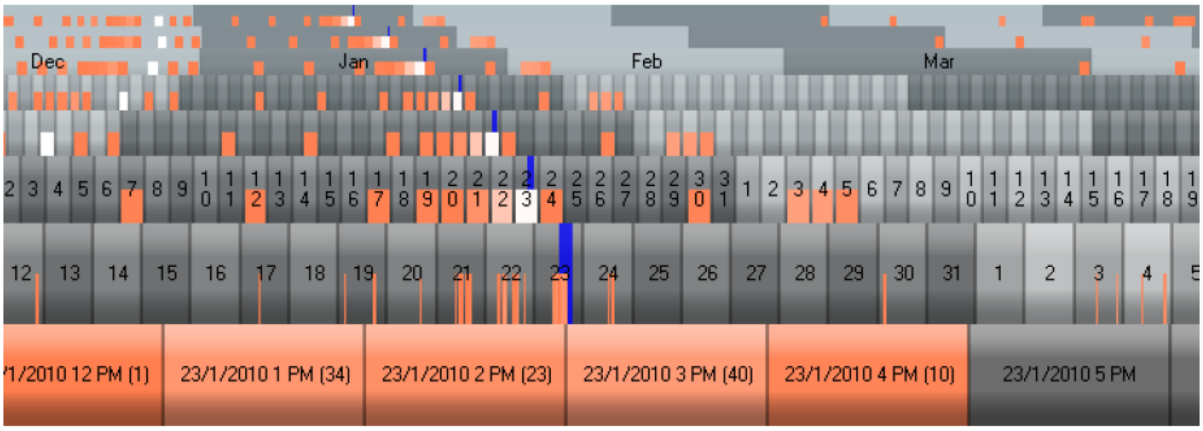


Figure 6.1: Smoothscroll Control showing a timeline of pictures as presented in [42]

Picture collection is an example of a time series data that can be navigated based on the timestamp that indicate when the picture was taken. As time is a quantity that exhibits hierarchy such that year can be divided into months, month into weeks/days, day into hours, etc., such aggregation can be easily depicted on the smoothscroll with each layer showing different levels of aggregation.

In the picture timeline example [42], the first three layers represents months. These layers are colored with alternating shades of gray with each shades representing a single month. Abbreviated month's name is added on the third layer to indicate the particular Month. The next three layers have additional shades of gray which indicate the day of the months with a day label added to the sixth layer. The last layer represents the detail layer and it shows the hour and the number of pictures taken in that hour. The bottom half of all the layers apart from the detail layer, shows the hour of the day the pictures were taken. This part is colored with different shades of orange with bright orange indicating that more pictures were taken within that hour. This orange indicator gives a quick overview of how pictures are distributed across the months and across the timeline. The blue indicator that runs through the timeline from the top to the second to the last layer indicates the users' current position in the timeline. This blue indicator runs through the month of January showing that the user is currently viewing pictures from January. Also the indicator is on 23rd which further depicts that the pictures on the detail layer are from the 23rd day of the month of January.

The tagesschau dataset, which we used as the source of our data, can be regarded as a time series data. The documents (subtitles) in this dataset have timestamps that indicate when the news event was broadcasted. This time element on the dataset can allow us to partition the data into aggregations such as year, month, week, and day. Once this partition is done, we can can navigate the dataset based on this aggregation. Just as we described for the picture timeline, the smoothscroll come in handy when data that exhibits hierarchy such as time needs to be navigated. For our application, we limited the number of layers on the smoothscroll to three.

As noted in [42], the smoothscroll requires considerable amount of screen space to display a reasonable number of layers. One of the reasons for this huge screen space requirement is because some layers are displaying redundant information. For instance, in the picture example, the number of layers could be reduced from eight to just three; month, day and hour. For instance, layers one and two repeat the information on layer three (Month). Likewise, layer four, five and six repeat the information on layer seven (Day).

The rectangular region of each layers could also be used to display additional information such as glyphs, text or highlight to the user. We decided to adopt this approach of using only the minimum number of layers that satisfy our aggregation level and to use glyphs on the layer rectangular area to convey additional information to the user. We chose this option to make room for displaying other visual elements such as scatterplot, topic view, etc., and we also do not want to distract the users with too many layers and details on the slider as other parts of the application can easily convey these details to the user more efficiently.

For instance, we thought of adding a layer that displays the word cloud of the documents from a particular month. While this is a useful feature because it gives a quick summary of the important topics discussed within that month, but having an additional layer that displays word cloud will clutter the slider interface and overwhelm the user with too many information on a single visual element. This approach will also create a repetition of information as word cloud and topic summary are available on other visual interfaces. For instance, the user can get insight into the important terms discussed within a cluster of document per month or day from the Lens implementation. User can also get a quick thematic insight into the major discourse of the corpus from the topic view interface.

Figure 6.2 shows the default interface for our smoothscroll implementation. We restricted the number of layers to three based on reasons given above.

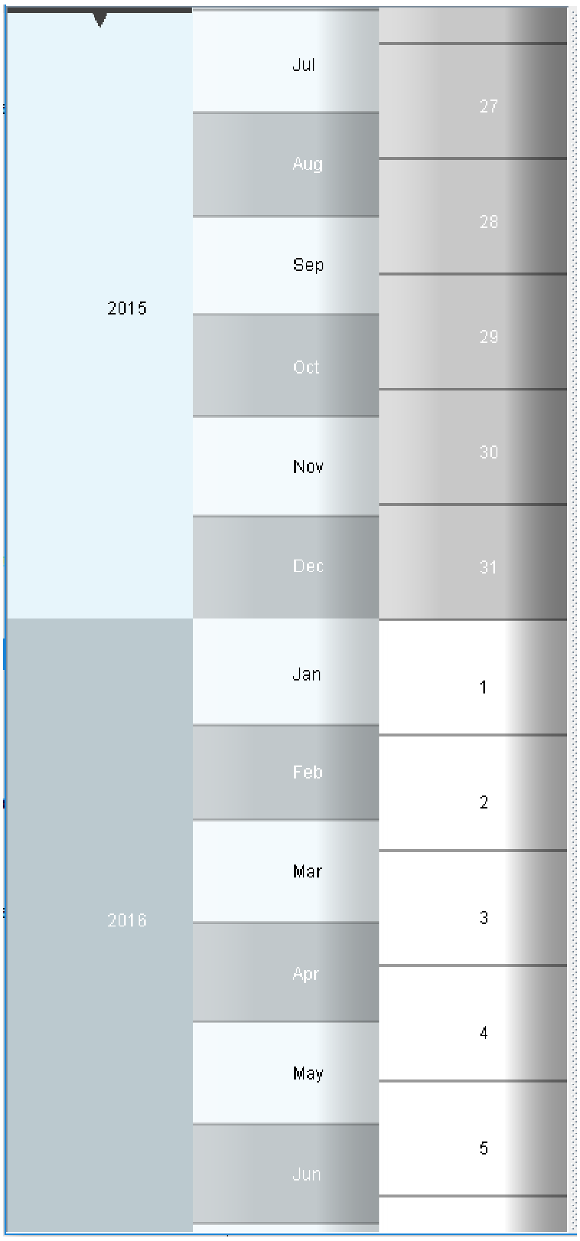


Figure 6.2: Smoothscroll implemented in vertical position with three layers ranging from detail, medium and coarse levels representing day, month and year respectively.

The detail layer shows the day, the middle layer indicates the months and the last layer shows the year. Bar chats and glyphs are used to convey additional informations to the user. When user explore the document corpus from the topic view, the number of documents that contains that topic is shown on a bar chat. This bar chat is plotted on the smoothscroll to show the distribution of the topic across the time span. The location of the bar chat on the smoothscroll indicates the exact point in time where the document occur. The chart is plotted on a daily basis. The height of the bar chat is proportional to the number of documents that contains the topic under investigation. The longer the height of the bar chats for a specific day on the slider, the more the number of documents for that day. This gives the user a quick overview of how the topic under consideration is distributed across the timeline.

When users explore the topic view, they probably also want to see the number of affected documents in a specific time frame. Important questions that could arise in such case includes, *when was topic A discussed? How many documents discussed topic A? How many of those documents are in the month of January?* for instance.

To provide insight into these questions, the slider displays a glyph that shows the number of topic within each time period. This value gives a quick summary of the number of document in each year, month or day that discusses that topic. The glyph that shows this value is drawn on the right side of each layer rectangular area to keep it separate from the bar chart which is drawn on the left.

Figure 6.3 shows the smoothscroll with bar charts indicating the highlighted topics and a glyphs showing the number of topics in each time frame. The red indicator shows the current user's position on the timeline. The figure shows that there are 291 total documents that contains both highlighted topics in year 2015. And that 24 of those documents are in the month of December, and that 3 of those documents are on the 14th of December, 2015.

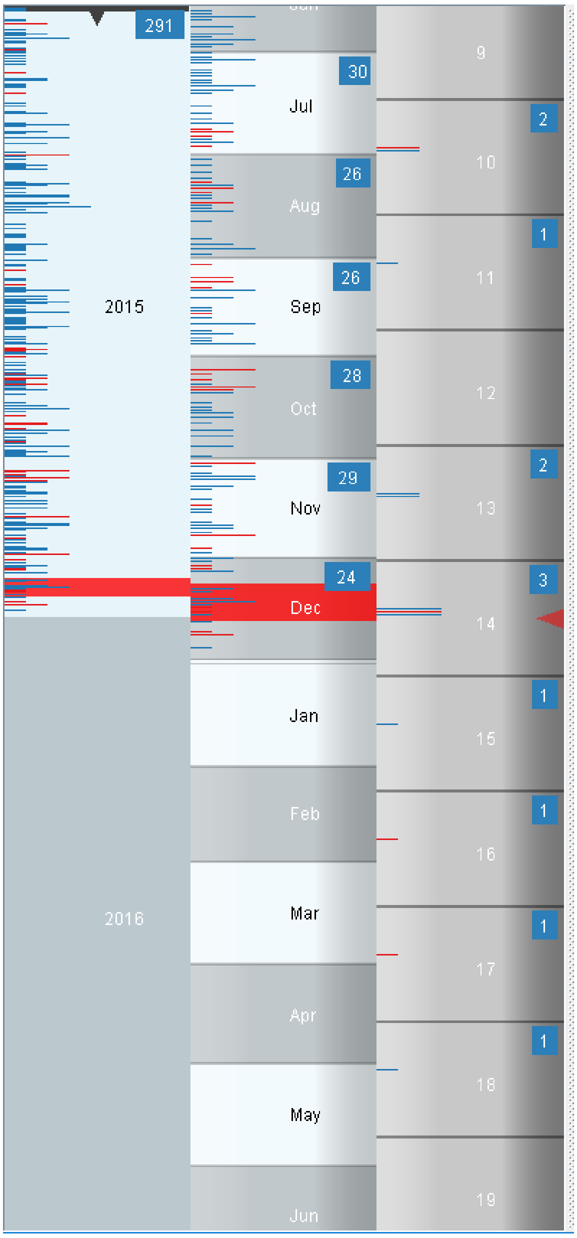


Figure 6.3: Smoothscroll showing topic distribution on the timeline. Two topics are currently selected on the topic view, and they are shown as bar charts with different colors on the slider. The value on the right side of each layer rectangular area indicates the total number of documents that contain both topics. The red indicator points to the date 14th December, 2015, which indicates the current user's position on the timeline. Documents with this timestamp will be shown on the picture view for further exploration.

