Institute for Visualization and Interactive Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Fachstudie Nr. 162

Social Media Analysis for Disaster Management

Dang Huynh
Nils Rodrigues
Reinhold Rumberger

Course of Study:	Software Engineering
Examiner:	Prof. Dr. Ertl
Supervisor:	Dipl.-Inf. Dennis Thom M. Sc. Harald Bosch
Commenced:	September 01, 2012
Completed:	March 03, 2012
CR-Classification:	H.1.2, J.7, K.4.m

Abstract

People use social networks, with increasing frequency, to communicate and publish status information. Twitter and Facebook are prominent examples. This status information contains useful, disaster related data together with an overwhelming amount of noise. For this reason, there are dedicated tools that filter this information to aid in disaster discovery, management and relief. Their development requires a lot of time and effort, which results in high costs. Generic visual analytics tools are relatively well known, supported and continuously developed by large companies. They may be a valid alternative to specifically developed tools that require specific, non-transferable training.

The primary aim of this study is to evaluate generic visual analytics tools with respect to their ability to handle disaster management scenarios. This includes, specifically, their ability to visualise and analyse the geospatial meta information provided by social media.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Current Situation

People use social networks, with increasing frequency, to communicate with each other and publish information, for example about their current situation and environment. This includes information on any disasters that might currently be occurring in their geographical or social vicinity. Among the most prominent of these social networks are Twitter[11] and Facebook[3].

Social networks produce large amounts of data on a daily basis, which can be used to discover events in real-time [17, 20, 23, 27, 30]. This data contains a lot of information about the users' current status, as well as their geographical location and the time the information was produced. As a result, part of the data is about any disasters, natural or otherwise, the users and their social milieu might currently be affected by.

The geographical meta information provided with this data allows data analysts to determine the location of such disasters. This is based on the assumption that there is a relatively large concentration of social media messages in the geographical vicinity of such a disaster. Judging from the results of our study, this assumption seems valid. We can therefore assume, that the information provided by social media contains the information necessary to find disasters and coordinate disaster relief efforts.

One problem with this kind of data is the amount of noise contained within it. This noise presents itself as an overwhelming amount of unrelated messages, which need to be removed from the data set. Since the manual removal of millions of data records is infeasible for humans, software is needed to perform this task.

Due to this fact, there is dedicated software that uses data provided by social media to extract and visually present data about current events in real-time. These tools are usually developed for a specific purpose, like monitoring the occurrence of health disasters. They tend to be rather expensive, as the time and effort that have to be invested in their development are relatively high, while the target audience is relatively small. On the other hand, they allow analysts to discover information in an intuitive and visually pleasing manner, while providing extra features like setting up alerts or integrating coordination functionality.

As these specialised tools are hard to develop and have a small target audience, there is the idea of using generic visual analytics tools to supplant them. These tools usually already have a large user base, which means that they are under constant development and fairly well known. This has the advantage that any problems using them can quickly be researched on the internet and that full-time support is offered by their owners.

Another advantage is their generality, since they do not only offer a small amount of tools targeted at a specific application. This allows analysts to configure visualisations and analyses to provide different, and potentially more useful, results, as well as allowing them to use visualisations and analyses not provided by specialised tools.

1.2 Purpose of this Fachstudie

The purpose of this study is to analyse generic visual analytics tools regarding their ability to deal with disaster management scenarios.

This means that disaster events need to be discovered quickly and analysts need to be supported in making far-reaching decisions to direct and coordinate disaster relief efforts. To determine how useful the tools are in this context, their ability to analyse and visually represent the information provided by social networks is evaluated. We particularly focus on the analysis and visual representation of geospatial meta information. The data provided by social networks is also evaluated to determine its suitability for use in disaster management.

This study also focuses on how generic visual analytics tools handle the kind of data typically supplied by social networks. In order to achieve this, we use a relatively small sample data set of roughly 1.26 million tweets[1] in our experiments. This data set is representative for a small one-day disaster with several typical small-scale problems like tremors in Japan or shootings in northern America.

These generic visual analytics tools should be usable without much specific training. For this reason, one large focus of our evaluation is ease of use.

1.3 Structure

We will first list the kind of information a first responder in the field of disaster management would need to successfully execute their job, with a special focus on what subset of this information can be provided by social networks. Next, we discuss some specific generic visual analytics tools, explaining why and how they were chosen, how well they are suited for disaster management and which features make them stand out as either particularly suitable or unsuitable for this task. Finally, we will introduce some generic open source toolkits that can be used to create new visual analytics tools.

[1] Expected data sets are in the hundreds of millions to billions of tweets.

1.4 Related Work

There are a few studies currently available that compare and evaluate the capabilities of generic visual analytics tools.

Zhang et al. [31] compare some commercial business intelligence tools and provide suggestions on how to improve them. They conclude, amongst other things, that real-time analysis of data generated on the internet and by modern equipment will likely play a big role in the future development of commercial business intelligence tools. Predictive analysis is another underrepresented field, which is seeing a rise in demand and will probably see development.

Harger and Crossno [19] compare a number of open source toolkits, which can be used in visual analytics, according to their functionality. They conclude that only very few toolkits provide comprehensive implementations of many visualisation and analysis techniques. Most toolkits focus on either visualisation or analysis and have a narrow target platform definition. Depending on the user's needs, most toolkits have their individual advantages and disadvantages, which prevents the declaration of any clearly superior software.

To the best of our knowledge, however, there are no studies on how suitable these tools are for disaster management.

Other related studies evaluate specially developed tools. These studies focus on a specific tool, developed for a specific task like identifying news events (TwitterReporter [22]), analysing disaster events in retrospect (TwitInfo [21]) or analysing only real-time data (SensePlace2 [20], Twitcident [15]). None of the tools support both the analysis of past event information and real-time event discovery as well as tracking. The aforementioned tools also have the weakness that they only support one source of information – Twitter.

Finally, there are studies about fundamental concepts, for instance identifying methods to extract important information from extremely large data sets. These concepts allow the collection, discovery and visual representation of data from social media sources. Using this as input, we created a questionnaire of information needs necessary in disaster management scenarios. This information needs to be both provided by the data source and extracted the visualisation tool. The following studies provide an important basis for understanding the current state of development of software support for disaster management. They also allow us to compare the generic tools we inspected to the capabilities of specially developed tools:

- "Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness" [29] analyses how Twitter was used by people within natural disaster areas to communicate status information and coordinate themselves. It "identifies features of information generated during emergencies, and leads to the development of a working framework to inform the design and implementation of software systems that employ information extraction strategies"[29, p. 1087].
- "Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation" [17] explores how to identify emerging topics in Twitter by detecting spikes in the usage of keywords. This paper presents a topic detection method that uses this information to find topics in real-time.

- “Event Detection in *Twitter*” [30] uses wavelet analysis to efficiently discover events in the data volumes supplied by Twitter. It uses this method to extract real-life events from sample Twitter data and verifies that the retrieved events actually took place in the identified time frame.
- “Spatiotemporal Anomaly Detection Through Visual Analysis of Geolocated Twitter Messages” [27] extracts up-to-date information on important events such as natural disasters using spatiotemporal anomalies to improve the result of generic event detection. The anomalies in focus are geographically and temporally dense occurrences of event-related topics.
- “Spatiotemporal Social Media Analytics for Abnormal Event Detection and Examination using Seasonal-Trend Decomposition” [18] employs anomaly detection techniques to extract small-volume rare events buried among large-volume frequent events.
- “Characterizing Microblogs with Topic Models” [25] demonstrates how to use topic models to measure the similarity between Twitter feeds and then use this information for relevance assessments of other feeds for a particular user. This information is used to create a recommendation of new Twitter feeds for a user to follow, based on their relevance to the currently followed feeds.
- “A Model for Mining Public Health Topics from Twitter” [23] takes a generalised approach to detecting public health information from Twitter messages based on keyword analysis. It demonstrates that and how this information can be extracted from social media and labels approximately 5,000 Twitter messages according to their relevance to health topics.

2 Social Networks as a Source of Information

2.1 The need for information in critical situations

Information in critical situations is not only needed for people directly under the effect of the circumstances, but also for those who want or have to get involved, for friends and relatives, for others who just want to offer the victims their help, sympathy or relief. Their need for information, its urgency as well as the specific types of information vary based on which roles they take in the process. In this study we composed a list of questions that might be asked by those people and try to answer them with available visualization tools by analysing messages from social networks.

The need for information, as mentioned above, differentiate depending on the subjects in question. In order to identify them, the subjects have first to be determined. People under direct effect of the situation have the utmost need of information as their well-being or even their lives could rely on it. The rescue teams could use the information to locate and help the victims. People who manage disaster situations would need a clear look at the big picture to make their decisions more appropriately and effectively. They also could use these modern media as an information channel to get warnings, instructions and suggestions reach their destination, which undoubtedly could be very valuable should they arrive timely. Friends and relatives, who need to ensure their loved ones are safe, could also notify the authorities and responsibilities about missing people. Analysts and scientists could use the data to assess the magnitude and latitude of the events, predict the consequences or other similar events that could happen in the future or record these information into history archives. Last but not least, people who are not directly involved with these persons might want to get informed to offer their spiritual relief or assistant from afar should it be needed.

If a critical situation arises, information on its kind, intensity, geographical whereabouts and impact range, initiation and termination time as well as its development over time is the most fundamental and thus is interested by many, if not all people. The condition on the spot, the cause and reasons, the possible outcomes and consequences, number of victims, casualties, involved persons and amount of infrastructure damage, etc. are also very often of interest. Base on those information, people could decide which actions are appropriate to take to get out of the situation. Sometimes these information comes as instructions, for example in case of need for evacuation. If the victims or situation are in the position where they cannot recover without help, the aid teams and crisis control teams will be needed along with information related to them. A mutual information channel could also be established to ease the rescuing process.

2.2 Questionnaire

The section reveals the full questionnaire we composed for the study. It certainly couldn't cover every extents of information necessities in every crisis, but it is our best attempt and thus is used as a guideline for this study.

A short explanation or simple questions and answers are given to further clarify the questions.

- Is there any interesting or noteworthy event happening right now?
  Abnormal event detection.
  Two forest fires in north America and Africa.
- What kind of event is of interest?
  The event classification, e.g. earthquake, fire, flood, storm, etc.
- Where does the event take place?
  The location of the event.
- How widespread is the event?
  The actual geographical range of the event.
- When does the event start?
  Initiation time of the event
- Did the event finish or when will it expectedly end?
  Termination time of the event
- Which phrases has the even go through?
  Current and past statuses of the event. Warning, impact, recovery? [29]
- What is the intensity of the event?
  The grade of an event base on a specific scale for its type.
  Earthquake richter, hurricane category, flood water level, tornado Fujita-scale, etc. [28]
- Is an event foreseeable?
- How many people are directly involved?
  Estimated and accurate number of affected people, deaths, casualties, missing, etc.
- What are adequate reactions? Evacuation, seek for shelter, stay alert, warn other people, store food and supplies, etc.
- How are the conditions where the event takes place and surroundings?
  Weather, people, animal, building, etc...
  Is there any closed roads, structural unsafe buildings to avoid?
  Should power plans be shut down?

- Is there any protocol or instruction to follow in order to get out of the situation safely or with minimal damage?
  Update on evacuation route, information on roads with heavy traffic, do not use elevators, wrap wet blanket to protect body in case of fire, wear gas mask or hazmat suit, escape to higher ground, etc.
- Would there be consequences of the event?
  A biohazard in an industrial factory could lead to chemical poisoning, radioactive environment, fire or explosion, etc.
  Earthquake could activate volcanoes, which could release volcanic ash cloud, which forces closure of airports.
- Which supporting organizations are already on site?
  Police, fire-fighter, military, first aider, voluntary
- Which state of mind are the people on site and surroundings?
  Calm, panic, grieving, waiting for help, etc.
- How is the health care conditions for people on site?
  Is food and clean water, medication, shelter, etc. available?
- Which communication channels are still functioning? Does mobile phones, landlines, internet, radio, TV, warning systems (sound, light) work?
  How could information, warning and instruction be delivered to the victims?
  How can rescue teams communicate with each other?
- Is information on location of victims and rescue teams available?
- Where to seek for official or trustful information?

2.3 Concrete tasks for the study

Although there is a wide range of real information need in these special situations, the big question is that, if useful relevant information could be extracted from social networks and more importantly, how visualisation software could assist in this process; what a tool can and cannot do and which information can be discovered. At the moment, computer software cannot completely replace the human mind in many complex tasks, especially when semantic analysis is involved in such tasks. Therefore, within the limited scope of this study, we decided on a number of realistic, tangible, short-listed tasks that we might be able to solve with the assistance of visualisation tools. These tasks are the base on which we construct and execute our tests.

Data source Which data sources does the tool support?

- Does it support database connection, plain text file, live data stream or internet protocols and formats e.g. REST, RSS, JSON, etc.?

Data format Can the software recognise the proper data type, format and interpret the data's meaning correctly?

- How are date and time formats handled? Are numbers and geographic coordinates correctly interpreted?

Data redundancy Can the tool detect duplicates and eliminate redundancies?

- Removal of identical data records, re-posts of messages, etc.

Filtering Can the tool filter data according to its content?

- Does the tool support inclusive and exclusive filtering?
- Can it filter text data using single or multiple keywords?
- Can it filter value ranges (numbers, coordinates, time period) and value lists?

Content analysis

- Can the tool identify relevant content of a specific topic through similar or misspelled keywords (earthquake, earth quakes, earth-quake, quake, erathquak), semantic similarity (storm, hurricane, wind, rain, lightning) or related topics (earthquake, volcano eruption, ash cloud, airport closure, economic downtime)?

Anomaly detection

- Can the tool detect unusual events by the amount of data records in a time period or by the abnormal geographical density of messages?
- Can the tool detect atypical keywords?
- Can the tool expose a small yet significant event buried in the information stream of other larger events?

Data grouping Can data with similar properties be grouped?

- Group messages which contain a specific keyword and were written in the same country.
- Group records within a one mile radius.
- Group records that were written within a period of 15 minutes.

Quantity visualisation

- Can the quantity and density of records be visualised on a map?
- Can the quantity of data be visualised in a period of time or in
- comparison of different time periods?
- Can the quantity of data be visualised in a comparison between geographic regions, e.g. on a heat map?

Time detection

- Can the first and last records of an event be detected? What about disruptions and duration of the event?

General visualisation feature

- Can a short list of properties or the whole original data record be retrieved quickly through its graphical representation in the visualisation?
- For example per mouse over, mouse click or context menu.

3 Analysis Tools in Disaster Management

The vast amount of data freely available in social media is too much for a human to process in a time frame that allows adequate responses and measures to be taken in a crisis situation. Therefore, software tools are needed to aid with information retrieval about and detection of disasters.

3.1 Specially Developed Tools

SensePlace2, TwitInfo, TwitterReporter and Twitcident are software that was implemented to fill a void in this market segment. They connect directly to Twitter or use, even temporary, mirror database systems as a source of information to find and show the most relevant tweets. The representation on a map allows the user to mentally associate the tweets with a location and detect abnormal accumulations. Temporal, spatial and content filtering help users to locate possible disasters and any other information in general. In another step, details on demand provide data that has been gathered by people on site and shared via the Twitter network.

3.2 Existing Visual Analytics Tools

Unfortunately, the specialised tools are tailored to specific networks or until now don't provide sufficient customisation options. On top of that, they are too restrictive or make assumptions like SensePlace2's that the user is already searching for a specific crisis or event. This makes it more difficult to find a crisis that was previously unknown. General visual analytics (GVA) tools have been around for quite some time now and are highly adaptable. If they could be used for the same tasks as the specialised tools, users would have the best of both worlds.

3.2.1 Tool Selection

To asses how suitable GVA tools are in the context of disaster management they have to be tested with some sample data. We chose a pre-recorded set of 1.26 million Twitter messages, sent during the entire day of 23rd August, 2011. This coincided with the 2011 Virginia Earthquake[12]. This earthquake occurred at 17:51 UTC and had a magnitude of 5.8. It could be felt all over the American east coast an eastern parts of Canada. It caused no reported deaths and only minor injuries, but caused widespread minor damage to buildings.