Time Navigation of the corpus is not limited to the default aggregation level such as year, months and days. Other aggregation levels exist such as year, months and weeks, or year, weeks and days, etc. The slider is flexible and configurable to allow users chose which aggregation level is appropriate for the current exploratory task. Other possibilities allow the user to configure the slider with a different aggregation level which gives the user the freedom to navigate the corpus based on the time division they are interested in.

When navigating the corpus with the slider, clicking on the slider will filter the documents on the scatter plot based on the time value of the area clicked. The document filtering is shown by highlighting only those documents that have that same timestamp as the value clicked on the slider. The highlight uses the same color as the color of the topic under investigation which helps the user to keep track of the document-topic correlation and to know which documents and topics are affected by the time series navigation.

6.2 DocuCompass

DocuCompass was presented by Heimerl et al., [19] as a focus + context technique for exploring text collections in which the user information need is not clearly defined. The approach is based on the Lens metaphor in which users move a Lens over a 2D spatialization of the corpus to get a quick summary of the important terms discussed in the collection of documents under the Lens. Figure 6.4 shows the default features of the DocuCompass application as presented in [19].

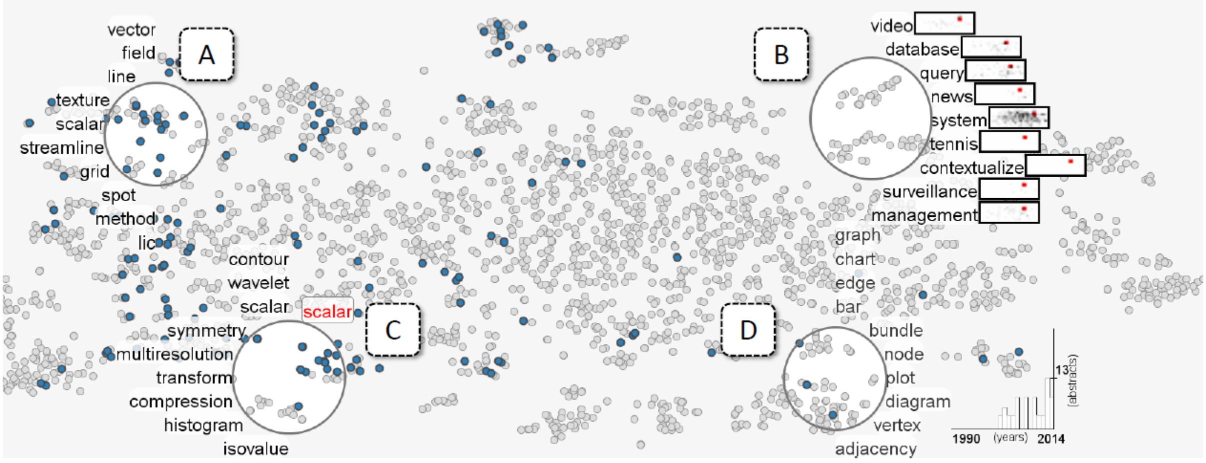


Figure 6.4: Main features of DocuCompass: (A) text labels on lens for characterizing focused documents, (B) lens with preview showing term distributions, (C) a lens with the term ‘scalar’ selected, (D) lens with bar chart showing the distribution of publication years. Source [19]

The DocuCompass technique is useful for analyzing 2D document spatializations for which user’s extraction tasks or analysis goal is not known in advance [19]. The technique makes no assumption on the user’s previous knowledge about the corpus. To discover documents of interest on the 2D spatializations, the user simply move the lens around, and the top terms from the cluster of all focused documents under the lens will be displayed around the border of the lens. This is similar to the way convex lens is used to magnify small print or small object to make them visible to the eyes for human interpretation. This makes DocuCompass a natural technique for exploring unfamiliar document specialization to uncover hidden thematic structure for further exploration.

In our application, the first point of interaction is probably the topic view where users are presented with a quick summary of the main thematic structure of the corpus. When users select a topic from the topic view, the documents that discuss that topic are highlighted on the 2D spatialization or scatter plot. At this point, the user probably wants to explore further the highlighted documents on the plot. We used the DocuCompass lens to fill this intermediate step and to provide the user with the opportunity to characterize the highlighted documents for further exploration.

In the default DocuCompass application, when users move the lens over the scatter plot, only the documents under the lens are highlighted, and all other documents outside the lens are grayed out. The exception to this is when users click on a term to see the documents that contain this term. In this case, documents matching this term are also highlighted on the scatterplot. The use of the lens in our application is a little bit different. First, moving the lens over the spatialization does not cause any document to be highlighted. Documents on the scatterplot are only highlighted based on the topic a user select under the topic view. Once, a user select a topic under the topic view, corresponding documents are highlighted on the scatterplot. When the lens is moved over the highlighted documents on the plot, a quick characterization of the documents is made available by showing the important terms from the cluster of the highlighted documents under the lens.

We reconfigured the DocuCompass lens to only provide characterization of documents which are already highlighted. Figure 6.5 shows the extension of the DocuCompass lens to make it suitable for the need of our application. Documents on the scatterplot are highlighted with different colors according to the topic selected on the topic view. The colors of the highlighted documents are the same as the colors of the respective topic selected on the topic view. This gives the user a quick overview of the spatial distribution of the selected topics in the document plot. Documents on the 2D spatialization are projected based on their similarities. The highlighted documents on the plot give the user a quick insight about the similarities of the documents that discusses the topics under investigation.

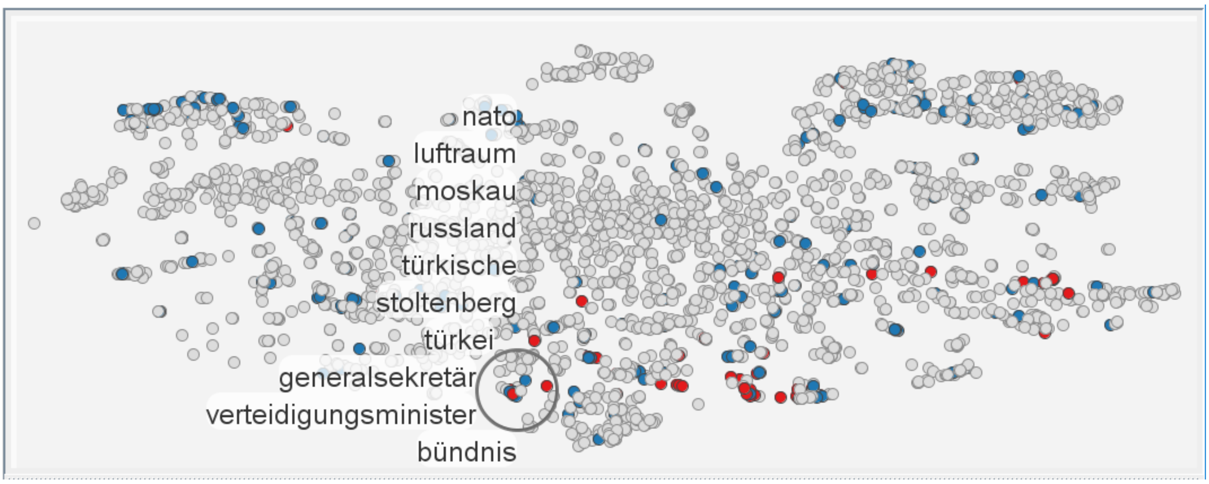


Figure 6.5: Extension of the DocuCompass lens to provide characterization of already highlighted documents. Documents are highlighted with different colors according to the topic selected in the topic view. In the plot, documents are highlighted with Red and Blue which indicates that two topics are currently selected in the topic view; one topic with Red and a second topic with Blue.

6.3 Topic View

The topic view gives a quick summary of the important topics discussed in the collection of documents. This is useful as it helps users to see at a glance the kind of topics presents in the corpus. With this, users can quickly find out if their topic of interest is available in the corpus or not. It also gives users a quick idea of what the main discourse of the corpus is about.

The first step of the topic view was done through topic modeling by applying hierarchical topic modeling technique to extract the topic hierarchies from the corpus. These topic hierarchies are displayed using a tree view UI element from the prefuse framework.

Topic hierarchies are useful as they give the topic correlation that exists in the document corpus. For instance, with topic hierarchies, topics such as football and table tennis would likely be grouped under a top level topic like sport. Similarly, finance and debt would be grouped under economic. Thus, topic hierarchies helps to extract hidden thematic structure from a seemingly unstructured document collection.

To visually display these hierarchies of topics, it is natural to use a tree view which shows graphically the hierarchical relationships that exist within the topics.

We extended the prefuse Tree View UI from the basic implementation which is shown in Figure 6.6

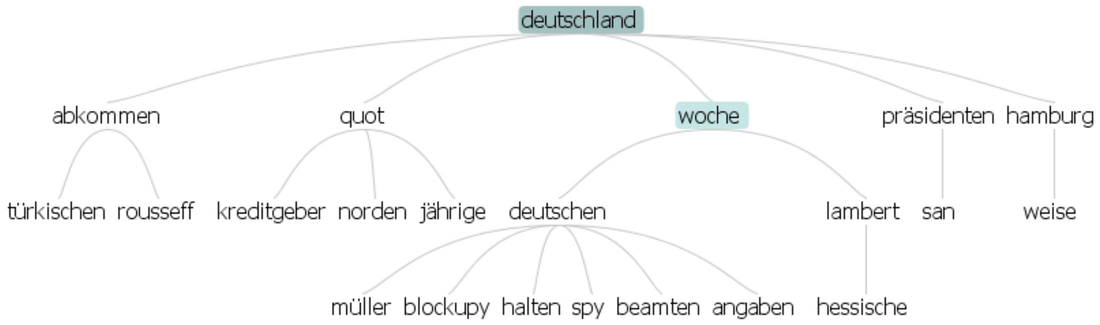


Figure 6.6: Basic Tree View UI from the prefuse framework showing the topic hierarchies that exist within the corpus. Users can interact with only one topic at a time.