The data was saved in a tab-separated format inside a text file in the UTF-8 encoding and without quoting. The following columns were included but had no header with their names:

Message ID An 18 digit integer assigned by Twitter to identify the message.

User ID An integer assigned by Twitter to identify the message author. In our sample it had between 2 and 9 digits.

Timestamp The date and time at which the message was written, in UTC. This is formatted as a concatenation of numbers, starting with the year and ending with minutes.
Example: 23rd August, 2011 20:54 ⇒ 201108232054.

Latitude A floating point number representing the Latitude of the author's position at the time of writing.

Longitude A floating point number representing the Longitude of the author's position at the time of writing.

Geotag A human readable description of the author's position at the time of writing. This information can be arbitrarily exact or omitted completely.
Examples: "South Carolina, US", "Brasil"

Text The message content as text.

The Latitude and Longitude columns were sometimes set to "0" when there were messages with missing columns.

Since this is a study without financial resources, we are not able to check tools that require payment of any kind. This leaves programs that are completely free or provide a free version that might have reduced functionality.

Based on the work of Zhang et al. [31] four popular tools with an established user base have been selected for analysis. An important aspect was their ability to perform various analyses and render many different visualisations.

- Spotfire
- QlikView
- Tableau
- JMP

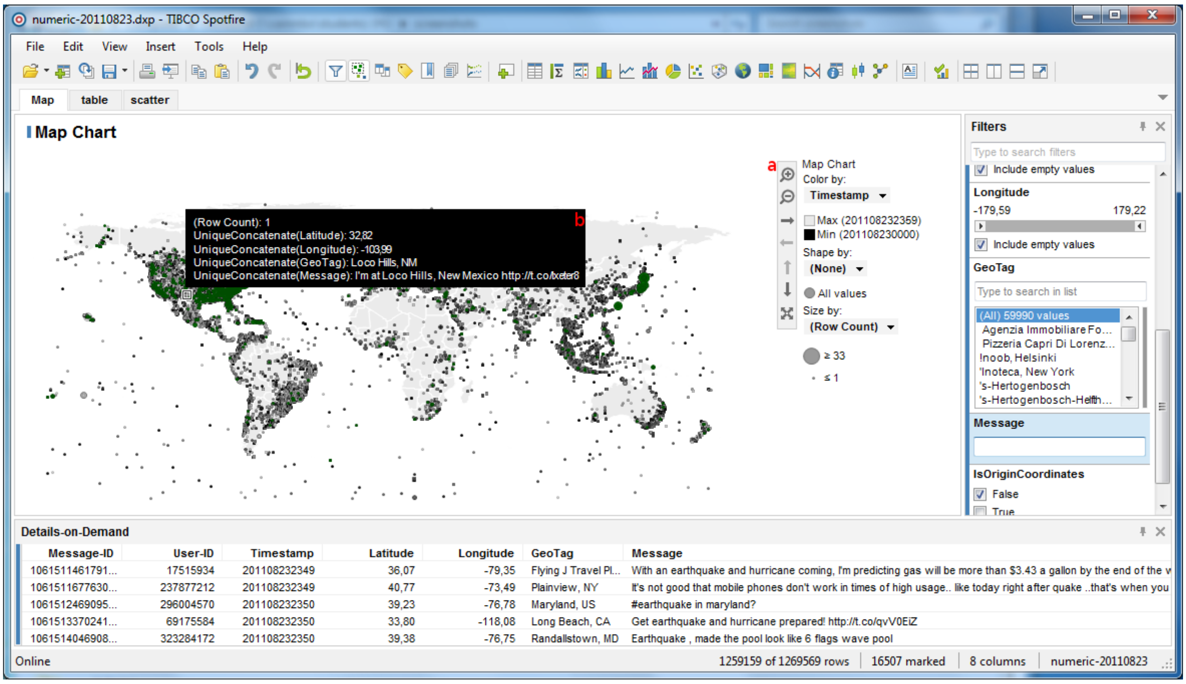
An additional tool caught our attention by promising the ability to correlate documents through the extraction and comparison of contained entities: Jigsaw.

The content of this section presents these tools in general, before they are analysed in more detail in section 3.3. To be able to get comparable results from our tests, we used a reference machine with the following configuration:

- An Intel Core i7 980X processor operating with 6 cores, simultaneous multi-threading and 3300 MHz.

- An nVIDIA Gefore GTX 580 graphics processing unit with a core clock of 772 MHz.
- 12 GiB of random access memory in a triple channel configuration of 6 equal double data rate synchronous dynamic (DDR3-SDRAM) dual inline modules operating at 1333 MHz with a column address strobe latency of 7 cycles.
- Microsoft Windows 7 Professional in the 64 bit variant with all updates available at the time of the tests applied.
- Microsoft Internet Explorer 9 with all updates available at the time of the tests applied.

3.2.2 Spotfire



(Row Count): 1
UniqueConcatenate(Latitude): 32,82
UniqueConcatenate(Longitude): -103,99
UniqueConcatenate(GeoTag): Loco Hills, NM
UniqueConcatenate(Message): I'm at Loco Hills, New Mexico http://t.co/xeter8

Map Chart
Color by: Timestamp
□ Max (201108232359)
■ Min (201108230000)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 33
• ≤ 1

Type to search filters
☑ Include empty values
Longitude
-179,59 179,22
☑ Include empty values
GeoTag
Type to search in list
(All) 59990 values
Agenzia Immobiliare Fo...
Pizzeria Capri Di Lorenz...
!noob, Helsinki
'Inoteca, New York
's-Hertogenbosch
's-Hertogenbosch-Helfth...
Message
IsOriginCoordinates
☑ False
□ True

Message-ID	User-ID	Timestamp	Latitude	Longitude	GeoTag	Message
1061511461791...	17515934	201108232349	36,07	-79,35	Flying J Travel Pl...	With an earthquake and hurricane coming, I'm predicting gas will be more than $3.43 a gallon by the end of the w
1061511677630...	237877212	201108232349	40,77	-73,49	Plainview, NY	It's not good that mobile phones don't work in times of high usage.. like today right after quake ..that's when you
1061512469095...	296004570	201108232350	39,23	-76,78	Maryland, US	#earthquake in maryland?
1061513370241...	69175584	201108232350	33,80	-118,08	Long Beach, CA	Get earthquake and hurricane prepared! http://t.co/qvV0EiZ
1061514046908...	323284172	201108232350	39,38	-76,75	Randallstown, MD	Earthquake , made the pool look like 6 flags wave pool

1259159 of 1269569 rows | 16507 marked | 8 columns | numeric-20110823

Figure 3.1: Spotfire displaying a map of the sample data with a thumbnail illustrating the data exploration capabilities.

TIBCO's Spotfire [10] is a commercial visual analytics tool, developed by TIBCO Software Inc. We chose it for further study because, according to Zhang et al. [31], it knows a number of relevant visualisations and should be able to handle our data set without problems.

General Impressions

Spotfire is available with several licenses, including a free evaluation license, which we chose for financial reasons. There seems to be no listing of the differences between the individual licenses or simply a site that lists the differences between the free trial license and the "professional" licenses. The free license is time-limited and loses the ability of doing more than trivial analyses or opening trivial save files after thirty days. In order to be able to enforce this time limit, Spotfire forces the user to log in every five days.

As a result of this constraint, the user needs to update Spotfire no later than five days after any update is published. The update process is rather time-consuming and takes at least 10 minutes on a 2.4 GHz computer with a fast internet connection. This means that the user both doesn't have the option of running a slightly outdated version which works better on their system and cannot refuse to upgrade, even if they need to do a time critical analysis.

Spotfire offers a lot of different visualisation options, see Zhang et al. [31] for more in-depth information. It feels slow with our a data set of 1.26 million Twitter messages. The program is only shipped as a 32 bit version, which means that it cannot keep large data sets in memory. As a result, it needs to use on-disk swap files to deal with such data sets.

The GUI looks nice and is quite intuitive. We were able to find several disasters, both natural and otherwise, using a simple keyword search.

Spotfire cannot read data from a stream. It can, however, update its data set from different data sources, including several databases and CSV files. This update takes between 10 and 30 seconds.

Loading and importing data from either a previously created project or an external data file takes some time, depending on the size of the data set and the available memory. The speed deficiency on low-memory systems is due to the data needing to be organised in such a way that Spotfire can keep summary data it deems important in memory, while maintaining more detailed information in an on-disk swap file.

Spotfire has one advantage over other tools, which is not a deciding factor in disaster management: it can be installed in a local directory without admin privileges. This means that users can handle their own installation on an as-needed basis, without the presence of system administrators.

The analysis tool has some minor problems when used with the newly published Windows 8, which illustrates that its target audience is business users which do not upgrade to new operating systems immediately. In the case of Spotfire, this limitation is mildly surprising, since it has a relatively quick release cycle with two new releases within five months.

Extensibility

Spotfire is based on a plugin architecture, which allows its functionality to be extended. For this purpose, there is an SDK which can be used to write new plugins. We were, however, unable to evaluate this functionality because it is only available for professional licenses. An internet search turned up a few plugins, none of which were offered by TIBCO. It is unfortunate that for this reason, we could not use more specific analyses better suited to disaster management scenarios.

Suitability for Disaster Management

Spotfire makes a nice first impression. It is able to load the sample Twitter data within five minutes, and displaying the data in a graphical map took only about 30 minutes for a first-time user. However, the amount of time it takes to update its graphical information, combined with the fact that the data cannot be read from a stream, makes disaster discovery with Spotfire a monotonous and time-consuming task.

In a disaster situation, 15 minutes is a long time. Spotfire takes this amount of time to read in new data, present it visually and let the user discover any changes that occurred since the last update. This shows the advantages of more specifically developed tools that can update and graphically represent these changes in real-time. A preprocessing step where the data is filtered according to disaster-specific criteria might help with this situation.

The lack of proper analysis tools and Spotfire's lack of extensibility further decrease its usefulness for disaster management. It does provide necessary filtering capability. Since these filters are barely configurable and every column has exactly one filter, using them can be difficult and time consuming. This fact is somewhat alleviated by the availability of "Calculated Columns", which allow the user to analyse the data using custom expressions. However, they are little better than crutches in disaster management scenarios as using them is time consuming.

In conclusion, while Spotfire is great for an after-the-fact analysis of a disaster, it is somewhat lacking in an ongoing disaster management scenario. An analyst using it will greatly profit from some specific training, as some configuration options are hidden away and some limitations are not intuitive.

3.2.3 QlikView

QlikTech's QlikView [2] is another general visual analytics tools with emphasis on business intelligence. QlikTech has provided a platform for users to share knowledge and extensions. The community that has evolved around this and other platforms is active and willing to help when someone has questions or problems.

Unfortunately, the effectiveness of this helpful community is often counteracted by the restrictions imposed on QlikView by the license used for this study. QlikTech provides a free version for individual use, called "personal edition", along with paid licensing between 1,350 and 15,000 U.S. dollars. With the personal edition, there is no need to log into any account or connect to activation servers. However, the user is limited to only viewing their own files. This makes it very difficult to collaborate with other users who, for instance, might upload a document with the solution to one's problem, because it cannot be opened.

To at least allow the user to open their own files after a reinstallation or transfer to another machine, there is a recovery feature that can be used up to four times. QlikTech provides the possibility to change the file's author four times for every installation. This registers the author's key in the file that is being opened as belonging to the current user. After the recovery, all files from the same author can be opened.

A problem not related to licensing arises because QlikView does not support data to be displayed on maps out of the box. One either has to define shapes that represent the areas on the map oneself or use third party extensions that provide dynamic background images. Therefore, the initial tests conducted for this study were limited to a map with a static background image that neither supports zooming nor provides location information beyond the user's knowledge (Figure 3.2).

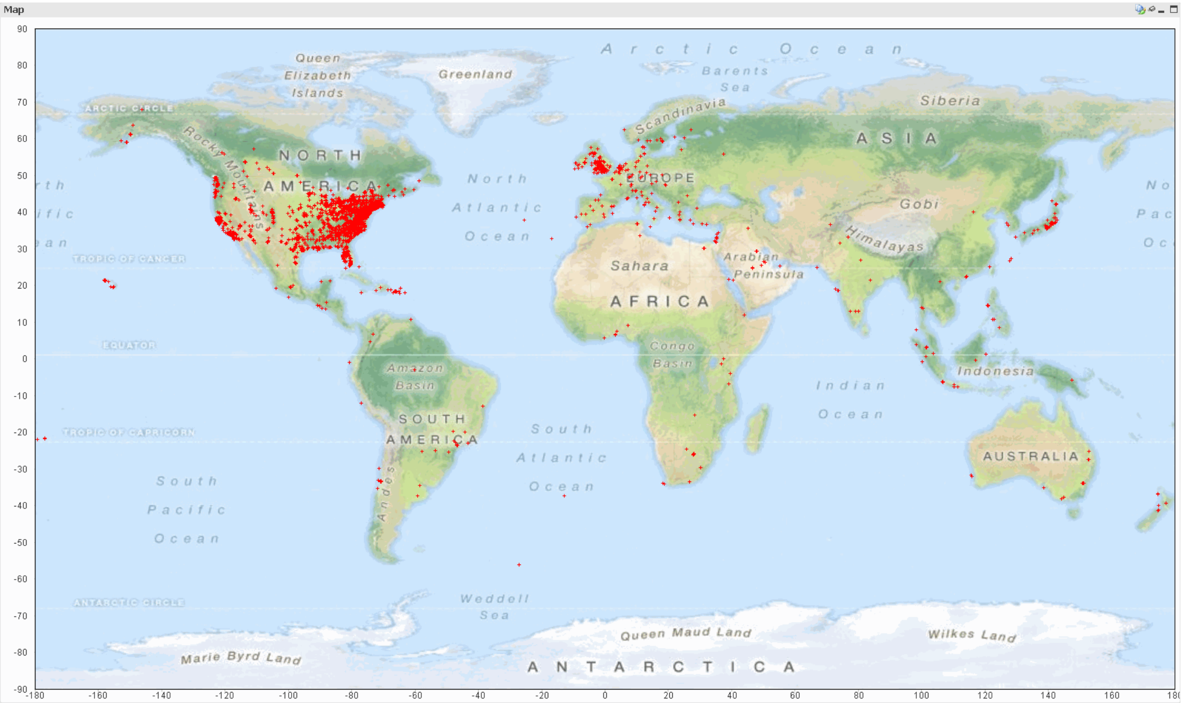


Figure 3.2: A scatter plot with a static background as map in QlikView.

QlikView always works on dedicated documents in its own format. However, the data can be retrieved from many other files or databases. While many different formats are supported, only MS Excel files are selectable when a new document is created. The other formats can be chosen when an existing document is to be opened. If the existing document is not in a QlikView format, a new one is generated and the data is imported.

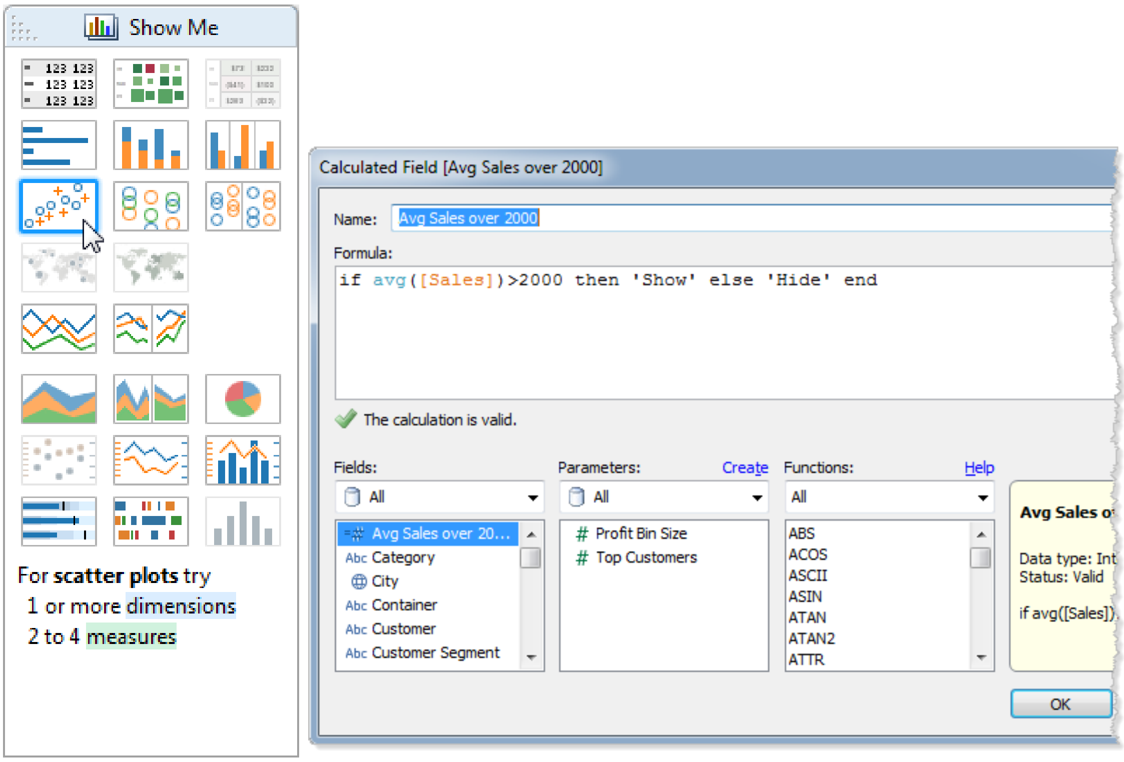
Extensibility

QlikView supports custom extensions written in Javascript. To enable them the user has to switch to the web view, which uses Microsoft's Internet Explorer to host all visual content. However, the current implementation is not compatible to the newest browser version 10 used in Windows 8 and Windows 7 with the platform update. The number of available data records in this so called "web view" is limited to 10,000.

3.2.4 Tableau

Tableau Software is an American company seated in Seattle, Washington, USA. They provide four main data visualisation products focusing on business intelligence [14]: Tableau Desktop for individual use on a personal computer; Tableau Server, which is accessible through a web browser; Tableau Public, which can create interactive web-page-embedded visualisations, but limits the amount of data records to 100,000 and only allows saving on Tableau's public sever; and finally Tableau Reader which can only read tableau package files. The latter two are provided free of charge, while the other two have different pricing policies, but time limited trial versions are also offered. All products require a 32 or 64 bit Windows platform. Additional mobile applications are also available for Apple's iOS and Google's Android.

Other than the mentioned limitations, Tableau Public, Tableau Desktop Professional and Personal versions differentiate themselves from each other mainly by the number of data sources they support. In this study, we use the Professional license kindly provided by the software company.



Show Me

For scatter plots try
1 or more dimensions
2 to 4 measures

Calculated Field [Avg Sales over 2000]

Name: Avg Sales over 2000

Formula:

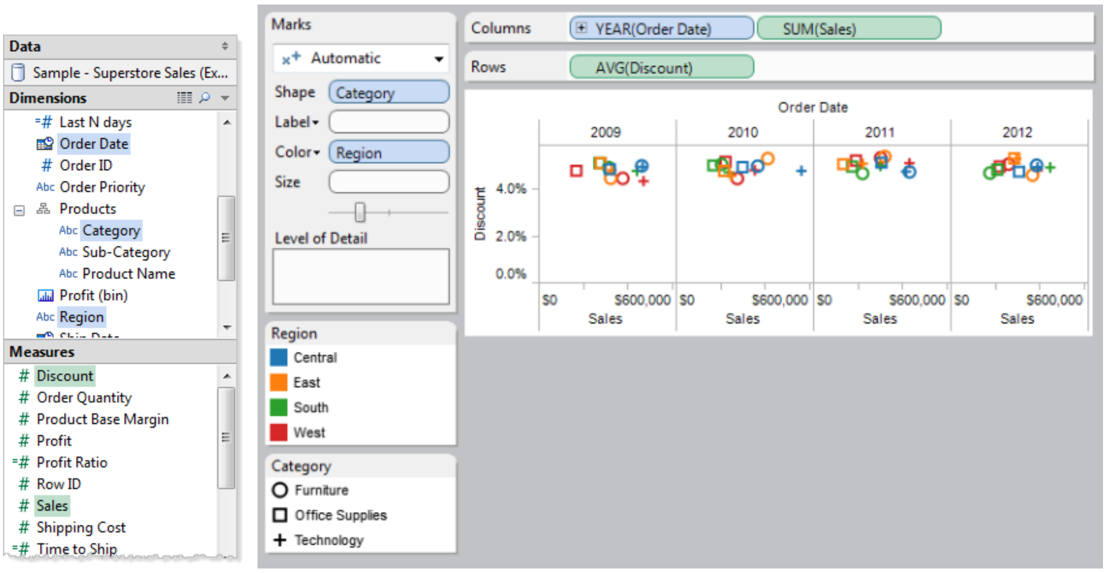
```
if avg([Sales])>2000 then 'Show' else 'Hide' end
```

The calculation is valid.

Figure 3.3: Tableau's "Show Me" visualisation type choosing toolbar (left) and dialog for creating a calculated field with field name and function completion assistance (right)

General Impressions

Tableau supports various types of visualisations through an intuitive, professional-looking, mouse-friendly user interface (Figure 3.3). Visualisations can be created by dragging and dropping. The type of visualisation can be automatically chosen based of the data types, for example, by dragging and dropping "Dimensions" and "Measures" into the "Columns" and "Rows" fields (Figure 3.4). Details and customisations can also be selected by mouse. More advanced functions, such as calculated fields or parameter definitions, are accessible through dialogs and can only be fully utilised with manual text input, although mouse assistance is offered (Figure 3.3).



Columns: YEAR(Order Date) SUM(Sales)
Rows: AVG(Discount)
Marks: Automatic
Shape: Category
Label:
Color: Region
Size:

Figure 3.4: Tableau's data source, dimensions and measures toolbar (left) – Customisation, optimisation detail toolbar and main visualisation (right)

Tableau, using the familiar concept of most of the widely-used spreadsheet applications, organises visualisations in a project file called "workbook", which contains one visualisation or a side-by-side comparison of multiple visualisations in each "sheet". A "dashboard" is a special sheet, in which other visualisations in the workbook can be placed freely to provide an overlook of the the whole project's data. A presentation mode is also available.

Even though real-time visualisation from a live data stream is not possible, Tableau supports numerous popular databases and file formats and can update the data at any time with a single click.

Extensibility

Unlike Spotfire and QlikView, Tableau is not extensible through plug-ins or extensions of any kind.

The only exception to this is that it allows the import of custom geocoding to further enhance its ability to work with maps.

It does, however, offer some filtering conditions and scripting functionality for advanced users, limited to the definition of calculated fields, which are mostly modifications of original data. Although very limited, it can be used creatively to solve some problems, for instance the problem of filtering for multiple keywords.

Suitability for Disaster Management

Tableau is supposed to “work with 100’s of million of rows of data right on your own computer and get answers in seconds” [26], which sounds very promising for our task of analysing 1.26 million Twitter messages. The software works relatively fast and responds quickly to user interactions. Most of the time-consuming operations can be aborted, should changes need to be made. Execution time and/or progress are also shown for such tasks, which could be taken as a responsiveness indicator in case of program malfunctions, so that the user can take appropriate actions to avoid wasting their time.

More importantly, Tableau offers good support for, among many others, visualisation with geographical maps, which is extremely useful for geospatial data analysis.

Advanced and quick filtering are also featured. Tableau’s flexible paging functionality, which mimics motion pictures by incrementally showing overlapping results for specific periods, is visually well-suited for observing and perceiving data changes over time.

Apart from those auspicious features of a generic visualisation tool, Tableau seems to offer next to no advanced data mining tools, especially none that could compare to dedicated solutions like Twitcident.

Nevertheless, we were able to find some encouraging results from our tests with Tableau.

3.2.5 JMP

SAS’s JMP software [7] is a general visual analytics tool with emphasis on business intelligence.

Licensing

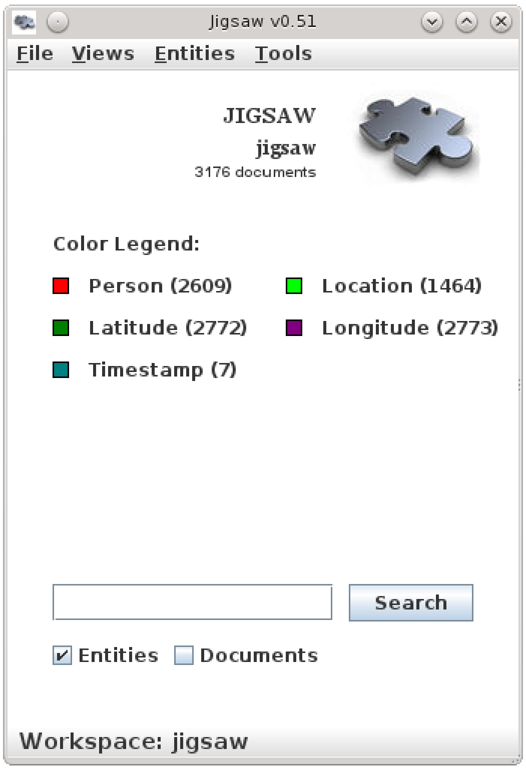
SAS provides a fully-featured trial version that will work for 30 days. Commercial licenses start at 1,200 Euros for a single user but their academic program provides licenses for periods of 6 to 12 months at much lower costs.

Suitability for Disaster Management

JMP was not able to load the timestamps in our data sets out of the box but required a user defined script to parse them. Grouping or clustering was only possible on a level known from various database systems. It only checked for exact data matches and did not support geographical distances. Once the data was loaded it could only be updated if it was provided by a database. Experiments to use a CSV file with ODBC drivers failed with error messages from the operating system.

Because of these critical failures early in the analysis, we decided that the program was not suitable for the tasks at hand. No further experiments were carried out with JMP.

3.2.6 Jigsaw



3176 documents

Color Legend:

🟥 Person (2609)
🟩 Location (1464)
🟩 Latitude (2772)
🟪 Longitude (2773)
🟦 Timestamp (7)

Workspace: jigsaw

Figure 3.5: Jigsaw's main screen after the completed import.

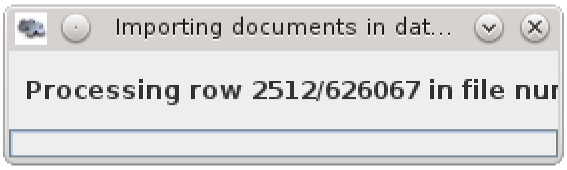
Georgia Tech Information Interfaces Research Lab's Jigsaw [6] allows the visualisation of relationships between documents by detecting entities within them. Using Twitter messages as documents would be a challenging task, since they are very short and differ wildly in grammatical and semantical quality. We chose to evaluate Jigsaw to determine its applicability to microblogs like Twitter and thus its ability to help in disaster management scenarios by finding relationships between social networking messages.

General Impressions

As opposed to the other tools evaluated in this study, Jigsaw runs natively on all three major operating systems. This is due to the fact that it is implemented in Java. There is also no license necessary for its use, which makes running it less tedious as there is no need to keep the license up-to-date or to worry about having the license run out. The installation process consists of extracting a ZIP file, which makes it quite effortless on modern systems.

Data Import

Upon reading the introduction document, the first of Jigsaw's shortcomings becomes apparent: it is designed to handle 5,000 to 10,000 short documents, not few large documents. This is problematic when combined with the fact that a collection of Twitter messages can be viewed as either one large document with many small data records or as a collection of millions of small documents. Both interpretations are outside the scope of Jigsaw's design, which becomes apparent after importing our sample data set: Figure 3.5 and Figure 3.6 both show that a large portion of the original data mysteriously went missing during the import. While Figure 3.6 shows that about 600,000 messages were ignored while processing the import, in the end only 3,176 of the original 1.26 million records survived the import process.



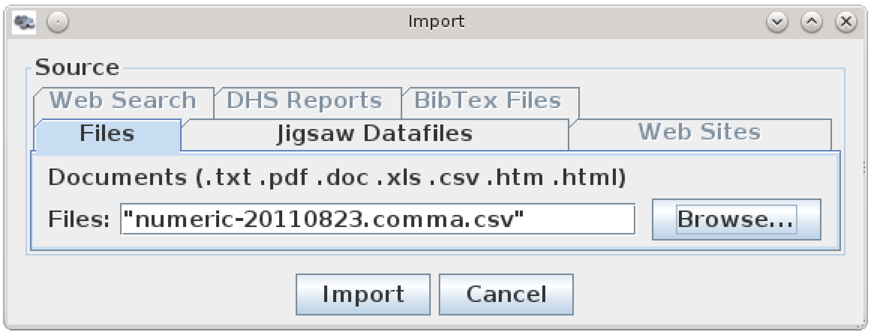
Processing row 2512/626067 in file nur

Figure 3.6: Jigsaw processing the import with approximately 600,000 records missing.

One possible way to combat this problem could be to preprocess the initial data set to include only relevant data records and regard all messages by an individual user as one document. Since the resulting data set would still contain several thousand messages and Jigsaw's entity detection doesn't work on Twitter messages, this would also be unlikely to result in a successful import. Due to time constraints, we were unable to explore this any further.

During the course of the evaluation, a document was found that appeared to contain multiple data records. This might explain the missing 600,000 messages in Figure 3.6. Since there was no message from Jigsaw to explain why it would join several data records into one document, we can only speculate that it assumed some sort of quoting. Since some Twitter messages probably contained odd numbers of quotation marks, these messages were likely joined with the subsequent data records, until another odd number of quotation marks was encountered. This theory does not, however, succeed in explaining the further reduction to 3,176 documents.

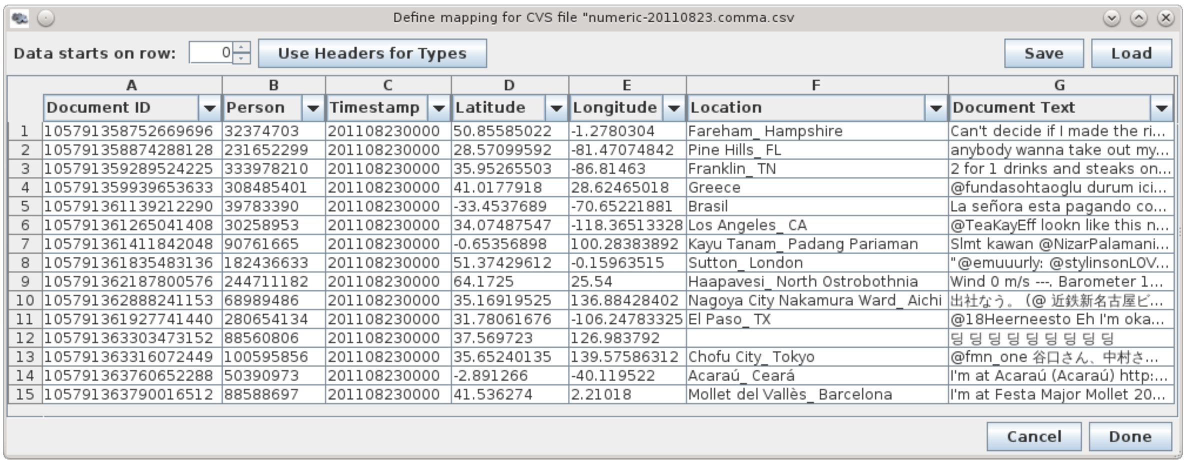
Apart from this critical import problem, which was not reported by Jigsaw, the import went smoothly. The file selection dialog (Figure 3.7) allows the selection of multiple documents, which – in the case of CSV files – need to have their columns mapped to data types. Surprisingly, however, these data types are not classical data types like "integer" or "string". They are freely definable strings, which makes the “data type mapping” a column name assignment (Figure 3.8).



Documents (.txt .pdf .doc .xls .csv .htm .html)

Files: "numeric-20110823.comma.csv"

Figure 3.7: Jigsaw’s file import dialog, showing the supported data formats.