Our Tree View enhancement is shown in Figure 6.7. Multiple topics on the tree can be selected.

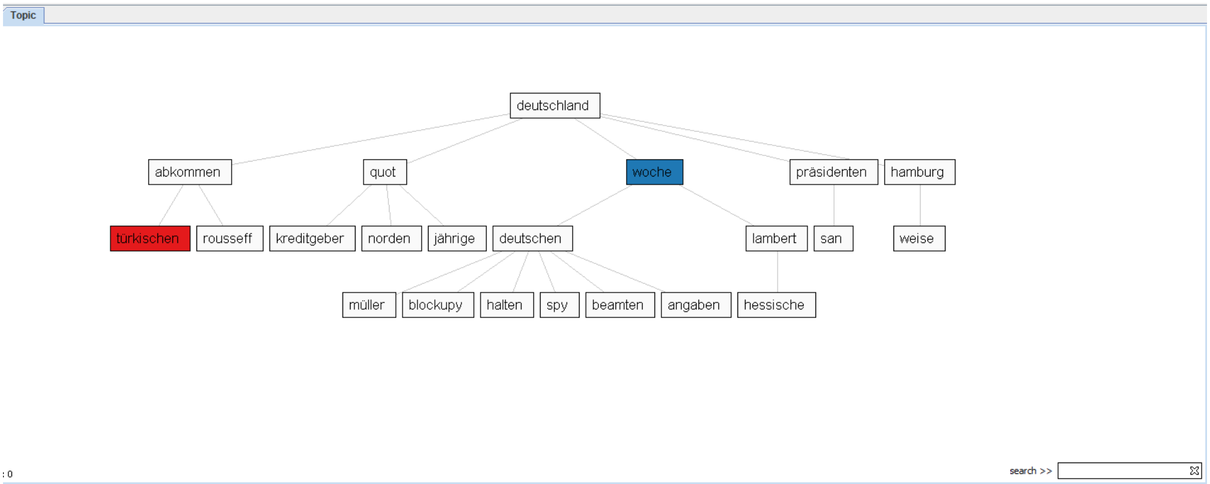


Figure 6.7: Topic View showing the topic hierarchies in the corpus. Two topics are currently selected on the view. Topic one with label, *türkischen*, is colored with red background and topic two with label, *woche*, is colored with blue background. Up to five different topics can be selected at once.

Selected topics are highlighted with different colors to help the user keep track of the topic under exploration. Up to five different topics can be selected at once for comparison and analysis. When user select a topic from the topic view, all documents that discuss the selected topic are highlighted on both the 2D spatialization and the smoothscroll. The 2D plot gives a spatial distribution of the selected topic in relation to the rest of the documents. And the slider gives a temporal distribution of the topics across the entire timeline, providing answers to questions such as when in time was the topic discussed and how often a topic was discussed over a certain time period.

For flexible exploration of the corpus, it would also be useful if users can search for specific topic of choice. It may happen that during exploration of the corpus that the user's desired topic is not listed in the topic view or perhaps hidden in the collapsed tree ui hierarchy. To overcome this problem, the topic view is equipped with a search functionality that help users to search for specific topic of choice. To search for a topic, users enter a search term in the *search box* on the lower right corner of the topic view. The topics that matched this term are highlighted on the tree ui. Also, the documents that discuss the search term are also highlighted on the 2D plot. Similarly, the historical distributions of the affected documents are also plotted as bar charts on the slider.

Figure 6.8 shows the topic view with the search functionality in action. A random term, *Kampfjets* is searched for, through the search box. The documents that match this term are highlighted on the plot and also shown as bar charts on the slider. The search term, *Kampfjets* is not in the hierarchical list of topics in the topic view.

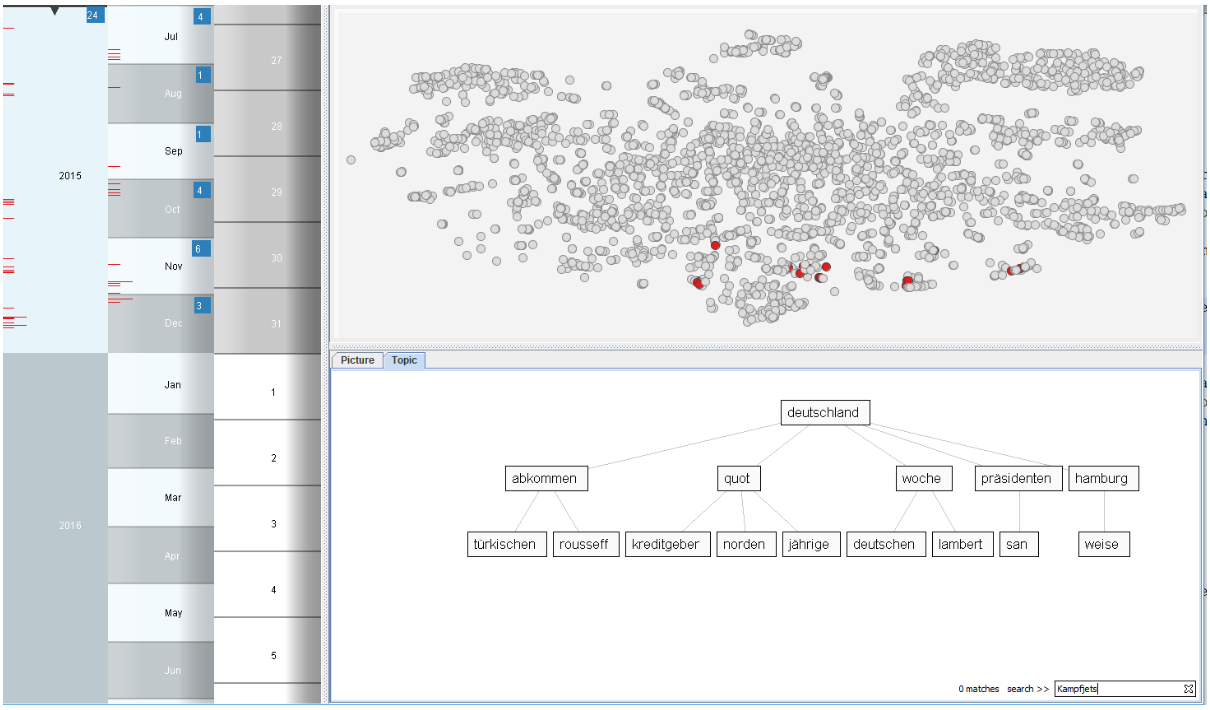


Figure 6.8: Project UI demonstrating a search for a random topic, *Kampfjets* which is not in the list of the default topics. The documents that contain this term are highlighted on the plot and also as bar charts on the slider. The slider also shows the document counts in each time period.

In Figure 6.9, a search is performed for the term, lufthansa.

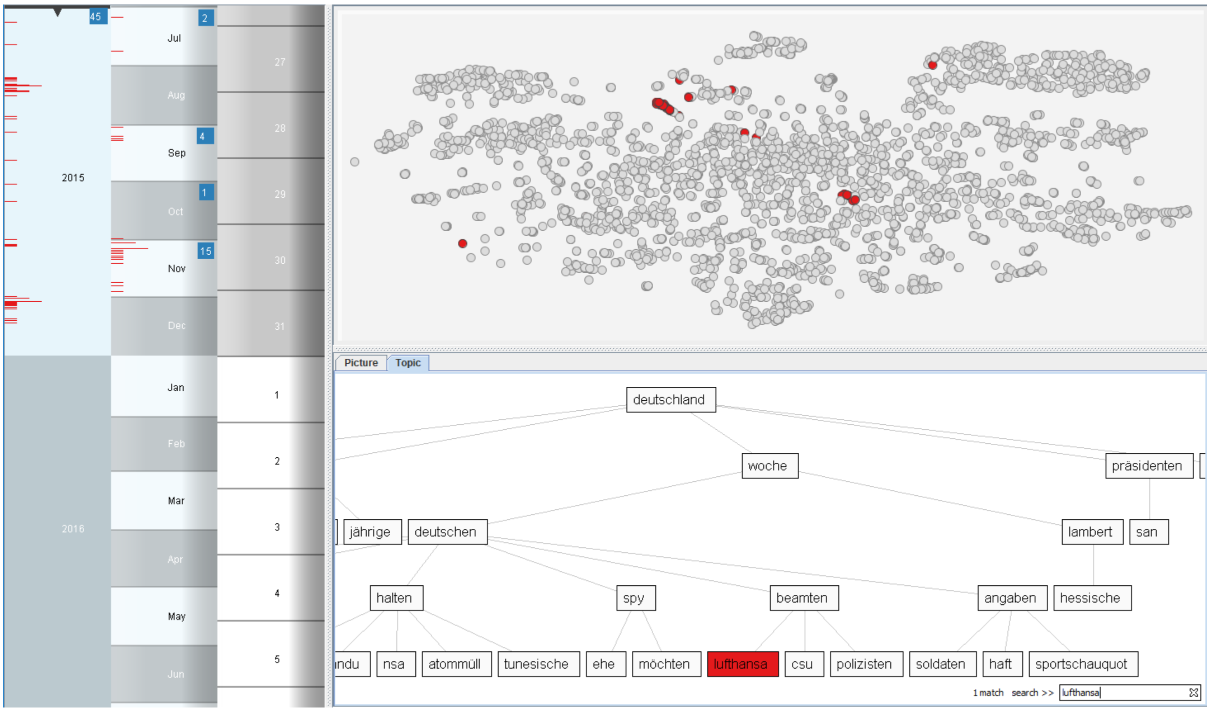


Figure 6.9: Project UI demonstrating the search for a topic, *lufthansas*. The tree view is automatically expanded to show the topic. The documents that contain this term are highlighted on the plot and also as bar charts on the slider.

The searched term is highlighted on the hierarchical list of topics to indicate that the term is part of the topics extracted during topic modeling. This term is not immediately visible on the topic view due to the collapsed nature of the tree. But when the search is performed, the tree ui is expanded to the exact location of the term to make it visible to the user. The documents that contain this term are also highlighted on the plot and also as bar charts on the slider.