Data starts on row: 0

	A	B	C	D	E	F	G
	Document ID	Person	Timestamp	Latitude	Longitude	Location	Document Text
1	105791358752669696	32374703	201108230000	50.85585022	-1.2780304	Fareham_ Hampshire	Can't decide if I made the ri...
2	105791358874288128	231652299	201108230000	28.57099592	-81.47074842	Pine Hills_ FL	anybody wanna take out my...
3	105791359289524225	333978210	201108230000	35.95265503	-86.81463	Franklin_ TN	2 for 1 drinks and steaks on...
4	105791359939653633	308485401	201108230000	41.0177918	28.62465018	Greece	@fundasohtaoglu durum ici...
5	105791361139212290	39783390	201108230000	-33.4537689	-70.65221881	Brasil	La señora esta pagando co...
6	105791361265041408	30258953	201108230000	34.07487547	-118.36513328	Los Angeles_ CA	@TeaKayEff lookn like this n...
7	105791361411842048	90761665	201108230000	-0.65356898	100.28383892	Kayu Tanam_ Padang Pariaman	Slmt kawan @NizarPalamani...
8	105791361835483136	182436633	201108230000	51.37429612	-0.15963515	Sutton_ London	"@emuuurly: @stylinsonLOV...
9	105791362187800576	244711182	201108230000	64.1725	25.54	Haapavesi_ North Ostrobothnia	Wind 0 m/s ---. Barometer 1...
10	105791362888241153	68989486	201108230000	35.16919525	136.88428402	Nagoya City Nakamura Ward_ Aichi	出社なう。(@ 近鉄新名古屋ビ...
11	105791361927741440	280654134	201108230000	31.78061676	-106.24783325	El Paso_ TX	@18Heerneesto Eh I'm oka...
12	105791363303473152	88560806	201108230000	37.569723	126.983792		딩 딩 딩 딩 딩 딩 딩 딩 딩
13	105791363316072449	100595856	201108230000	35.65240135	139.57586312	Chofu City_ Tokyo	@fmn_one 谷口さん、中村さ...
14	105791363760652288	50390973	201108230000	-2.891266	-40.119522	Acaraú_ Ceará	I'm at Acaraú (Acaraú) http:...
15	105791363790016512	88588697	201108230000	41.536274	2.21018	Mollet del Vallès_ Barcelona	I'm at Festa Major Mollet 20...

Figure 3.8: Jigsaw’s type mapping dialog.

Some of the predefined “types” – like “Document ID” and “Document Text” – are later interpreted by Jigsaw in a hardcoded fashion. Figure 3.10, for example, uses the contents of the “Document Text” column to populate the tooltips. It is implied that the “Document ID” column is used to identify individual documents throughout.

Data Analysis

Jigsaw provides several analysis tools (see Figure 3.9). To evaluate them, we ran them through the handy “Compute All” menu entry. This caused Jigsaw to launch all tools sequentially on the imported data and with default parameters. The run time of all analysis tools combined was more than five hours on a 2.1 GHz computer. Jigsaw, like all evaluated visual analytics tools, only used one thread to run all four analyses, despite the inherent independence of the algorithms. After the analyses completed, there were no obvious changes in the user interface and no obvious way to access the calculation results. As shown in “Visualisation”, the analysis results can be used in at least one View.

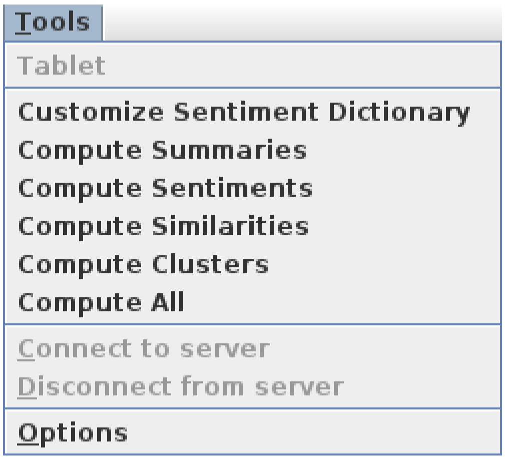


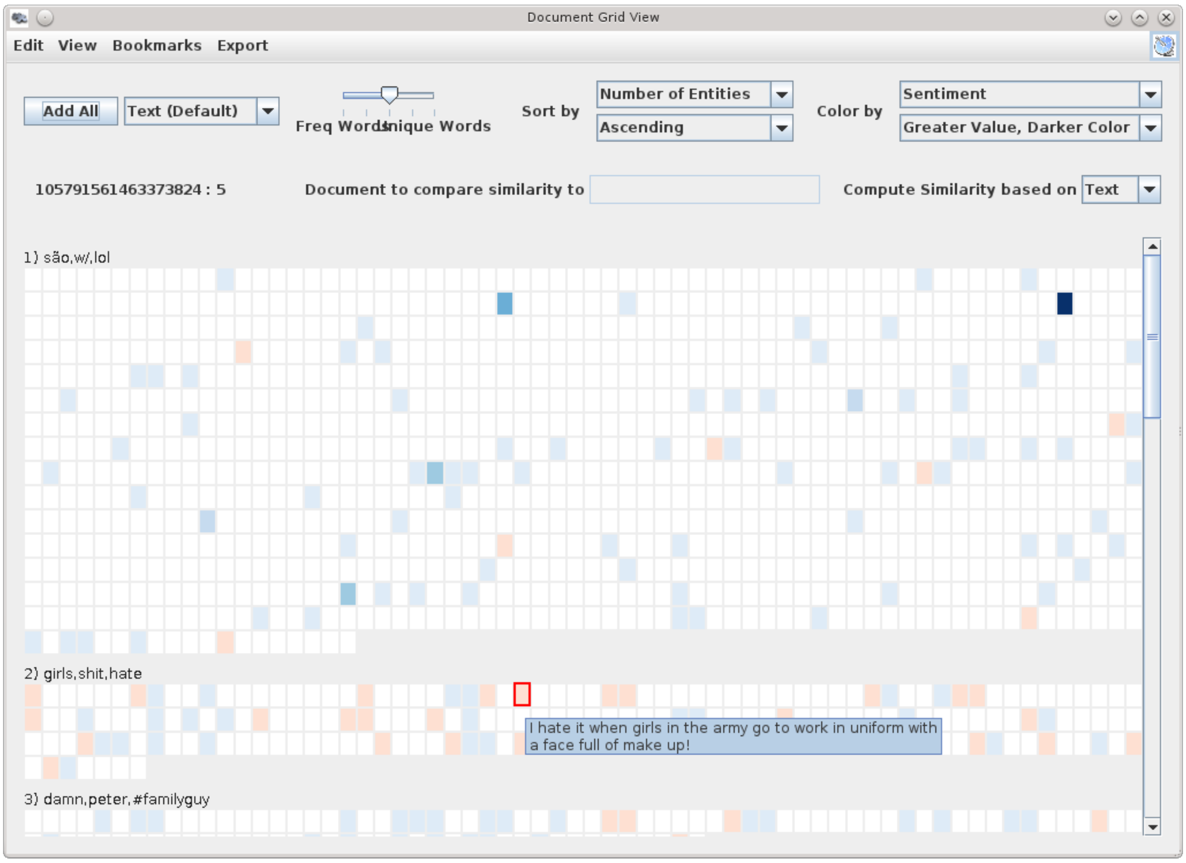
Figure 3.9: Jigsaw: Available Analysis Tools

Visualisations

Visualising the imported data proved to be somewhat difficult. Most “Views” – Jigsaw’s term for visualisations – failed to open with an Exception that was only visible on the console Jigsaw was launched from. Of the Views that did open successfully, most were empty or failed to provide useful information due to the amount of missing data. The only View that did open and provide useful information was the “Document Grid” View (Figure 3.10). This particular view allows the user to view the results of the analysis tools. To demonstrate, Figure 3.10 contains some clustering results, sorted by the amount of documents in the cluster and coloured using the result of the sentiment analysis.

Extensibility

Jigsaw contains a plugin architecture, as evidenced by the “plugins” directory in its toplevel directory and the fact that one plugin is used for code obfuscation. There are no plugins available on the project website [6], and plugin support is not mentioned. An internet search for plugins – after first yielding ambiguous results due to jigsaw puzzle plugins for software like Photoshop and the GIMP – did not turn up any results. This suggests that the plugin architecture is currently either under development or simply for internal use. Combined with the fact that the used plugin is manually added to Jigsaw in the startup file, the only logical conclusion seems to be that plugins are currently not usable to extend Jigsaw.



Text (Default)

Sort by Number of Entities

Ascending

Color by Sentiment

Greater Value, Darker Color

105791561463373824 : 5

Document to compare similarity to

Compute Similarity based on Text

1) são,w/,lol

2) girls,shit,hate

I hate it when girls in the army go to work in uniform with a face full of make up!

3) damn,peter,#familyguy

Figure 3.10: Jigsaw's Document Grid View, using the results of the sentiment analysis.

Suitability for Disaster Management

While the "Document Grid" View yielded promising results, there were severe import problems and the fact that design decisions make Jigsaw's usability for the analysis of microblogs doubtful. Due to these results, Jigsaw currently isn't a viable candidate for use in disaster management scenarios.

If Jigsaw was to be amended to properly detect entities in a microblogging context, it could serve as a valuable addition to disaster management by highlighting relationships between individual documents and therefore find details relevant to current areas of interest. To be truly great at this, the analysis tools provided by Jigsaw need to be optimised since they currently cannot exploit parallelism in hardware and are generally very slow with the amount of data provided by social networks.

Because of Jigsaw's general unfitness to the task at hand, we have decided not to include it in our tool comparison.

	Spotfire	**QlikView**	**Tableau**	**JMP**	**Jigsaw**
Data import from DB	✓	✓	✓	✓	–
Data import from CSV	✓	✓	✓	✓	✓[a]
Live data streams	–	–	–	–	–
Duplicate elimination	✓	✓	–	✓	–

Table 3.1: Data sources that are supported by the evaluated programs.

[a] partial import

3.3 Tool Comparison

Our in depth analysis of the selected tools shows how well they fare in the realm of disaster management scenarios. To obtain comparable results, the same input data and set of tasks were used. This allows us to rank the tools according to their suitability for disaster management.

3.3.1 Data Import

At the beginning of any workflow lies the import of data into the program. All tools can load data in the "character separated format" (CSV). However, not all treat the text they read in a semantically correct way.

The format of the timestamps in our source data was *YYYYMMDDhhmm*. This way, "long integers" yielded a valid temporal sequence. Since every tool would have required substantial manual labour to load these strings as temporal data, we decided to leave the long integer interpretation.

A tool would ideally know the semantics of geographical coordinates, but their treatment as decimal numbers is still adequate. JMP supports coordinates with degrees, minutes and seconds and handles them as a separate data type. However, since the data isn't supplied in this format, this doesn't provide an additional advantage with our data set.

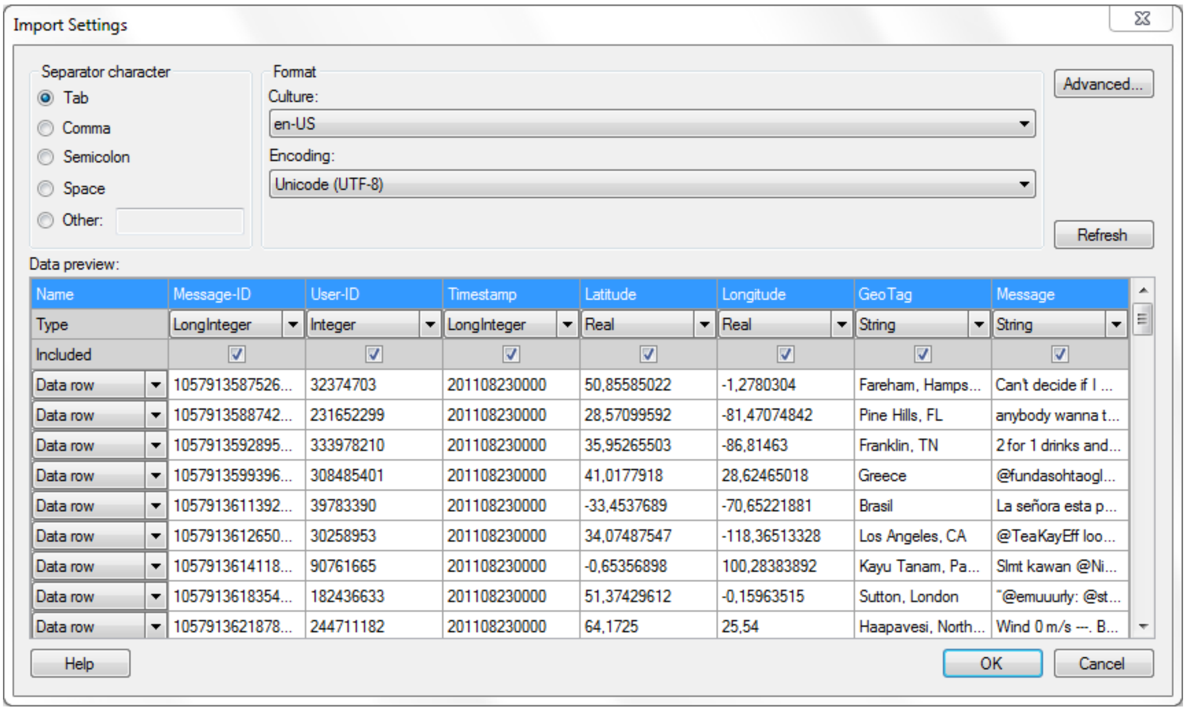
As Table 3.1 shows, all programs support at least one database system as input. JMP even needs such a system to be able to reload the data and was therefore excluded from further testing. Otherwise, the user is stuck with the data that was previously imported. Live data streams could also be helpful in discovering disasters and other information in real time, but none of the tools support them.

Every program handles the data import in its own way. We will therefore look at each tool individually to be able to properly reproduce the advantages and disadvantages offered by each tool. JMP and Jigsaw both proved to have some fatal flaws, which is why we chose to exclude them from further analysis.

Spotfire

When creating a new project, the user has the option to create a blank project, to import an existing one or to start with a data import. Since we had a predefined data set of Twitter messages, we chose the import option.

The CSV file was properly detected as tab-separated and the assigned data types were chosen appropriately. There were only three problems with the automatic configuration. Spotfire was not able to handle the timestamp format. As discussed above, we chose to interpret them as integers. The second issue was with the document encoding detection. Spotfire assumed a “Western European” encoding – which probably means ISO-8859-1 – instead of the UTF-8 encoding the document was actually in. Lastly, Spotfire assumed that there was quoting within the data records and enabled the corresponding option in the import configuration. Since any quoting in the data set was purely coincidental, this option had to be disabled. As a final step to completing the configuration, we named the individual columns according to their contents. The resulting import configuration is shown in Figure 3.11.



Separator character

● Tab
○ Comma
○ Semicolon
○ Space
○ Other:

Format

Culture: en-US

Encoding: Unicode (UTF-8)

Data preview:

Name	Message-ID	User-ID	Timestamp	Latitude	Longitude	GeoTag	Message
Type	LongInteger	Integer	LongInteger	Real	Real	String	String
Included	☑	☑	☑	☑	☑	☑	☑
Data row	1057913587526...	32374703	201108230000	50,85585022	-1,2780304	Fareham, Hamps...	Can't decide if I ...
Data row	1057913588742...	231652299	201108230000	28,57099592	-81,47074842	Pine Hills, FL	anybody wanna t...
Data row	1057913592895...	333978210	201108230000	35,95265503	-86,81463	Franklin, TN	2 for 1 drinks and...
Data row	1057913599396...	308485401	201108230000	41,0177918	28,62465018	Greece	@fundasohtaogl...
Data row	1057913611392...	39783390	201108230000	-33,4537689	-70,65221881	Brasil	La señora esta p...
Data row	1057913612650...	30258953	201108230000	34,07487547	-118,36513328	Los Angeles, CA	@TeaKayEff loo...
Data row	1057913614118...	90761665	201108230000	-0,65356898	100,28383892	Kayu Tanam, Pa...	Slmt kawan @Ni...
Data row	1057913618354...	182436633	201108230000	51,37429612	-0,15963515	Sutton, London	"@emuuurly: @st...
Data row	1057913621878...	244711182	201108230000	64,1725	25,54	Haapavesi, North...	Wind 0 m/s ---. B...

Figure 3.11: Spotfire’s import settings dialog.

After these parameters were determined, the actual data import was triggered. The import took roughly 15 minutes, and included a duplicate elimination step where duplicate rows were eliminated. There were no problems during the import: errors in the expected data format were ignored and reported after the import, but didn’t cause any failures.

Spotfire has the option of saving the imported data within the project file or simply keeping a link to the source file. Saving the data as part of the project has the advantage of not having to repeat the import every time it is loaded. This means that saving the project takes a little longer, though. In case of a link to the source, the re-import on load happens automatically, so no user interaction is required, but it does increase the load time.

Since Spotfire cannot read from a stream, the data sometimes needs to be refreshed. This refresh operation is realised as a new import with an immediate join of the imported and existing data tables.

Overall, the data import step was easy and intuitive, albeit a bit lengthy due to internal optimisations and restructuring of the data.

QlikView

Like many other functions of QlikView, the data import is based on a script (Figure 3.12). However, the user does not have to write it himself. A wizard is used to gather information about the source file's format. Here, the tab-separation and the UTF-8 encoding were detected automatically, but the quoting had to be turned off manually. After the different data columns were named, the script was generated and displayed. At this point, the user could specify different options for the representation of numbers, currencies, dates and timestamps, while having the default values as templates. This allowed the correct handling of the different data types.

QlikView reads all data, duplicate or not. Built-in visualisations only show distinct records, but the duplicates still appear in evaluations, such as aggregations. Thus the user has to keep redundant data in mind while editing the import script and include the word "DISTINCT" to cause QlikView to eliminate duplicates.

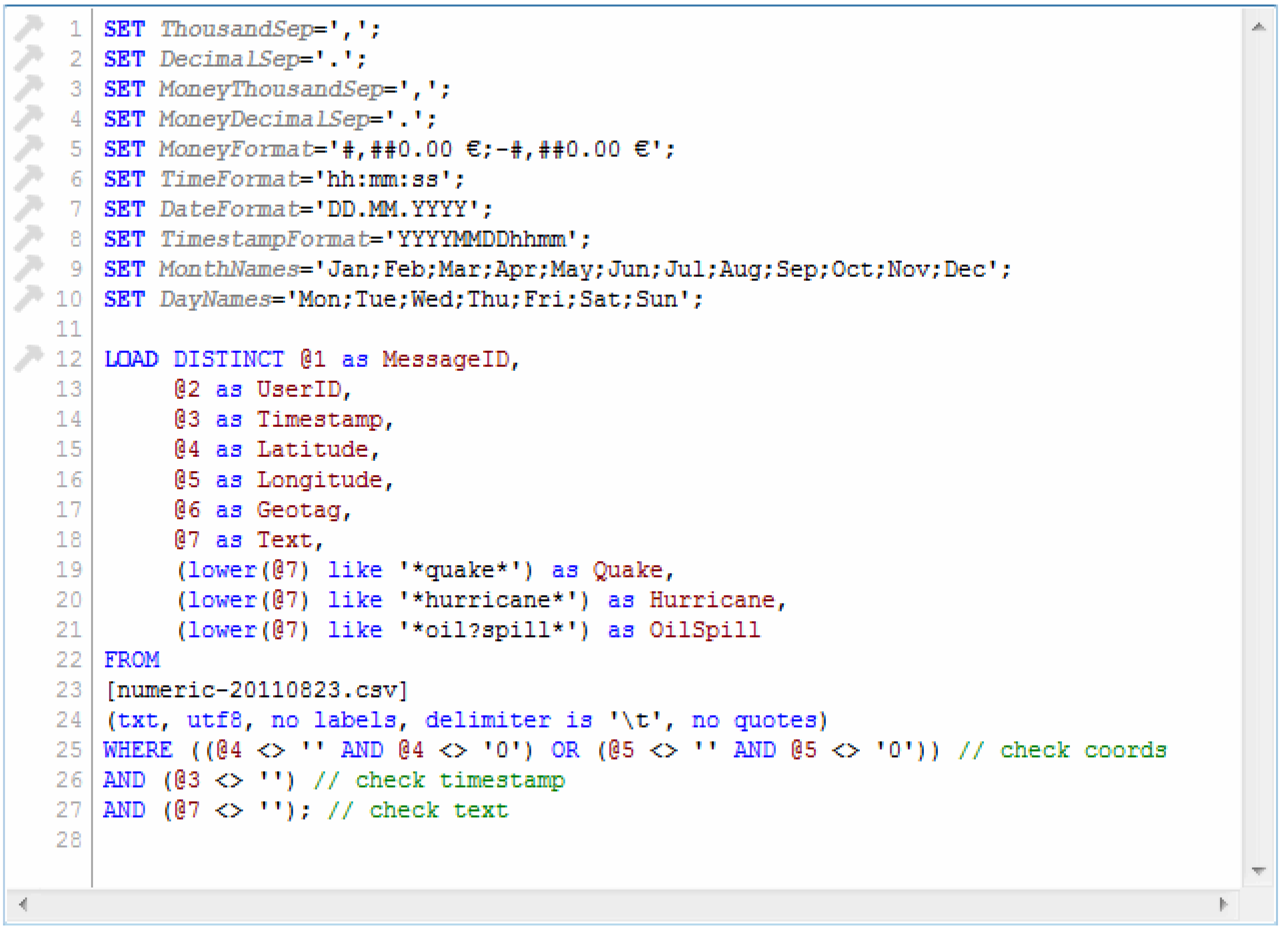
Once the data is loaded, it has to be stored in QlikView's project file. When a new document is created instead of opened, it's data can only be saved in Microsoft's various spread sheet formats for Excel.

QlikView lacks a concept of live data streams. However, it can synchronise with a data server or reload the source files, which took only seconds with our sample data set. This is fast enough not to disturb the user's workflow and is therefore not considered a major downside compared to having live streams.

Tableau

Tableau supports more than 30 sources of data, from static files like Microsoft® Excel or CSV to widely used relational database systems, even from clipboard. Furthermore, unsupported data sources could be indirectly accessed through the standard ODBC connection, which leaves the door open for the widespread Web 2.0 data exchange formats, such as XML (RSS, Atom) or JSON.

Data can be read entirely or partially at connection time, incrementally only when needed, directly from the sources or stored locally in a separate file as an extract of data. The software



```
SET ThousandSep=',';
SET DecimalSep='.';
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='#,##0.00 €;-#,##0.00 €';
SET TimeFormat='hh:mm:ss';
SET DateFormat='DD.MM.YYYY';
SET TimestampFormat='YYYYMMDDhhmm';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';

LOAD DISTINCT @1 as MessageID,
     @2 as UserID,
     @3 as Timestamp,
     @4 as Latitude,
     @5 as Longitude,
     @6 as Geotag,
     @7 as Text,
     (lower(@7) like '*quake*') as Quake,
     (lower(@7) like '*hurricane*') as Hurricane,
     (lower(@7) like '*oil?spill*') as OilSpill
FROM
[numeric-20110823.csv]
(txt, utf8, no labels, delimiter is '\t', no quotes)
WHERE ((@4 <> '' AND @4 <> '0') OR (@5 <> '' AND @5 <> '0')) // check coords
AND (@3 <> '') // check timestamp
AND (@7 <> ''); // check text
```

Figure 3.12: An automatically generated QlikView import script with manual customisations.

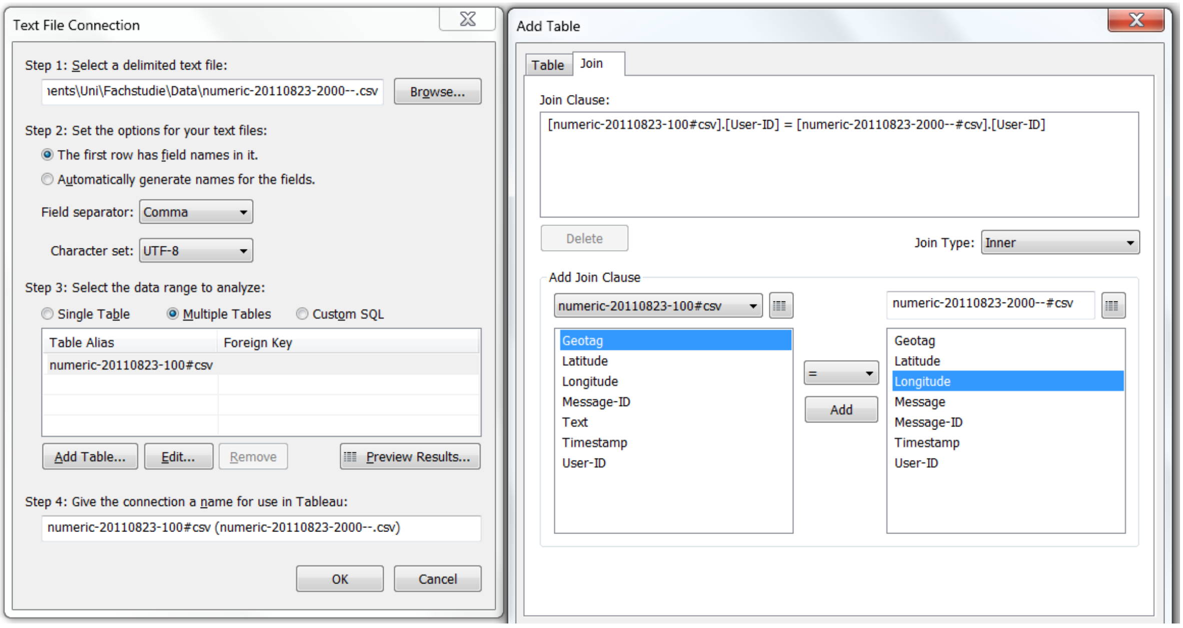
can work with the data extract as well as with a "real" data source. This extract, however, offers a significant performance improvement, especially when working with networked data sources. It sometimes offers features that don't exist in the original data sources, for instance the *count distinct* or a median calculating function for Excel files [16]. These data extracts can be included in a packaged workbook and seamlessly opened on another computer.

Tableau can visualise from different data sources simultaneously. It also supports referential constraints across multiple tables as known from relational databases, for example the "inner join",. This feature is also applicable to tables created from non-relational data sources.

when "Connect to Data" is chosen and a CSV file is selected, Tableau asks the user for the *delimited text file's* field[1] separator and encoding – which is correctly detected as UTF-8. The user can choose whether the first row of the file contains field names. Single or multiple table importing is allowed, but the multiple table feature only works for files in the same folder in

[1] In Tableau, "columns" are called "fields".

this dialog. Tableau provides a second dialog to help the user join multiple tables by identifying the primary and foreign key fields. Inner, left and right joins are supported. There's an option in the file import dialog to allow a custom importing SQL expression to be edited. Aliases for tables and fields can be named as well. Figure 3.13 illustrates some of these features.



Text File Connection

Step 1: Select a delimited text file:

ıents\Uni\Fachstudie\Data\numeric-20110823-2000--.csv

Step 2: Set the options for your text files:

● The first row has field names in it.

○ Automatically generate names for the fields.

Field separator: Comma

Character set: UTF-8

Step 3: Select the data range to analyze:

○ Single Table ● Multiple Tables ○ Custom SQL

Table Alias	Foreign Key
numeric-20110823-100#csv	

Step 4: Give the connection a name for use in Tableau:

numeric-20110823-100#csv (numeric-20110823-2000--.csv)

Add Table

Join Clause:

[numeric-20110823-100#csv].[User-ID] = [numeric-20110823-2000--#csv].[User-ID]

Join Type: Inner

Add Join Clause

numeric-20110823-100#csv		numeric-20110823-2000--#csv
Geotag	=	Geotag
Latitude		Latitude
Longitude		Longitude
Message-ID		Message
Text		Message-ID
Timestamp		Timestamp
User-ID		User-ID

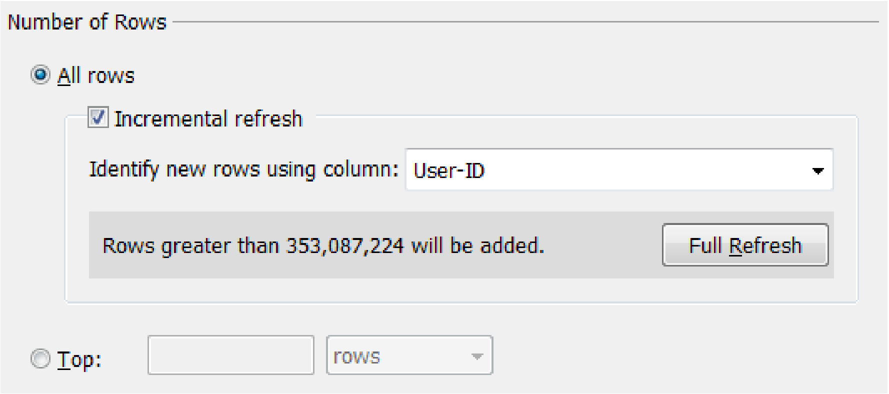
Figure 3.13: Tableau import dialogs.

The user then has the option to connect to the data live, import all or just some of it. If one of the latter two options is chosen, Tableau reads the data from the source, creates an extract, stores it locally in a special format which allows the software to work with the it independently of the data source's speed. This data extract, just like the data from a live connection, can be updated at any time in the workflow. Tableau also allows updating data extracts incrementally by appointing a field to indicate new data records, which usually is the primary key (Figure 3.14). This feature only works with integer values, which is unfortunate, because Tableau interprets the message IDs in our data sample as double precision floating point due to their big values and thus can't make use of the feature in our test scenarios.

Tableau provides no option for eliminating duplicate data.

Tableau automatically recognises some primitive data types like numbers, text, etc. It can even assign appropriate geographical roles to fields if their names are provided. The geographical role or the data type of fields can be chosen manually.

More complex data formats can be specified by choosing common predefined data formats or by creating fully customisable so-called "calculated fields" based on the original data with the aid of scripts and a wide variety of predefined mathematical, statistical, text manipulating, etc. functions.



Number of Rows

● All rows

☑ Incremental refresh

Identify new rows using column: User-ID

Rows greater than 353,087,224 will be added.

○ Top: rows

Figure 3.14: Tableau's incremental data update feature.

We did, however, encounter a problem with the data type interpretation of Tableau. It doesn't have an option to specify the decimal separator and uses the system's number format, instead of the interface language's default. This caused a misinterpretation of the geographic coordinates in the sample data and thus lead to a critical performance problem, which will be described more accurately in the detailed performance evaluation (3.3.6 Perfomance, Tableau).

After the problem had been identified, we found two solutions without having to change the original data: changing the operating system's default number format or creating a calculated field, which is much more complex because of the data's format. For example, a latitude value of 40,233483 (40° 14' 1" N) was interpreted as an integer with the value of 40233483. The calculated latitude field could be given by the formula: New Latitude = Original Latitude/10^6. However, in the sample data, the latitude does not always have six numbers after the decimal separator, which makes this solution much more complicated than it already is.

Like the other tools, Tableau doesn't interpret the data sample's timestamp format correctly. A calculated field can solve this problem.

	Spotfire	**QlikView**	**Tableau**
Date & Time (i/mi/r/mr)[a]	✓/✓/✓/–	✓/✓/✓/–	✓/✓/✓/–
Numerical (i/mi/r/mr)[a]	✓/✓/✓/–	✓/✓/✓/–	✓/✓/✓/–
Text (s/w/f)[b]	✓/✓/–	✓/✓/✓	✓/✓/–
Coordinates (s/r/gr/cs)[c]	✓/✓/✓/–	✓/✓/✓[d]/✓[d]	✓/✓/–/–
Invert filters	–	✓	✓
Selection history	✓	✓	–
Persistent selections	✓	✓	✓
Custom expressions	✓	✓	✓
Intuitive quality measurement	–	–	–
Intuitive distance measurement	–	–	–
Intuitive density measurement	–	–	–

Table 3.2: A comparison of processing and filtering capabilities.

[a] i = item filter; mi = multiple items filter; r = range filter; mr = multiple range filters

[b] s = substring; w = wildcards; f = fuzzy

[c] s = single point; r = range; gr = geographical area; cs = custom shape

[d] Functionality is provided by extensions.