6.4 Picture View

The picture view displays the cover pictures of all the document currently under exploration. When user selects a specific date on the slider, the cover pictures of all documents with this timestamps are shown on the picture view. Also during user's exploratory activities with the lens on the scatter plot, all highlighted documents under the lens are displayed on the picture view. The displayed picture contains the title of the document and appropriate graphics contents that illustrate the content of the document title. The picture view move users a step closer to confirming the hypothesis they set out at the beginning of their exploratory task.

The graphics cues provided by the picture view helps to solidify the user's knowledge of the topics under investigation. Using graphics to convey such information has been shown to be very effective [1, 29, 40].

Figure 6.10 shows the picture view. The topics, *lufthansas* and *Polizisten* are currently selected on the topic view, and filtering was done on the slider by selecting a specific date, *26th March, 2015*. The pictures shown on the picture view corresponds to the documents matching these criteria.

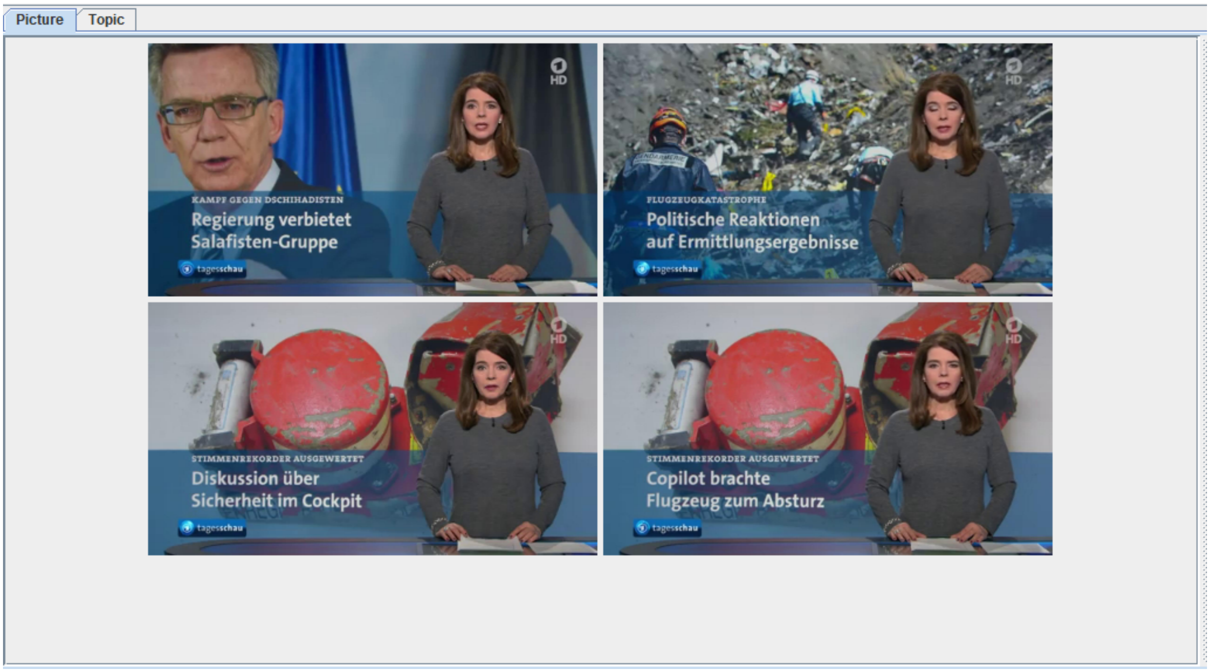

Picture
Topic
KAMPF GEGEN DSCHIHADISTEN
Regierung verbietet Salafisten-Gruppe
tagesschau
HD
FLUGZEUGKATASTROPHE
Politische Reaktionen auf Ermittlungsergebnisse
tagesschau
HD
STIMMENREKORDER AUSGEWERTET
Diskussion über Sicherheit im Cockpit
tagesschau
HD
STIMMENREKORDER AUSGEWERTET
Copilot brachte Flugzeug zum Absturz
tagesschau
HD

Figure 6.10: Picture view with a list of cover pictures of the corresponding documents that discuss the topics currently under exploration. The topics, *lufthansas* and *Polizisten* are currently selected in the topic view and further filtering was done on the slider to show only those documents from *26th of March, 2015*.

6.5 Content View

The content view allows users to read the actual news contents. This helps users to verify their analysis tasks by reading the actual content of the documents. With the content view, users can finally confirm the usefulness of their exploratory tasks, and to confirm if their information need has been satisfied. Reading the actual news contents also help users to solidify their knowledge about the topics under exploration and to confirm the hypothesis they set out at the beginning of their exploratory tasks.

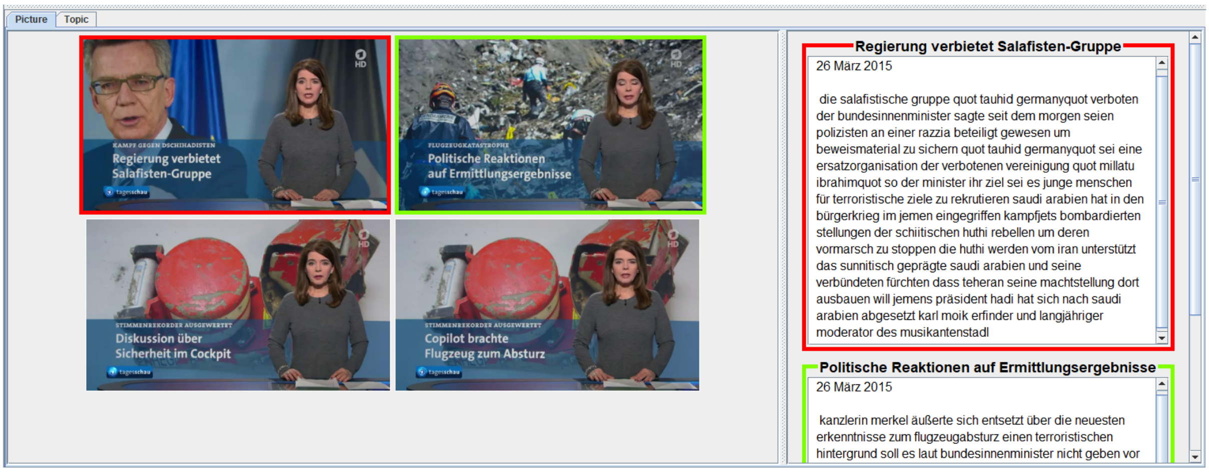
The content view allows users to read multiple contents at once. This gives users the opportunity to compare contents from different documents at the same time in order to discover the differences and similarities among the documents. Also, documents that discuss the same topic from different perspectives can be compared. This allows users to get new insights and perspectives about the topic, which helps to widen their understanding of the topic under investigation.

The content view is directly accessible from the picture view. When a picture item is clicked, the content corresponding to that cover picture is opened on the right side of the picture view. The content and its corresponding picture item are highlighted with the same color to help users keep track of which content correspond to which picture item. This highlight is done by giving the content and picture frame the same border color.

Multiple picture items can be clicked to open multiple contents which will all be highlighted with different border colors for easy context switching between the contents and the picture items.

It would also be useful for users to be able to delete content they are no more interested in or contents they have read previously or contents that does not match their analysis tasks. To achieve this, a context menu is provided with the content view, which allows users to remove undesired content from the lists of contents. Figure 6.12 shows the content view with the delete option. The context menu can be opened by right clicking on the content area of the undesired content. When a specific content is deleted, the highlight on the corresponding picture item is removed which gives the user a visual clues of the effect of their action and to help them focus on which picture items are still selected for their analysis tasks.

Figure 6.11 shows the picture view with the content view opened on the right side. Two picture items were clicked and the corresponding contents are opened on the content view. The highlight on the picture items correspond to the border colors of their corresponding contents.


Picture
Topic
KAMPF GEGEN DSCHIHADISTEN
Regierung verbietet Salafisten-Gruppe
FLUGZEUGKATASTROPHE
Politische Reaktionen auf Ermittlungsergebnisse
STIMMENREKORDER AUSGEWERTET
Diskussion über Sicherheit im Cockpit
STIMMENREKORDER AUSGEWERTET
Copilot brachte Flugzeug zum Absturz

Regierung verbietet Salafisten-Gruppe

26 März 2015

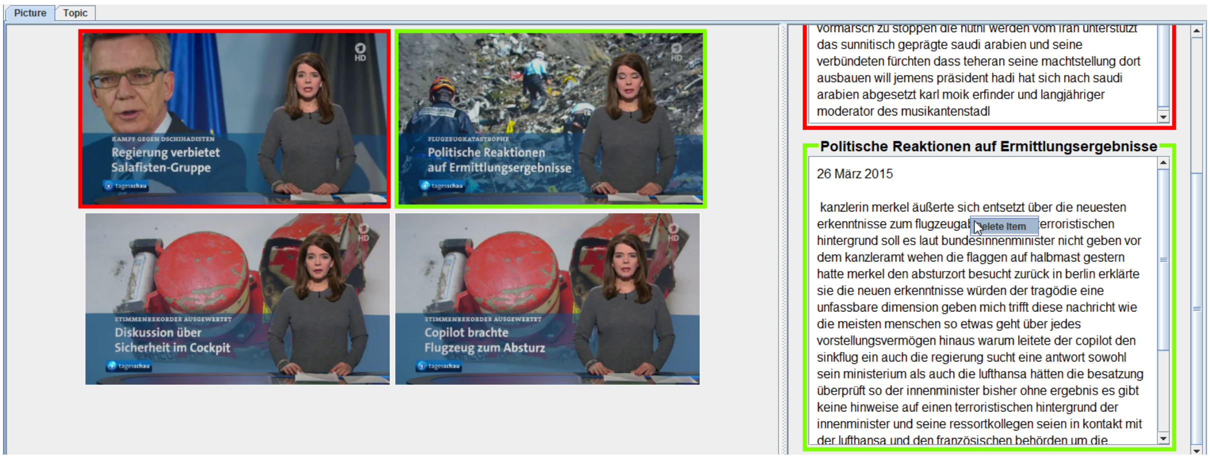
die salafistische gruppe quot tauhid germanyquot verboten der bundesinnenminister sagte seit dem morgen seien polizisten an einer razzia beteiligt gewesen um beweismaterial zu sichern quot tauhid germanyquot sei eine ersatzorganisation der verbotenen vereinigung quot millatu ibrahimquot so der minister ihr ziel sei es junge menschen für terroristische ziele zu rekrutieren saudi arabien hat in den bürgerkrieg im jemen eingegriffen kampfjets bombardierten stellungen der schiitischen huthi rebellen um deren vormarsch zu stoppen die huthi werden vom iran unterstützt das sunnitisch geprägte saudi arabien und seine verbündeten fürchten dass teheran seine machtstellung dort ausbauen will jemens präsident hadi hat sich nach saudi arabien abgesetzt karl moik erfinder und langjähriger moderator des musikantenstadl

Politische Reaktionen auf Ermittlungsergebnisse

26 März 2015

kanzlerin merkel äußerte sich entsetzt über die neuesten erkenntnisse zum flugzeugabsturz einen terroristischen hintergrund soll es laut bundesinnenminister nicht geben vor

Figure 6.11: Content view with multiple contents opened on the right side of the picture view. Two picture items are clicked on the picture view and their corresponding contents are opened on the content view and highlighted with corresponding border colors.


Picture
Topic
KAMPF GEGEN DSCHIHADISTEN
Regierung verbietet Salafisten-Gruppe
FLUGZEUGKATASTROPHE
Politische Reaktionen auf Ermittlungsergebnisse
STIMMENREKORDER AUSGEWERTET
Diskussion über Sicherheit im Cockpit
STIMMENREKORDER AUSGEWERTET
Copilot brachte Flugzeug zum Absturz
Delete Item

vormarsch zu stoppen die huthi werden vom iran unterstützt das sunnitisch geprägte saudi arabien und seine verbündeten fürchten dass teheran seine machtstellung dort ausbauen will jemens präsident hadi hat sich nach saudi arabien abgesetzt karl moik erfinder und langjähriger moderator des musikantenstadl

Politische Reaktionen auf Ermittlungsergebnisse

26 März 2015

kanzlerin merkel äußerte sich entsetzt über die neuesten erkenntnisse zum flugzeuga erroristischen hintergrund soll es laut bundesinnenminister nicht geben vor dem kanzleramt wehen die flaggen auf halbmast gestern hatte merkel den absturzort besucht zurück in berlin erklärte sie die neuen erkenntnisse würden der tragödie eine unfassbare dimension geben mich trifft diese nachricht wie die meisten menschen so etwas geht über jedes vorstellungsvermögen hinaus warum leitete der copilot den sinkflug ein auch die regierung sucht eine antwort sowohl sein ministerium als auch die lufthansa hätten die besatzung überprüft so der innenminister bisher ohne ergebnis es gibt keine hinweise auf einen terroristischen hintergrund der innenminister und seine ressortkollegen seien in kontakt mit der lufthansa und den französischen behörden um die

Figure 6.12: Content view showing the delete context menu. Analysts can delete contents they are no more interested in. This will simply remove the content from the content area. The content can be opened again by clicking the corresponding picture item.

7 Evaluation

In this chapter, we present some core use cases to demonstrate how our application can be applied to effectively explore multimedia news corpus. The questions we set out in the introduction of this work will now be answered in the course of this chapter. These questions present some of the concerns which news analysts are likely to face when they attempt to mine knowledge from news corpus.

The use cases we present here will among others attempt to answer the following design goals

1. A quick and general overview of the major topics discussed in the corpus to know if it is worth exploring further or not
2. Easily compare two or more topics for similarity
3. Easily search the corpus for topics of interest not listed in the list of hierarchical topics
4. Easy mechanism to see the details of multiple topics at once
5. Easily verify the accuracy and relevance of the data clustering on the scatter plot
6. Easy navigation through the corpus
7. Temporary (time) distribution of a topic (how often a topic was discussed over a certain time period)

7.1 Single Topic Exploration

To explore topics from the corpus, an analyst begins by loading the tageschauss news data into the system. Once the application finishes the internal processing of the data, the analyst might navigate to the topic view to get a general idea of the major topics in the dataset which reveals to the user what the content of the corpus is all about. From the topic view, the analysts can browse the list of topics; expand a topic tree to see the children of a parent topic, etc. Once the analyst has a general overview of the topics in the corpus, she might pick a particular topic of interest or select multiple topics for further exploration.

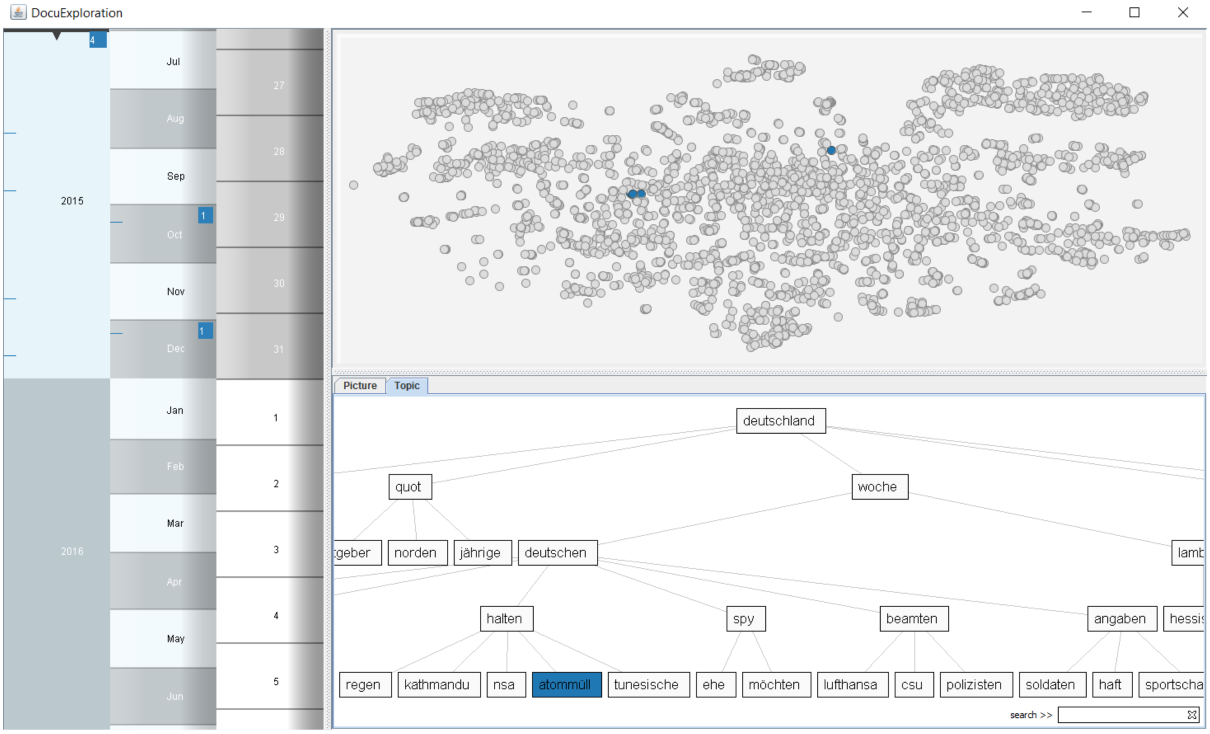


Figure 7.1: User browses through the lists of topics and the topic "**atommüll**" catches the user's attention. The user switches to the smoothscroll control and notice that the topic was discussed four times in 2015 with each in 20th of April, 19th of June, 10th of October and 8th of December.

As the user browses the list of topics in the topic view, assuming the topic "**atommüll**" catches the user's attention and she wants to know how this topic was discussed in year 2015. To find out this information, the analysts switches to the smoothscroll slider control, and from the glyphs on the slider, she noticed that this topic was discussed four times in the year 2015. Looking at the time distribution of the topic, she noticed that the topic was actually discussed once in each month of April, June, October and December. By using the slider to drill down to a specific date, the analysts noticed that the topic was discussed on 20th of April, 19th of June, 10th of October and 8th of December as shown in Figure 7.1.

Now, the analyst is curious as to what was said about this topic in each of those months that the topic was discussed. She right clicks on each of those dates on the slider to select multiple documents for exploration, insight and comparison. The selected dates on the slider are highlighted to give the analysts a visual clue on which items are already selected.

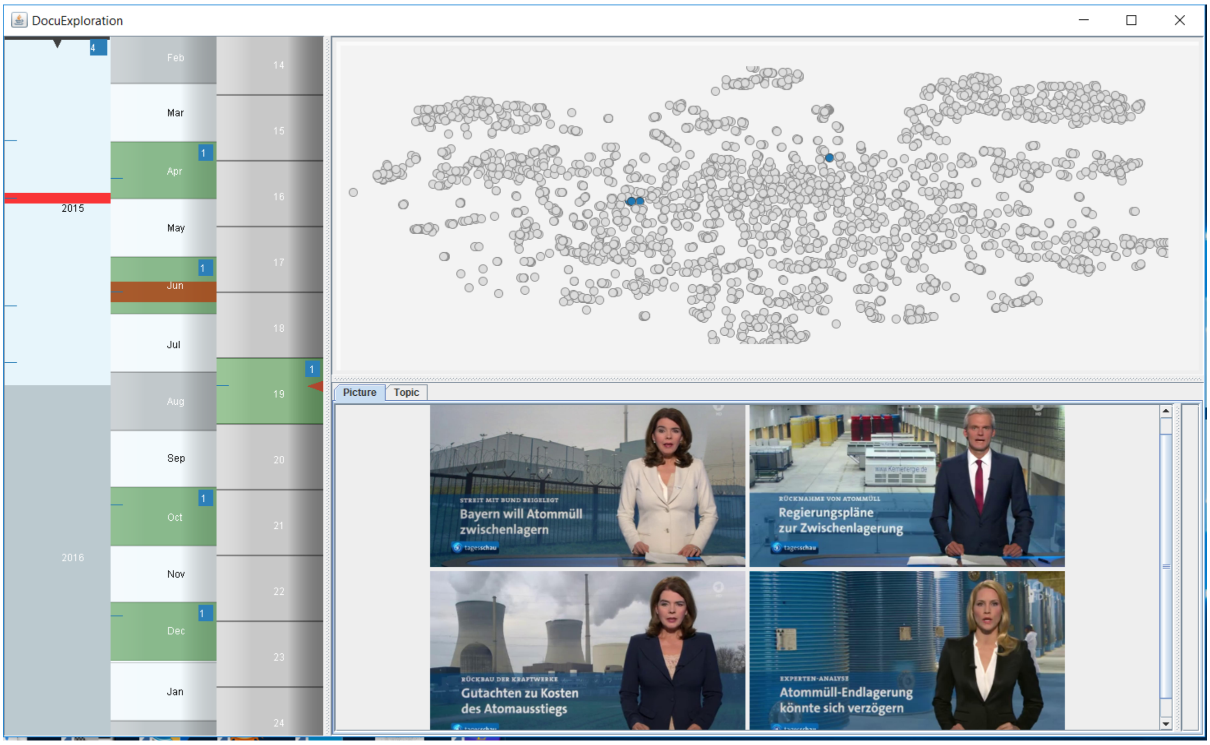

DocuExploration
4
2015
2016
Feb
Mar
1
Apr
May
1
Jun
Jul
Aug
Sep
1
Oct
Nov
1
Dec
Jan
14
15
16
17
18
1
19
20
21
22
23
24
Picture
Topic
STREIT MIT BUND BEIGELEGT
Bayern will Atommüll zwischenlagern
RÜCKNAHME VON ATOMMÜLL
Regierungspläne zur Zwischenlagerung
RÜCKBAU DER KRAFTWERKE
Gutachten zu Kosten des Atomausstiegs
EXPERTEN-ANALYSE
Atommüll-Endlagerung könnte sich verzögern

Figure 7.2: The analyst selects all four documents from the slider to compare them for similarities and differences. She notice that the months are highlighted when at least one date from each month is selected which gives her a visual clue on which month the document she is investigating is from. The analyst also notice that the red indicator is on the 19th of June; one of the dates she has already indicated interest in.