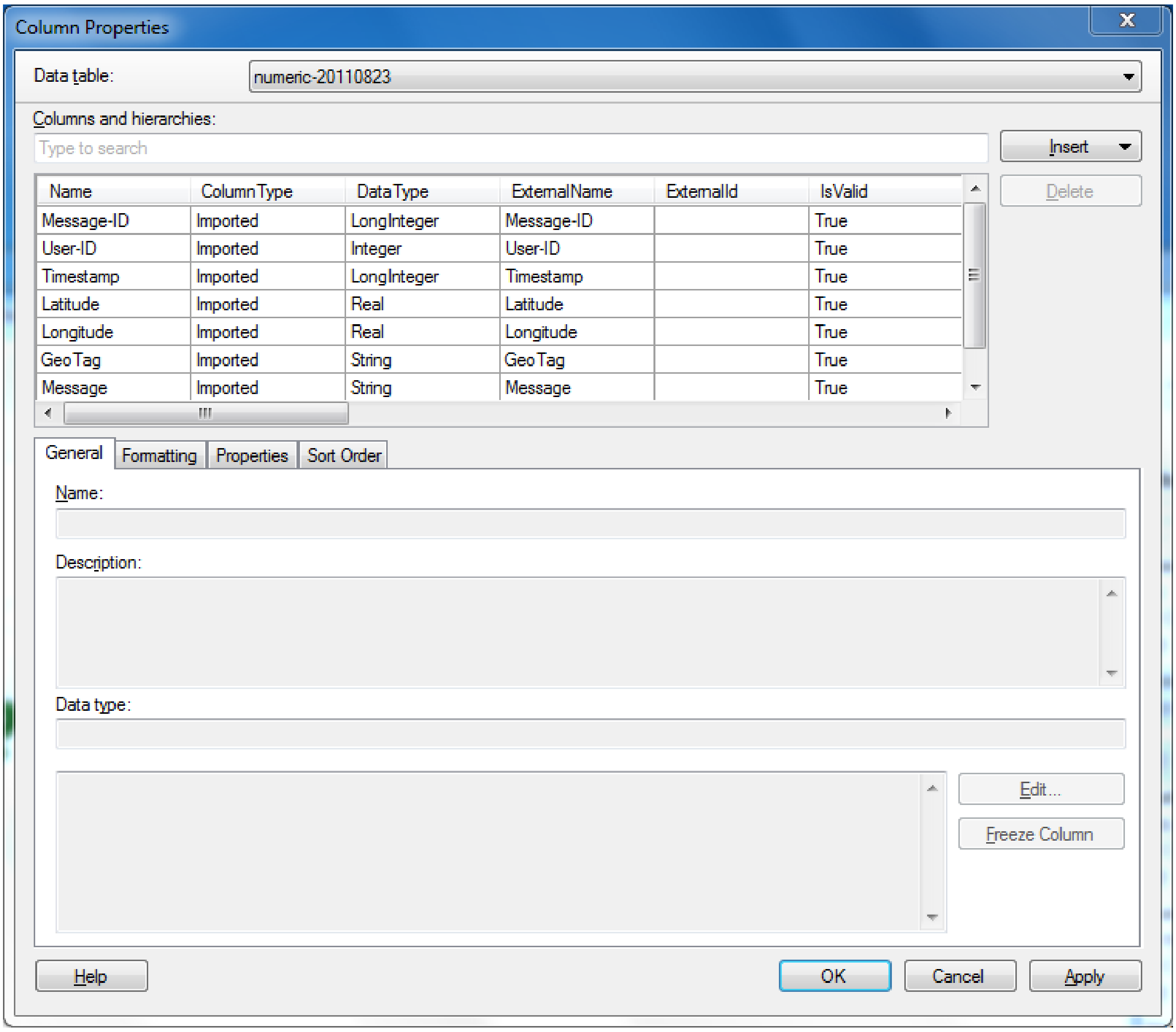
3.3.2 Processing and Filtering Data

Once the data has been imported, it has to be processed and filtered to extract and find relevant information. Processing is necessary to be able to link individual Twitter messages to events. Filtering discards irrelevant data to make it more manageable. All tools support this in one way or another, but differ in the details of their execution.

Spotfire

There is no easy interface for summarising data. Spotfire does, however, offer a proprietary expression language that can be used for manipulating data in the internal tables. Figure 3.15 shows the metadata of the sample data's table and the interface for adding new "Calculated Columns". These "Calculated Columns" contain an expression which is evaluated to summarise data from other columns. The evaluated data is cached in the project file, which increases the saving and loading time, but allows Spotfire to use the results without having to constantly re-evaluate the expressions. For our sample Twitter data, this resulted in a significant overall speed up.

To test this functionality, we created a summary column called "IsOriginCoordinates", which used an expression that evaluated to `true` for data records with coordinates in the geographical centre of the map. On the surface, the expression language was quite intuitive, with advanced functionality available but subtly hidden from the inexperienced user. The expression editor contains a graphical drag-and-drop interface which allows it to be used without prior training.



Data table: numeric-20110823

Name	ColumnType	DataType	ExternalName	ExternalId	IsValid
Message-ID	Imported	LongInteger	Message-ID		True
User-ID	Imported	Integer	User-ID		True
Timestamp	Imported	LongInteger	Timestamp		True
Latitude	Imported	Real	Latitude		True
Longitude	Imported	Real	Longitude		True
GeoTag	Imported	String	GeoTag		True
Message	Imported	String	Message		True

Figure 3.15: Spotfire's internal table of Twitter data.

For any problems that occur, the help files are readily accessible and contain descriptions of most syntactical elements.

To facilitate filtering, Spotfire uses six filter types:

Range Filter This filter type provides a slider which allows the selection of a range of data. In Figure 3.16 (a), we use this filter type for the "Latitude" column.

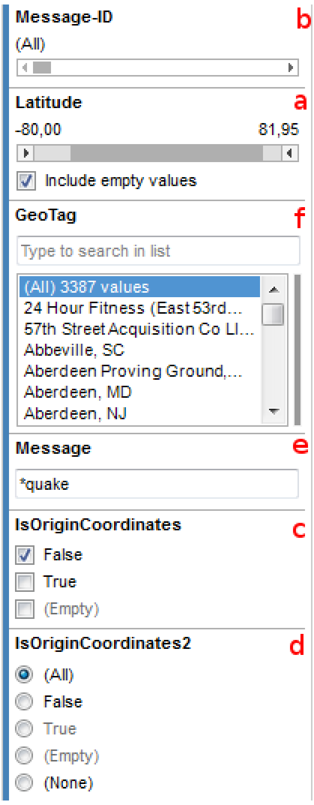
Item Filter The *Item* filter is similar in appearance to the *Range* filter, but only allows one data record to be chosen. The far right and left hand sides of this filter type have special meanings: the far left contains the "(All)" filter, which matches all data records, while the far right contains the "(None)" filter, which deselects all records. This filter type can be seen in Figure 3.16 (b), used for the "Message-ID" column.

Check Box Filter This filter type is used for columns containing discrete data like "boolean" values. It allows the selection of one or more of these discrete values. An example of this type can be seen in Figure 3.16 (c) for the "IsOriginCoordinates" column.

Radio Button Filter This type is similar to the *Check Box* type, with two exceptions. On the one hand, it only allows the selection of one value. On the other hand, it adds the options "(All)" and "(None)", which work like their counterparts in the *Item* filter type. The "IsOriginCoordinates2" column in Figure 3.16 (d) demonstrates this filter type.

Text Filter The *Text* filter allows matching of column contents to freely definable strings with wildcards. To search our sample data set for any messages containing a reference to an earthquake, one needs to use this filter type on the "Message" column, as demonstrated in Figure 3.16 (e). One would then enter the search term "*quake" to match all messages containing a word ending in "quake". The *Text* filter is case-insensitive.

List Box Filter Figure 3.16 (f) contains an example of this filter type used for the "GeoTag" column. It allows the selection of any combination of values found in a column. Like the *Item* and *Radio Button* filters, it also contains an entry named "(All)", which matches all values. It does not contain a "(None)" entry.



Message-ID
(All)

Latitude
-80,00 81,95
☑ Include empty values

GeoTag
Type to search in list
(All) 3387 values
24 Hour Fitness (East 53rd...
57th Street Acquisition Co Ll...
Abbeville, SC
Aberdeen Proving Ground,...
Aberdeen, MD
Aberdeen, NJ

Message
*quake

IsOriginCoordinates
☑ False
□ True
□ (Empty)

IsOriginCoordinates2
● (All)
○ False
○ True
○ (Empty)
○ (None)

Figure 3.16: Filter types supported by Spotfire.

Some of these filter types are data type specific. As an example, the *Check Box* and *Radio Button* filters cannot be chosen for text or integer data, but are applicable to boolean values.

Spotfire does not support any other filters, and the supported filters cannot be configured other than through their visible elements. This means that any advanced filtering needs to be done using “Calculated Columns”. It has the advantage that the default filter chosen by Spotfire is usually suitable for the column’s data type. Using advanced methods, the user can implement inverse filters, which are one of the more serious omissions of the filtering system.

An example of how to do more advanced filtering can be found in the previously created “Calculated Column” named “IsOriginCoordinates”. The contents of this column depended on the contents of both the “Latitude” and “Longitude” columns: if both were 0, this column contained `true`. One special case occurred in cases where the import resulted in one of the values being empty. In this case, “IsOriginCoordinates” contained the value “(Empty)”. If only the rows containing `false` were selected by, for example, a *Radio Button* filter, only those messages not in the geographical centre of the map were displayed.

Spotfire only allows the user to configure exactly one filter per column. This means that both multiple filters for one column are not supported and each column always has a filter. These filters are always joined with a logical `AND` operation. It would have been nice to be able to configure a join expression, so that filters could be joined with at least `OR` and possibly more complex operators. This would have provided the user with more flexibility and the ability to, e.g. explore multiple disasters in parallel, somewhat alleviating Spotfire’s lack of speed.

QlikView

QlikView allows selecting individual messages for display. The user can enter text into a searchbox and select all messages that contain this text. Selecting messages that were written at a certain point in time is best done by using a slider. Sliders can be configured to select ranges so that a period of time can be selected. There is no native functionality to select only messages inside a certain geographical area, but the user can select a rectangular area on a scatter plot to limit the selection. A user defined function can check the coordinates and perform a selection, too. This even allows an arbitrary shape for the selection, as long as the user implements a test to see whether a message was written within that area.

QlikView’s project files contain all applied filters and selections, thus allowing the user to continue at the last saving point. Furthermore, the program provides data type independent functionality for inverting existing selections, as well as allowing the user to move forward and backwards through their history.

While the user can use generic search fields to restrict the displayed data, this process can be optimised by moving it into the load script. This is realised by performing a substring search and saving the result in a new, calculated column. These new columns can then be used to, for instance, quickly select different disaster types, as if the user had check boxes at their disposal.

The messages' timestamps can be bound to a slider control to allow the selection of single values or ranges. The user can then manipulate it with the mouse or keyboard. If the number of selected messages is not too large, this can lead to a real time update of the visual representation. QlikView also provides built-in support for animations. They can be used to control the selected time range and therefore view an animated map where new messages appear and older ones disappear. This effect can be used to create time lapse videos of tweets all over the map.

Finding the first occurrence of a keyword is not a feature that is explicitly advertised. This can be achieved by displaying the data in a table, filtering it by keyword and sorting the data by timestamps.

Tableau

Tableau provides many options for data filtering, either intuitive with drag and drop, on an intermediate level with simple conditions or more advanced by scripting.

Data can be filtered completely out of the working project at the time of import, which like the extract, can be changed at any time.

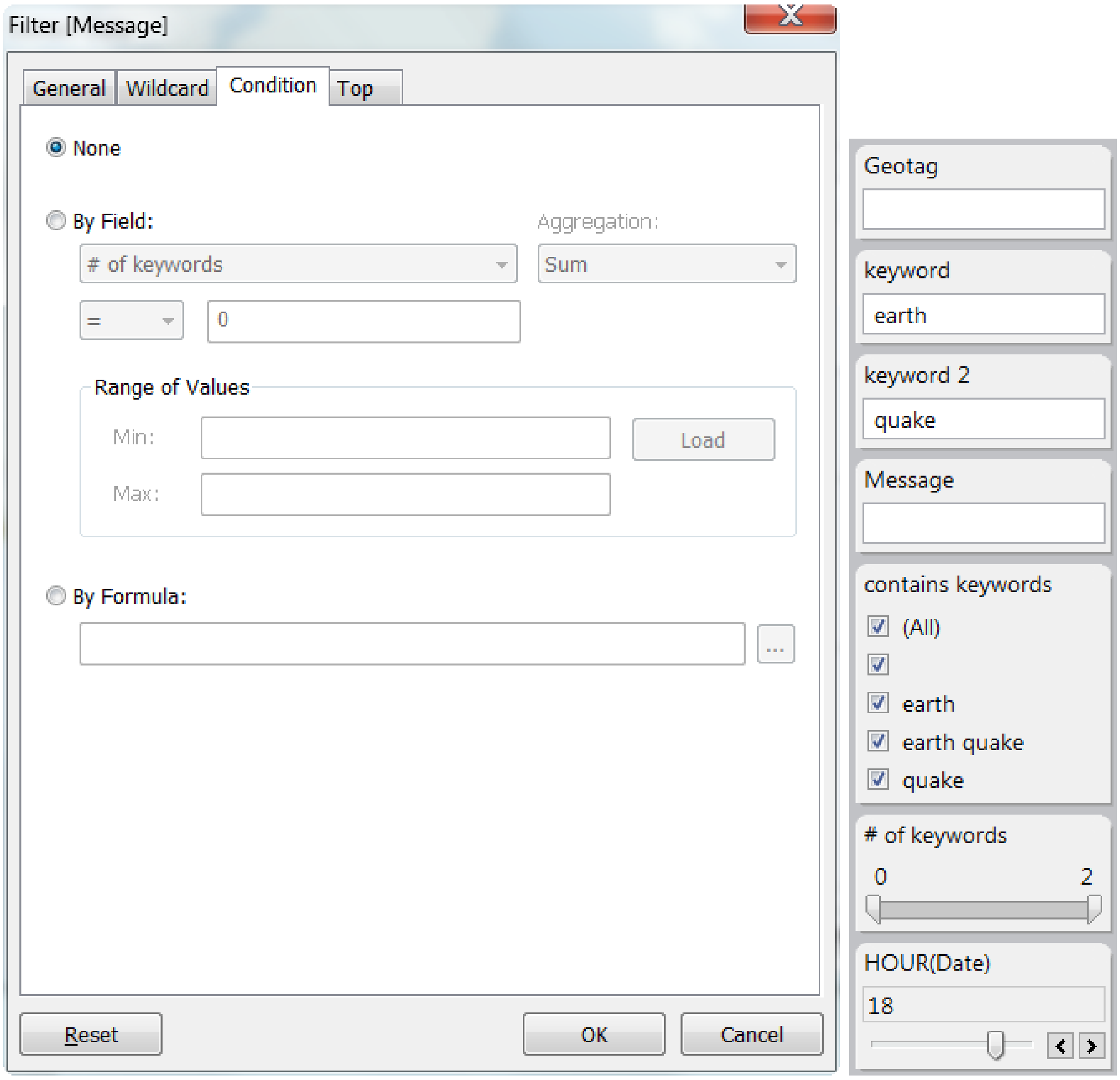
Tableau supports filtering of single or multiple exact values. It also accepts values from a list, wildcard searches, a specific number of rows on top or advanced conditions involving multiple data fields and their aggregations (SUM, AVERAGE, MIN, MAX, etc.) (Figure 3.17).

For numerical, date and time fields, a custom range of values can be defined for filtering.

Tableau doesn't support multiple keywords filtering. This, however, can be bypassed by combining the power of the calculated fields and a parameter. For instance, to create a second filter for a user input keyword at run time, a parameter "keyword 2" could be created. A calculated field "contains keyword 2" which checks if the to-be-filtered field contains this keyword and returns a simple value like a boolean or an integer. The filter can finally be constructed on the "contains keyword 2" field, effectively selecting data records which fulfil the condition. This task is computationally intensive and would decrease the tool's performance, especially when multiple keywords are involved. For our study, we created two extra parameters functioning as keyword filters for the text message in the sample data. Another parameter called "Number of keywords", which has integer values from zero to two, indicates how many of the keywords a specific message contains.

Data can be selected directly on the visualisation and chosen to be included or excluded from the visual representation or even from the internal data set. This also works for other filtering methods.

Quick filter panels (Figure 3.17) are available right beside the visualisation, which makes quick changes much more convenient.



Filter [Message]

● None

○ By Field:

# of keywords

Aggregation: Sum

= 0

Range of Values

Min:

Max:

○ By Formula:

Geotag

keyword: earth

keyword 2: quake

Message

contains keywords

☑ (All)

☑

☑ earth

☑ earth quake

☑ quake

# of keywords: 0 2

HOUR(Date): 18

Figure 3.17: Conditional filtering (left) and quick filter panels (right) from Tableau.

3.3.3 Visualisations

After a tool has filtered and summarised data, it still has to be understood by humans. Visualisations facilitate this process by presenting it in a more comprehensible way. Geographical meta information is very important to disaster management. This means that it needs to be visualised appropriately, for example on a map chart. There are two general approaches to realising this visualisation:

1. Draw a scatter plot with an appropriately stretched image of the relevant geographical area as background.

	Spotfire	QlikView	Tableau
Scatter plot	✓	✓	✓
Map chart	✓	✓[a]	✓
Heat map	–	✓[a]	✓
Table	✓	✓	✓
Custom information in tooltips	✓	✓	✓

Table 3.3: A comparison of visualisation capabilities.

[a] Functionality is provided by extensions.

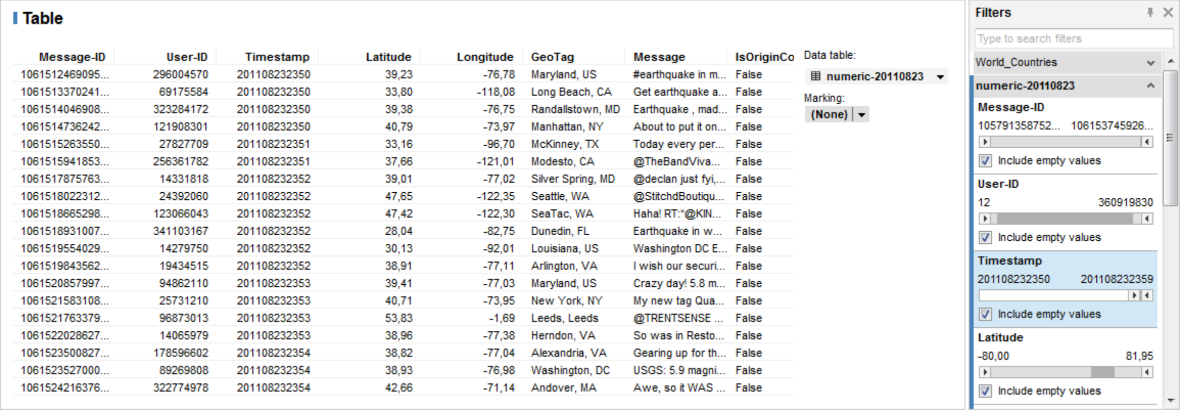
2. Use a second set of data, which contains the outline(s) of the relevant area(s), draw the area accordingly and plot the messages in the appropriate locations.

The second option has the advantage of being able to zoom and pan freely and select natural geographical areas.

The evaluated tools offer many different and configurable visualisations. Due to time constraints, we chose not to focus on every single one of them, but instead focused on those that appeared most relevant to disaster management. These consist of "scatter plot", "heat map" and "map chart". For completeness' sake, we also made sure there was a proper tabular visualisation, which could be used to browse the internal data set and check for import problems.

Spotfire

The data table (Figure 3.18) is trivial to create and use in Spotfire. It was also fast compared to the other visualisations, even when displaying the complete data set.



Table

Message-ID	User-ID	Timestamp	Latitude	Longitude	GeoTag	Message	IsOriginCo
1061512469095...	296004570	201108232350	39,23	-76,78	Maryland, US	#earthquake in m...	False
1061513370241...	69175584	201108232350	33,80	-118,08	Long Beach, CA	Get earthquake a...	False
1061514046908...	323284172	201108232350	39,38	-76,75	Randallstown, MD	Earthquake , mad...	False
1061514736242...	121908301	201108232350	40,79	-73,97	Manhattan, NY	About to put it on...	False
1061515263550...	27827709	201108232351	33,16	-96,70	McKinney, TX	Today every per...	False
1061515941853...	256361782	201108232351	37,66	-121,01	Modesto, CA	@TheBandViva...	False
1061517875763...	14331818	201108232352	39,01	-77,02	Silver Spring, MD	@declan just fyi,...	False
1061518022312...	24392060	201108232352	47,65	-122,35	Seattle, WA	@StitchdBoutiqu...	False
1061518665298...	123066043	201108232352	47,42	-122,30	SeaTac, WA	Haha! RT:"@KIN...	False
1061518931007...	341103167	201108232352	28,04	-82,75	Dunedin, FL	Earthquake in w...	False
1061519554029...	14279750	201108232352	30,13	-92,01	Louisiana, US	Washington DC E...	False
1061519843562...	19434515	201108232352	38,91	-77,11	Arlington, VA	I wish our securi...	False
1061520857997...	94862110	201108232353	39,41	-77,03	Maryland, US	Crazy day! 5.8 m...	False
1061521583108...	25731210	201108232353	40,71	-73,95	New York, NY	My new tag Qua...	False
1061521763379...	96873013	201108232353	53,83	-1,69	Leeds, Leeds	@TRENTSENSE ...	False
1061522028627...	14065979	201108232353	38,96	-77,38	Herndon, VA	So was in Resto...	False
1061523500827...	178596602	201108232354	38,82	-77,04	Alexandria, VA	Gearing up for th...	False
1061523527000...	89269808	201108232354	38,93	-76,98	Washington, DC	USGS: 5.9 magni...	False
1061524216376...	322774978	201108232354	42,66	-71,14	Andover, MA	Awe, so it WAS ...	False

Data table: numeric-20110823

Marking: (None)

Filters

Type to search filters

World_Countries

numeric-20110823

Message-ID: 105791358752... 106153745926...

☑ Include empty values

User-ID: 12 360919830

☑ Include empty values

Timestamp: 201108232350 201108232359

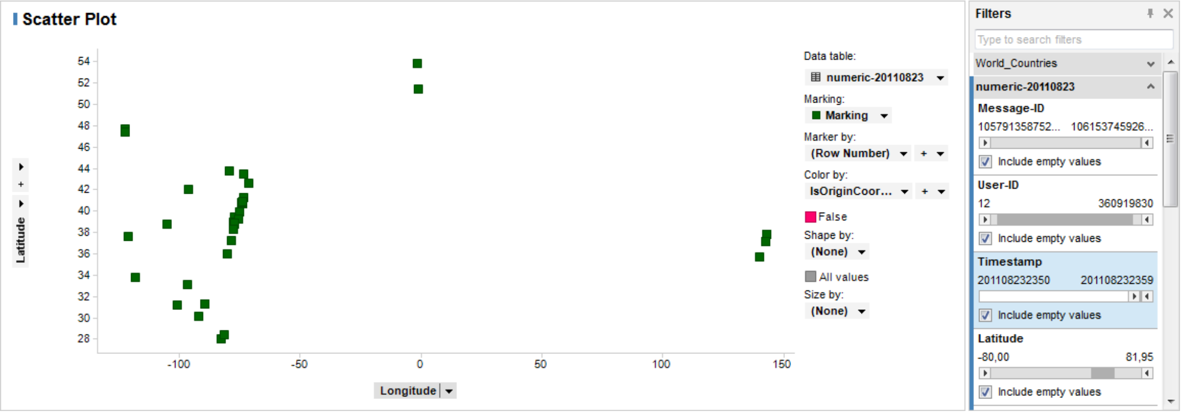
☑ Include empty values

Latitude: -80,00 81,95

☑ Include empty values

Figure 3.18: Table visualisation rendered by Spotfire.

A scatter plot of latitude by longitude like Figure 3.19 allows for a quick overview of the geographical distribution of the data, without having to spend time setting up a map chart. This permits a cursory analysis of the data, as clusters like those in Figure 3.27 will still stand out. It has the disadvantage of needing a very complete knowledge of the depicted area, as otherwise, one can just tell whether something is happening, not where. This visualisation is rather slow, with load times in the minutes for data sets like ours. Filtering the data to more manageable sizes like tens of thousands of messages makes this visualisation much more responsive.



Data table: numeric-20110823

Marking: 🟩 Marking

Marker by: (Row Number)

Color by: IsOriginCoor...

🟥 False

Shape by: (None)

▨ All values

Size by: (None)

World_Countries

numeric-20110823

Message-ID: 105791358752... 106153745926...

☑ Include empty values

User-ID: 12 360919830

☑ Include empty values

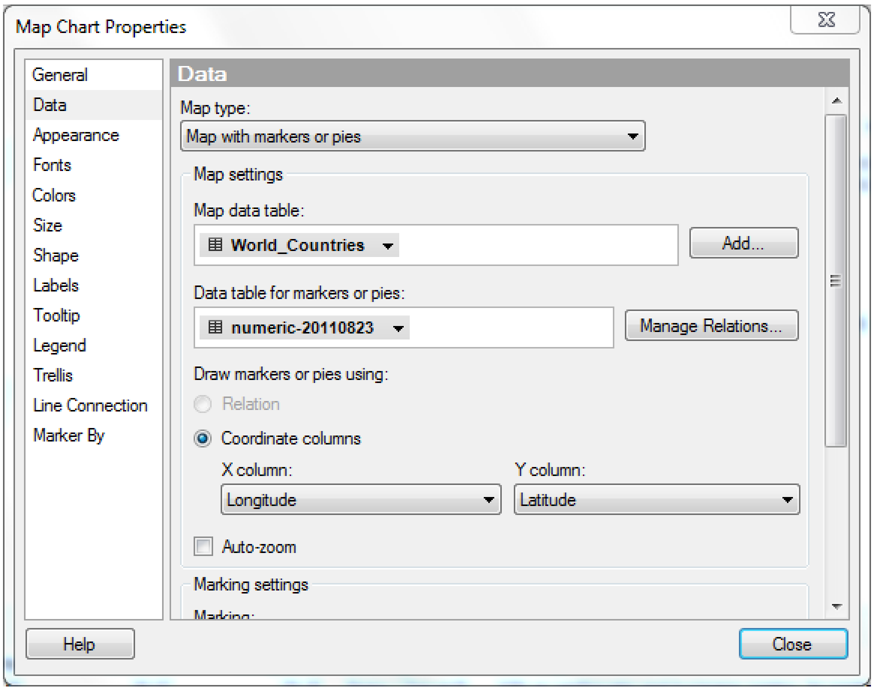
Timestamp: 201108232350 201108232359

☑ Include empty values

Latitude: -80,00 81,95

☑ Include empty values

Figure 3.19: Spotfire's scatter plot of earthquake data - not very useful without a map.



Map type: Map with markers or pies

Map settings

Map data table: World_Countries

Data table for markers or pies: numeric-20110823

Draw markers or pies using:

○ Relation

● Coordinate columns

X column: Longitude

Y column: Latitude

□ Auto-zoom

Figure 3.20: Configuring a map chart in Spotfire.

The best visualisation for finding events and determining their geographical location and extent is the map chart (Figure 3.1). Since Spotfire supports both options of realising a map chart, we decided on using the second one. Figure 3.20 demonstrates how to set up this kind of map chart in Spotfire. The map data table “World_Countries” was imported from the “World (Countries)” map chart example that was distributed as part of the default Spotfire installation.

The map chart has an option to make the size of the markers relative to the amount of data at the same geographical coordinates. For our data set that meant that there was a large marker at the geographical centre of the map. This is due to the relatively large amount of people who have set their coordinates to 0. This kind of geographical information was useless, so we removed it from the map chart.

Spotfire can also colour the markers on the map according to the value of a freely definable column. This feature can be used to visualise the timeline of an event, especially its spread. Markers can, however, obscure each other. This is problematic in disaster events, where new messages are constantly published in the affected area and older and newer messages obscure each other, contaminating the timeline.

This feature can be used to simulate a heat map by setting the colour of the markers according to the amount of data at the same geographical coordinates. This suffers from the same problem as the simulated timeline: since the markers obscure each other, a “hot” area can be obscured by many “cold” markers, and vice versa.

To make the displayed information more useful, Spotfire has the option of individually configuring the markers’ thumbnails. To demonstrate this, we added the following information to the thumbnails, as seen in Figure 3.1 (b):

- The amount of messages represented by each marker, represented by the “(Row Count)” line.
- The latitude and longitude for precise geographical location information.
- The “GeoTag” provided by the Twitter data.
- The actual message.

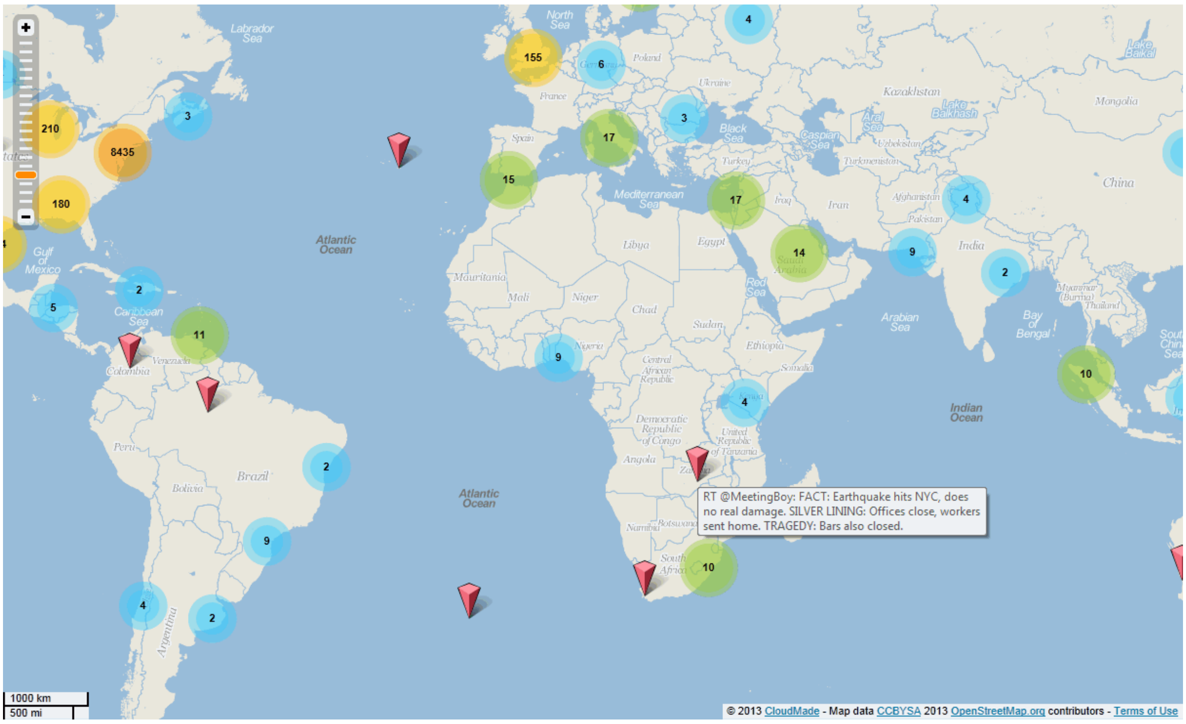
Exploring the map in detail is difficult, since the only way to manipulate the view is to use the controls between the chart and the legend, as shown in Figure 3.1 (a). Here, direct mouse interaction with the map instead of just the buttons, like the one offered by Google Maps [5], would improve the usability.

We hoped that the visualisation offered by Spotfire as “heat map” would be a map representation of the data, in which individual areas are coloured according to a parameter. One such parameter might be the amount of messages in each area. It turned out to be a two-dimensional chart representing matrix information as described in Wikipedia’s “Heat map” article [13]. Since we could not see a disaster management application for this chart, we did not explore it further.

QlikView

To show the tweets and their location on a map, QlikView can render a scatter plot using the latitude and longitude as coordinates and the message text as value. However, this is quite slow. Performance is better when something different, like the message ID, is selected as value and the text is only shown as a “popup” (QlikView’s name for tooltips). The locations of the messages formed an outline of the continents when the amount of messages was high enough. To add a map underneath the scatter plot, a static or dynamic image can be selected. The use of a static image is sufficient if the user is only interested in a certain geographical region and does not need to zoom dynamically.