As the analyst selects the dates on the slider, the picture view is updated with the cover pictures of the document that correspond to the items selected on the slider as illustrated in figure 7.2. From the picture view, the analyst sees at a glance all four documents that discuss the topic "**atommüll**" in 2015. At this point the analyst is interested in reading the contents of each documents to solidify her knowledge about the topic and to discover the similarities and differences among the contents of those four documents that discuss the topic.

7.2 Multiple Topics Exploration

In many cases an analyst may want to compare multiple topics at once. To do this, the analyst navigates to the picture view to get an impression of the thematic structure of the corpus. In the process, two topics strike the user's attention; **türkish** and **soldaten**. The user is curious to know what was said about these two topics. The user clicks the two topics and noticed immediately that there are 196 documents that discuss the topics in 2015 as shown in Figure 7.3.

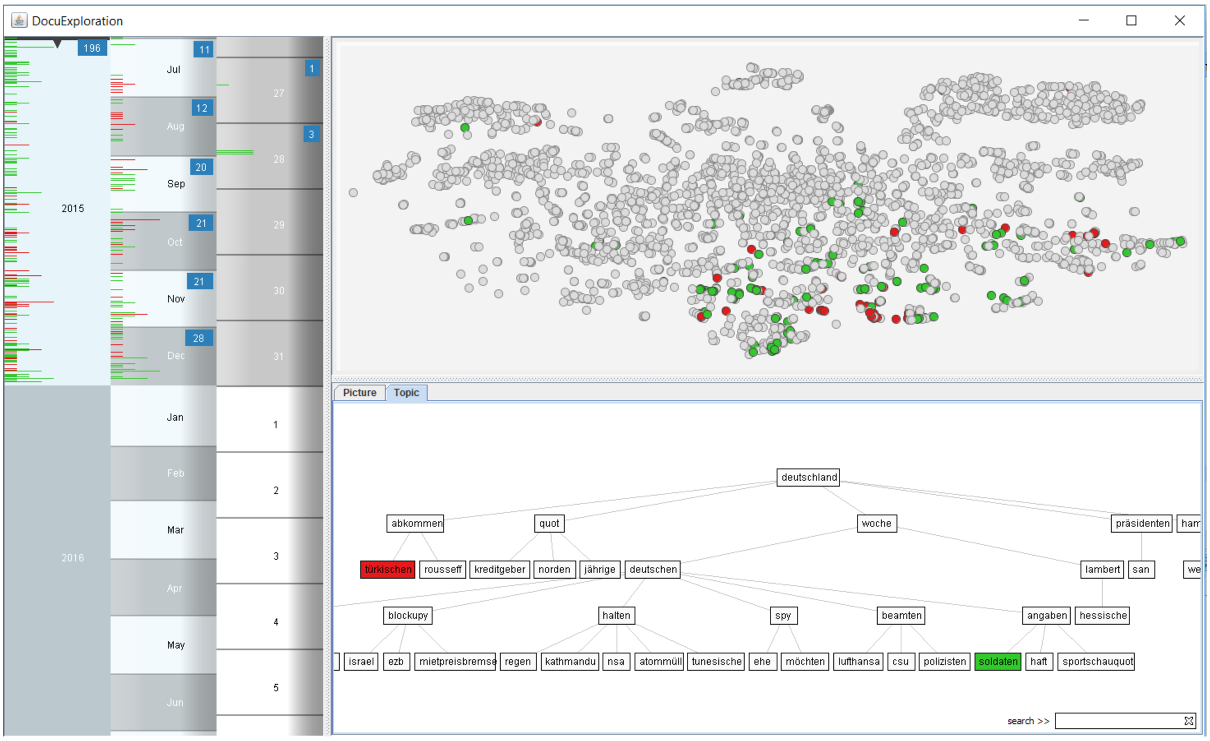


Figure 7.3: The user selects two topics for exploration and noticed the temporary distribution of the topics on the timeline with a total of 196 documents that discuss the topics. She also noticed some clusters on the 2D plot.

The user also noticed some clusters about the topics are visible on the 2D spatialization. The user is curious to find out what these clusters said about the two topics. She fired up the lens with TFIDF option. She moves the lens over each of these clusters and a word cloud is generated and shown around the lens. The word "**nato**" strikes the user's attention in one of the clusters and she wants to investigate this cluster in details. This is illustrated in Figure 7.4.

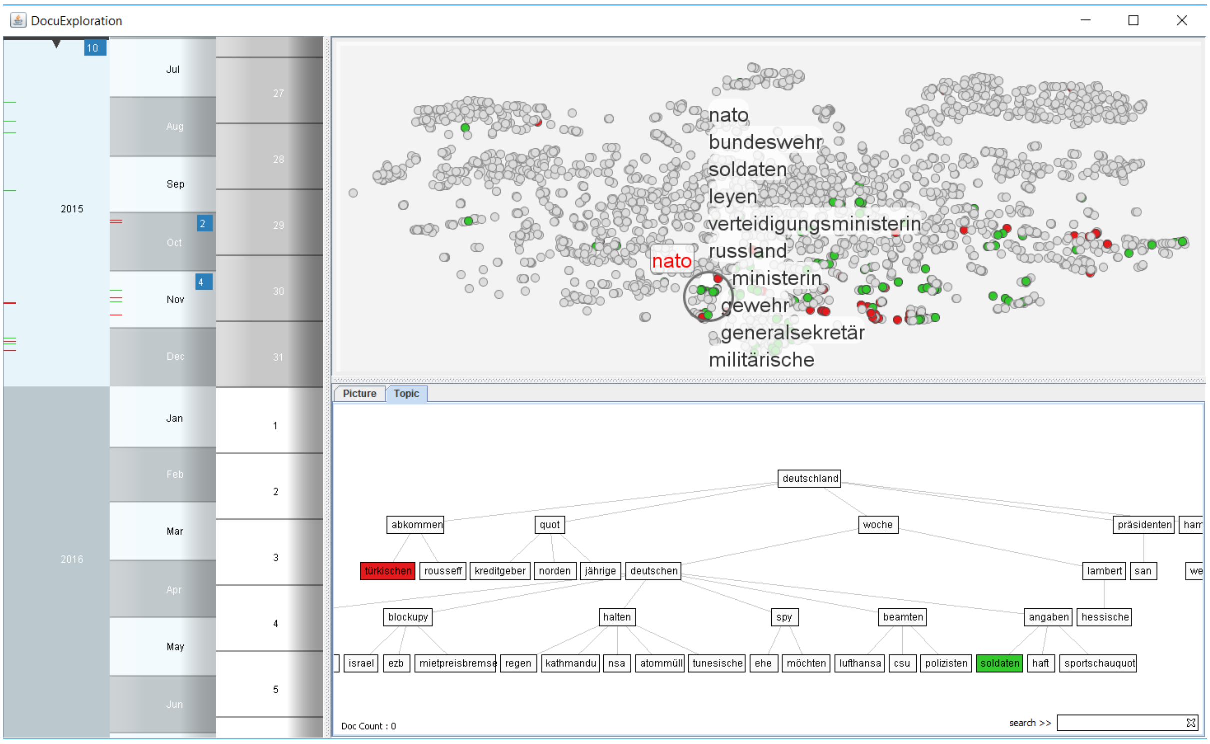


Figure 7.4: The user investigates the clusters and finds the cluster with the word "**nato**" interesting for further exploration. From the slider, she noticed that this cluster contains 10 highlighted documents in 2015.

She looks at the slider to see the temporal distributions of the documents from this cluster she finds interesting. She noticed that 10 documents discuss these two topics in 2015 in the months of January, February, March, June, July, October and November, with 3 documents in March and 2 in November, and one in each of the rest months. The analyst selects the two documents in the month of November to compare them and gain insight on what they say about the topics under exploration. As the two documents are selected on the slider, the cover pictures of the documents are displayed on the picture view which gives the user a graphics description of the topics she is investigating. She noticed immediately that one of the pictures contains members of the G20 countries and has the title "**G20-Gipfel in der Türkei**". The second picture contains graphics about soldiers patrolling the Eiffe Towel in France with the title "**Hollande will Allianz für Kampf gegen IS**" as shown in Figure 7.5.