However, for this study, a more flexible visualisation on a map was needed. Therefore the extension “CloudMade Web Maps” [9] (CWM) was installed. It allows to visualise the messages on a map that can be zoomed and panned, while selecting sensible default viewports (Figure 3.21). A very good feature of this extension is the automatic clustering that combines nearby messages into circles. The amount of messages in the clusters is then used to select a size and colour for each circle. Unfortunately, there is one caveat to using extensions in general: QlikView limits the number of rendered data records to 10,000.



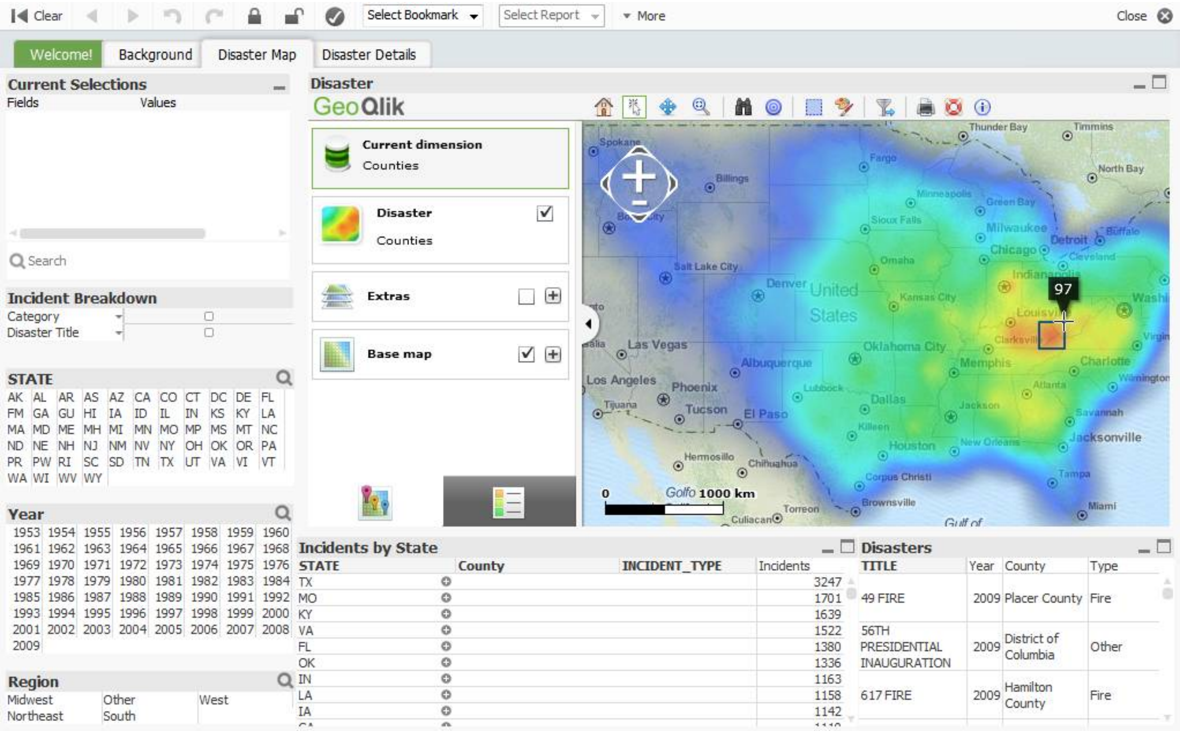
RT @MeetingBoy: FACT: Earthquake hits NYC, does no real damage. SILVER LINING: Offices close, workers sent home. TRAGEDY: Bars also closed.

Figure 3.21: A map generated with QlikView and the CloudMade Web Maps extension.

The “Another Google Maps” [8] (AGM) extension also allows showing data points on a map. Like CWM, it only loads up to 10,000 messages and clusters them. Unlike CWM, it does mandatory clustering: the user can not opt out. While CWM automatically centres the map to fit all data points, AGM reverts to a default start position and zoom level every time the selection changes. This problem makes AGM unusable.

There is no built in way of creating heat maps in QlikView. However, the background colour of a graph's dimension can be set by an expression that returns a colour. This allows the user to implement a quasi heat map with custom preferences and parameters.

There also is an extension that has been developed for geographical information systems: GeoQlik [4]. It shows the data on a map and can also overlay it with heat maps (Figure 3.22). However, this extension is not available in an unlimited, free version and did not work properly in our tests. While it worked flawlessly on the developer's demo website [1], it suffered from reproducible freezes on the reference machine. The freezes always occurred while the extension object was being configured and we therefore never saw a rendering of our data.



Incidents by State

STATE	County	INCIDENT_TYPE	Incidents
TX	⊕		3247
MO	⊕		1701
KY	⊕		1639
VA	⊕		1522
FL	⊕		1380
OK	⊕		1336
IN	⊕		1163
LA	⊕		1158
IA	⊕		1142

Disasters

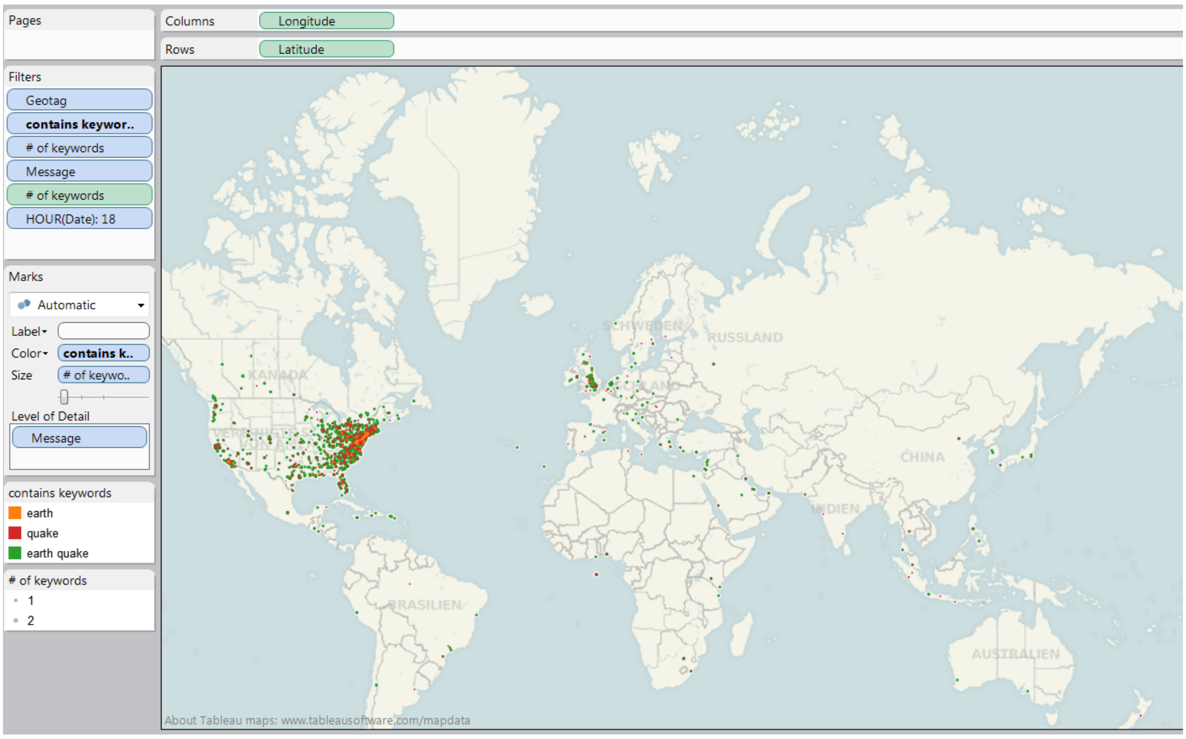
TITLE	Year	County	Type
49 FIRE	2009	Placer County	Fire
56TH PRESIDENTIAL INAUGURATION	2009	District of Columbia	Other
617 FIRE	2009	Hamilton County	Fire

Figure 3.22: A heat map of fire-related disasters rendered with GeoQlik inside a web browser.

Tableau

Tableau supports visualisations of geographical data natively. Fields can be assigned geographic roles, not only longitude and latitude but other regional location indicating values as well, for example country/region, state/province, state, area code, etc. With the appropriate geocoding data, the software can present a value at the proper position on a map based on only this simple geographical information. The map data (e.g. graphics) are stored on the web.

Figure 3.23 shows the Twitter messages that contain either the keyword “earth” (orange), “quake” (red) or both (green) between 18:00 and 19:00 UTC from our sample data.



Pages

Columns: Longitude

Rows: Latitude

Filters
- Geotag
- contains keywor..
- # of keywords
- Message
- # of keywords
- HOUR(Date): 18

Marks
- Automatic
- Label
- Color: contains k..
- Size: # of keywo..
- Level of Detail: Message

contains keywords
- 🟧 earth
- 🟥 quake
- 🟩 earth quake

# of keywords
- 1
- 2

Figure 3.23: Tableau geospatial visualisation.

Tableau’s “filled map” visualises data with geographical regional information by colour-coding regions on a map. This kind of visualisation is well-suited for inspecting, for instance, the density differences of Twitter messages that contains the keyword “earthquake” in each ZIP code zone of the USA.

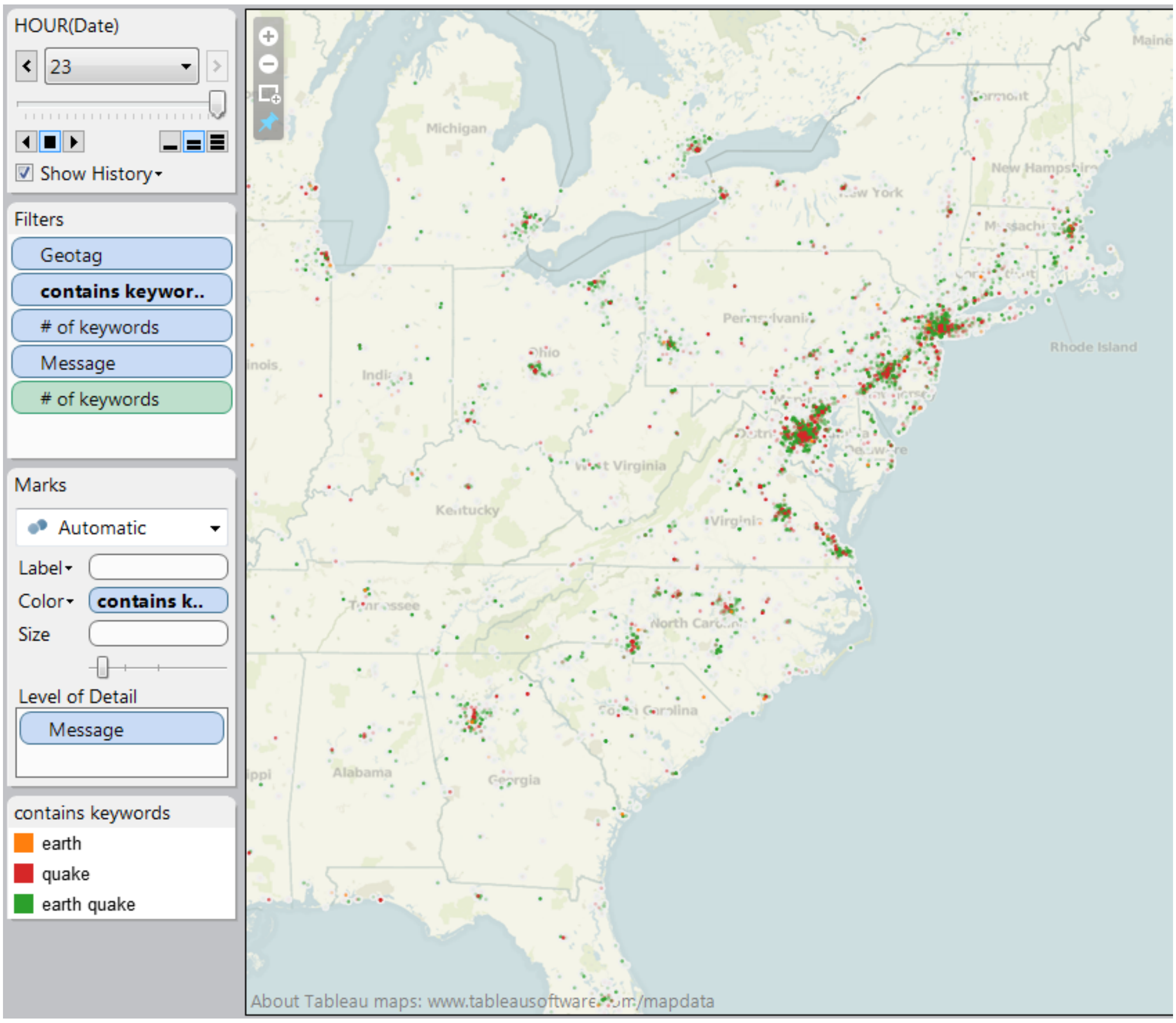
In the *Desktop* version, Tableau only has detailed geocoding data (e.g. ZIP code) for the USA and the United Kingdom. The data is limited and it is different for other countries, for instance the highest granularity of regional geocoding data provided for Germany is on the state level. However, Popper [24] has successfully imported and published the German “Postleitzahl” (ZIP code equivalent) data. The same technique can be used for other countries.

For the USA, there are also further integrated data layers (e.g. Population statistics etc.).

Tableau automatically chooses the type of visualisation based on the selected fields’ data types. The result is usually very appropriate. If it is to be changed, the “Show Me!” panel provides great advice on which kinds of data are appropriate for the desired type of visualisation.

Visualisations are easily customisable, from the shape, size, colour and labels of the representational icons, to which information is to be shown when the mouse hovers over it. Actions can be set to execute when an icon is clicked, for example to open a URL within the data record, activate a filter or highlight some records.

Other visualisation types are also helpful for our scenarios, for example trend lines or the mentioned “animation” might help detect abnormal amounts of messages over time (3.2.4 Tableau’s Suitability for Disaster Management).



HOUR(Date) 23

☑ Show History

Filters

Geotag

contains keywor..

# of keywords

Message

# of keywords

Marks

Automatic

Label

Color contains k..

Size

Level of Detail

Message

contains keywords

🟧 earth

🟥 quake

🟩 earth quake

Figure 3.24: Tableau’s paging feature.

Figure 3.24 shows the Twitter messages that contain either the keyword “earth” (orange), “quake” (red) or both (green) between 23:00 and 24:00 from our sample data on the east coast of the USA. Messages from the prior five hours are also shown, but are faded out gradually according to their age. The time period after which the visualisation changes can be set to short, normal or long, which controls the animation speed.

	Spotfire	QlikView	Tableau
Clustering	–	✓[a]	–
User relationships	–	–	–
Message relationships	–	–	–

Table 3.4: A comparison of data analysis capabilities.

[a] Functionality is provided by extensions.

3.3.4 Data Analysis

Some data relationships cannot be determined by simply using visualisation techniques. This is why the tools offer additional analysis functionality, which allows the user to extract further information from the data.

Spotfire

Spotfire supports advanced data analysis through the "Tools" menu. It has six data analysis tools, of which we tried all except for the *Line Similarity* and *Regression Modeling* tools, as the sample data seemed unsuitable:

Data Relationships This tool allows the determination of how closely different columns are related. It does not allow the user to easily visualise relationships between individual messages, as they are contained in a single column.

K-means Clustering This implementation of *K-means* needs a line chart to run. As this limitation is quite severe and not intuitive, we chose to ignore this analysis tool. It is rather difficult to create a useful line chart of messages or geographical coordinates, which means the most important use cases – clustering messages by content and geographical coordinates – are infeasible.

Line Similarity This analysis needs a line chart like *K-means Clustering*. Since there are no apparent disaster management applications for this type of analysis, we chose to not pursue it any further.

Hierarchical Clustering This clustering method allows Spotfire's "heat maps" to be clustered. Since this does not meet our expectations, we did not investigate this analysis tool any further.

Regression Modeling Regression analysis is a predictive method. Since we focused on disaster management applications, for