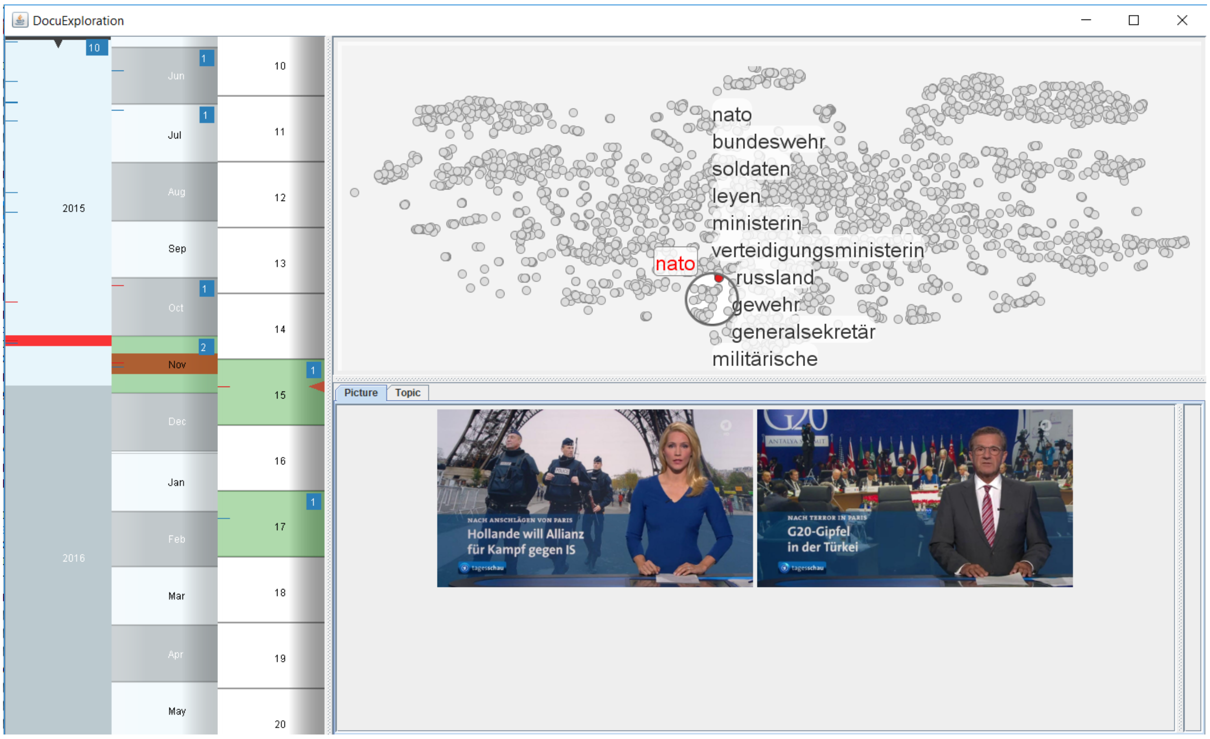
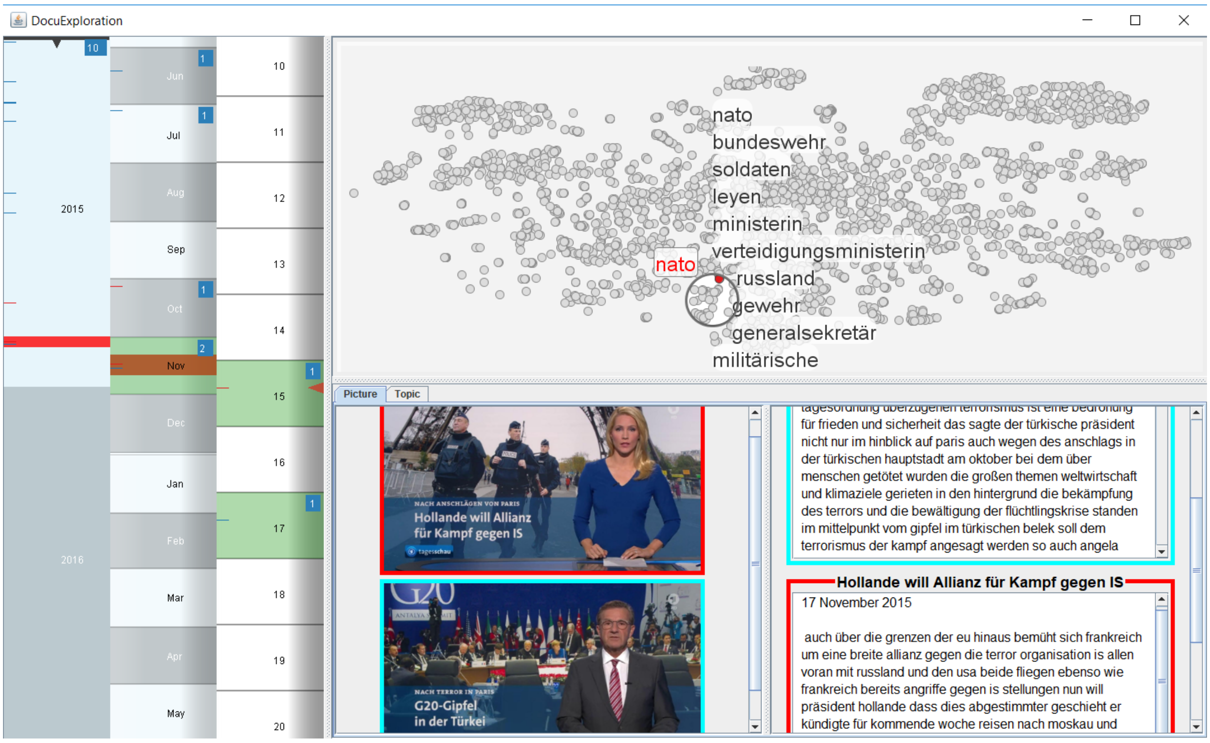


Figure 7.5: The user selects the two documents in November and makes some hypothesis about the cover pictures. The 2D plot is updated to highlight only documents from these dates.

At this point, the analyst is probably hypothesizing that the two topics she is investigating have something to do with the Middle East crisis and the actions major countries around the world are taking to solve the issue. She clicks on the two picture items to compare the contents side by side, from which she immediately confirms that her hypothesis was right. Figure 7.6 illustrates this step.



tagesordnung überzugenen terrorismus ist eine bedrohung für frieden und sicherheit das sagte der türkische präsident nicht nur im hinblick auf paris auch wegen des anschlags in der türkischen hauptstadt am oktober bei dem über menschen getötet wurden die großen themen weltwirtschaft und klimaziele gerieten in den hintergrund die bekämpfung des terrors und die bewältigung der flüchtlingskrise standen im mittelpunkt vom gipfel im türkischen belek soll dem terrorismus der kampf angesagt werden so auch angela

Hollande will Allianz für Kampf gegen IS

17 November 2015

auch über die grenzen der eu hinaus bemüht sich frankreich um eine breite allianz gegen die terror organisation is allen voran mit russland und den usa beide fliegen ebenso wie frankreich bereits angriffe gegen is stellungen nun will präsident hollande dass dies abgestimmter geschieht er kündigte für kommende woche reisen nach moskau und

Figure 7.6: The user investigate the details of the news contents from the content view and later confirm that her hypothesis was right.

From figure 7.6, the user carry out further investigation of the news contents by clicking on the picture elements to read the details of the news. The news details are opened on the content view, on the right side of the picture view. The analyst read the contents of each news article, and from which she confirms the hypothesis she made during the analysis.

8 Discussion

We presented our approach which allows users to explore multimedia news content in a new and exciting way. Our method affords users the opportunity to do topic discovery, detail analysis, insight generation, idea formulation, and general expository of the corpus. The use cases discussed confirm these features and expanded on the potential of the application. With the different visual elements, users are given the flexibility to explore the corpus the way and manner they deem fit for their analytic tasks.

Different scenarios of use cases exists with our application. We only discussed two cases to show what is possible. Other application scenarios will depend on the specific requirements of the analysts. And with the different features and options available in the application, we are confident that the application will meet a wide varieties of user's multimedia news analytical need.

8.1 Number of selected topics

Our approach limits the number of topics a user can investigate at any given time to five. The decision is to help users focus on achieving their analytic task as exploring too many topics at once could distract the user from their goal. Due to this, we make the topic selection flexible to allow users de-select topics which they are no more interested in to make room for new topics to be selected.

This limit also affects the user defined topics. Both selected topics and user defined topics are limited to five, and both topics can be de-selected to remove them from the list of focused topics. This makes the topic selection process very flexible allowing users to freely configure the topic of choice to meet their analytic tasks.

8.2 Scalability with large dataset

Though, the method appears scalable to handle corpus from multiple years, this aspect of the application has only been tested with one year. The year, 2015 we evaluated has a total of 3107 documents. Having multiple of such documents in each year will definitely impact scalability. We would like to test the limit of the number of years appropriate for a single exploration.

The application is already built to explore documents from multiple years. For instance, in the use cases we presented, the range of the slider is set to handle documents from 2015 and 2016. And also, the various internal implementation of the application are aware of documents from multiple years. With this possibility, it would be exciting to see how documents from different years are explored and analyzed by the approach.

8.3 Topic Modeling with large corpora

Another aspect of scalability that needs to be given attention is the topic modeling of the corpus. Topic modeling is a time and resource demanding operation. The basic LDA model requires a complexity of $O(Nd^2)$, where N is the number of documents and d is the number of topics [4]. LDA implementation with Gibbs sampling produces a complexity of $O(NKM)$, where N is the number of documents, K number of topics and M is the average length of the documents [5]. For a very large dataset, this complexity becomes a concern if the topic generation process must converge within acceptable time period. For instance, LDA and PAM were evaluated with NIPS dataset which contains 1647 documents, 11708 vocabulary and 114142 tokens, with 2000 iterations and it took about 20 hours for the topic extraction process to converge on a 2GB RAM, 2.4 GHz Opteron machine [24]. Our experience with hierarchical topic modeling (hLDA) with 3129 documents (from 2015), and total tokens of 37860, with 1000 iterations took 28.47 minutes on a 64-bit, 2.4GHz, intel core i5 PC with 4GB RAM.

This area of scalability needs to be carefully considered to be able to handle large datasets from multiple years within acceptable time slot, and with available system resources. To overcome this bottleneck with large corpora, one can take advantage of the various optimized algorithms for topic modeling that uses Metropolis-Hastings sampler with results that are considerably faster than the traditional implementations [23, 44]. Some of these optimized algorithms offers the possibility to run topic modeling on distributed computing, thereby, providing the avenue to scale the process to handle big corpora with huge number of topics [43].

9 Conclusion And Future Work

In this paper, we presented a framework that allows users to interactively explore topics in multimedia news corpus in a new and interesting way. We developed new visual elements and combine them with existing tools to allow users gain more insight into the corpus and to deepen their understanding of the topic under investigation. Our approach effectively provide answers to questions analysts will likely ask when they attempt to explore large collection of documents.

With our approach users can explore the provided list of topics determined from topic modeling or they can directly search the corpus for any topic of choice. We present the extracted topics in a tree visual context that shows the hierarchical structure of the topics thereby allowing easy navigation of the data.

Our approach uses the lens from DocuCompass to provide characterization of the documents on the scatter plot to offer further insights from topic segmentation and word cloud generation.

We enabled time series navigation of the corpus through the smoothscroll visual components that allow users to easily browse through the dataset in different time slots such as year, months, weeks and days, enabling users to drill down to the specific point in time when their topic of interest was discussed. Apart from time series navigation, we utilized the rectangular area of the slider to convey additional visual clues such as glyphs that shows the number of topic in each time period, and bar charts that shows the temporary distribution of topics along the time frames. Highlights are also used on the slider to provide interactive feedback to the user.

To strengthen user's perception of the topics, we provided users with an opportunity to access the main picture of each document to further assist them to confirm their hypothesis or formulate new ones. Users also have the possibility to read the actual contents of each document to verify their findings. We presented some use cases that demonstrated these features and confirm the effectiveness of the framework.

9.1 Future Work

While our approach proves very promising for wide varieties of text and multimedia news analytical tasks, a survey to evaluate the benefits of the method by real users could help to through more light on the different applicability of the approach. Mechanism that allows users to easily switch the smoothscroll slider from one level of aggregation to another will be an added advantage for analysts. Also, an interactive visual elements to manage the user defined topics involving user interactions, such as to *add topics* and *remove topics*, will boost the flexibility of the topic view.

An interesting area for future work would be the applicability of the approach to handle large corpora from multiple years. This possibility will stretch the framework and reveal the limit of corpus, and the number of years fit for a single exploration. It would be exciting to see how documents from different years are explored and analyzed by the approach. Other areas of research could be to utilized the improved algorithms for topic modeling to scale the topic modeling process to handle large corpora with huge number of topics.

Also the option to watch the actual video content of the documents under exploration would further enrich users' understanding of the topics, and also provide extra avenue to verify their findings from the original video source where the text/subtitles were extracted from. This option is just a step away as the video start and end time of each document are already available in the current framework. A suitable interaction mechanism to visually present this video content would boost user's confidence in the approach as they can watch the video (source of the text/documents) to verify their findings.

Bibliography

[1] D. E. Alvermann. “The compensatory effect of graphic organizers on descriptive text.” In: *The Journal of Educational Research* 75.1 (1981), pp. 44–48 (cit. on p. 56).

[2] J. Beel, B. Gipp, S. Langer, C. Breitinger. “paper recommender systems: a literature survey.” In: *International Journal on Digital Libraries* 17.4 (2016), pp. 305–338 (cit. on p. 17).

[3] D. M. Blei, J. D. Lafferty. “Correlated topic models.” In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*. MIT Press. 2005, pp. 147–154 (cit. on pp. 21, 22).

[4] D. M. Blei, A. Y. Ng, M. I. Jordan. “Latent dirichlet allocation.” In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022 (cit. on pp. 20, 21, 68).

[5] D. Cai, X. He, J. Han. “Training linear discriminant analysis in linear time.” In: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE. 2008, pp. 209–217 (cit. on p. 68).

[6] S. K. Card, J. D. Mackinlay, B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (cit. on p. 31).

[7] E. H.-h. Chi. “A taxonomy of visualization techniques using the data state reference model.” In: *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. IEEE. 2000, pp. 69–75 (cit. on p. 31).

[8] A. Cockburn, A. Karlson, B. B. Bederson. “A review of overview+ detail, zooming, and focus+ context interfaces.” In: *ACM Computing Surveys (CSUR)* 41.1 (2009), p. 2 (cit. on p. 25).

[9] T. F. Cox, M. A. Cox. *Multidimensional scaling*. CRC press, 2000 (cit. on p. 19).

[10] W. M. Darling. “A theoretical and practical implementation tutorial on topic modeling and gibbs sampling.” In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 2011, pp. 642–647 (cit. on p. 21).

[11] W. Dou, X. Wang, D. Skau, W. Ribarsky, M. X. Zhou. “Leadline: Interactive visual analysis of text data through event identification and exploration.” In: *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. IEEE. 2012, pp. 93–102 (cit. on p. 28).

[12] S. Florian, W. Jochen, C. Seibold. “Automatic annual review of the Tagesschau.” In: (2017) (cit. on p. 37).

[13] G. W. Furnas. *The FISHEYE view: A new look at structured files*. Tech. rep. Citeseer, 1981 (cit. on p. 25).

[14] T. Gao, J. R. Hullman, E. Adar, B. Hecht, N. Diakopoulos. “NewsViews: an automated pipeline for creating custom geovisualizations for news.” In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2014, pp. 3005–3014 (cit. on p. 26).

[15] L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, L. Pietarinen, D. Srivastava. “Using q-grams in a DBMS for approximate string processing.” In: *IEEE Data Eng. Bull.* 24.4 (2001), pp. 28–34 (cit. on p. 19).

[16] T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, D. M. Blei. “Hierarchical topic models and the nested chinese restaurant process.” In: *Advances in neural information processing systems*. 2004, pp. 17–24 (cit. on pp. 20, 23).

[17] J. A. Hartigan, M. A. Wong. “Algorithm AS 136: A k-means clustering algorithm.” In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108 (cit. on p. 15).

[18] M. A. Hearst. “Untangling text data mining.” In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics. 1999, pp. 3–10 (cit. on p. 13).

[19] F. Heimerl, M. John, Q. Han, S. Koch, T. Ertl. “DocuCompass: Effective exploration of document landscapes.” In: *Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on*. IEEE. 2016, pp. 11–20 (cit. on pp. 11, 36, 43, 49, 50).

[20] S. Koch, M. John, M. Wörner, A. Müller, T. Ertl. “VarifocalReader—in-depth visual analysis of large text documents.” In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 1723–1732 (cit. on p. 35).

[21] N. Koudas, A. Marathe, D. Srivastava. “Flexible string matching against large databases in practice.” In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment. 2004, pp. 1078–1086 (cit. on p. 19).

[22] J. B. Kruskal, J. M. Landwehr. “Icicle plots: Better displays for hierarchical clustering.” In: *The American Statistician* 37.2 (1983), pp. 162–168 (cit. on p. 25).

[23] A. Q. Li, A. Ahmed, S. Ravi, A. J. Smola. “Reducing the sampling complexity of topic models.” In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2014, pp. 891–900 (cit. on p. 68).

[24] W. Li, A. McCallum. “Pachinko allocation: DAG-structured mixture models of topic correlations.” In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 577–584 (cit. on pp. 20, 22, 68).

[25] Y. Lu, M. Steptoe, S. Burke, H. Wang, J.-Y. Tsai, H. Davulcu, D. Montgomery, S. R. Corman, R. Maciejewski. “Exploring evolving media discourse through event cueing.” In: *IEEE transactions on visualization and computer graphics* 22.1 (2016), pp. 220–229 (cit. on p. 26).

[26] L. v. d. Maaten, G. Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605 (cit. on p. 20).

[27] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, D. McClosky. “The Stanford CoreNLP natural language processing toolkit.” In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60 (cit. on p. 32).

[28] A. K. McCallum. “MALLET: A Machine Learning for Language Toolkit.” http://mallet.cs.umass.edu. 2002 (cit. on p. 33).

[29] K. B. McDermott, H. L. Roediger. “Effects of imagery on perceptual implicit memory tests.” In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 20.6 (1994), p. 1379 (cit. on p. 56).

[30] D. Mimno, W. Li, A. McCallum. “Mixtures of hierarchical topics with pachinko allocation.” In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 633–640 (cit. on p. 23).

[31] M. Oded, L. Rokach. *Data Mining and Knowledge Discovery Handbook. A survey of Clustering Algorithms*. Springer, 2010 (cit. on pp. 16, 17).

[32] J. Ramos et al. “Using tf-idf to determine word relevance in document queries.” In: *Proceedings of the first instructional conference on machine learning*. Vol. 242. 2003, pp. 133–142 (cit. on p. 17).

[33] R. Rao, S. K. Card. “The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 1994, pp. 318–322 (cit. on p. 25).

[34] G. G. Robertson, J. D. Mackinlay. “The document lens.” In: *Proceedings of the 6th annual ACM symposium on User interface software and technology*. ACM. 1993, pp. 101–108 (cit. on p. 25).

[35] F. Sebastiani. “Machine learning in automated text categorization.” In: *ACM computing surveys (CSUR)* 34.1 (2002), pp. 1–47 (cit. on p. 15).

[36] J. Stasko, C. Görg, Z. Liu. “Jigsaw: supporting investigative analysis through interactive visualization.” In: *Information visualization* 7.2 (2008), pp. 118–132 (cit. on p. 27).

[37] F. Stoffel, D. Jäckle, D. A. Keim. “Enhanced News-reading: Interactive and Visual Integration of Social Media Information.” In: *LREC 2014, Ninth International Conference on Language Resources and Evaluation Conference*. 2014, pp. 21–28 (cit. on p. 29).

[38] J. J. Van Wijk, E. R. Van Selow. “Cluster and calendar based visualization of time series data.” In: *Information Visualization, 1999.(Info Vis’ 99) Proceedings. 1999 IEEE Symposium on*. IEEE. 1999, pp. 4–9 (cit. on p. 26).

[39] J. J. Webster, C. Kit. “Tokenization as the initial phase in NLP.” In: *Proceedings of the 14th conference on Computational linguistics-Volume 4*. Association for Computational Linguistics. 1992, pp. 1106–1110 (cit. on p. 13).

[40] M. S. Weldon, H. L. Roediger, B. H. Challis. “The properties of retrieval cues constrain the picture superiority effect.” In: *Memory & Cognition* 17.1 (1989), pp. 95–105 (cit. on p. 56).

[41] S. Wold, K. Esbensen, P. Geladi. “Principal component analysis.” In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52 (cit. on p. 19).

[42] M. Wörner, T. Ertl. “Smoothscroll: A multi-scale, multi-layer slider.” In: *International Conference on Computer Vision, Imaging and Computer Graphics*. Springer. 2011, pp. 142–154 (cit. on pp. 11, 25, 35, 39, 43–45).

[43] H.-F. Yu, C.-J. Hsieh, H. Yun, S. Vishwanathan, I. S. Dhillon. “A scalable asynchronous distributed algorithm for topic modeling.” In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, pp. 1340–1350 (cit. on p. 68).

[44] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, W.-Y. Ma. “Lightlda: Big topic models on modest computer clusters.” In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2015, pp. 1351–1361 (cit. on p. 68).

All links were last followed on May 22, 2018.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature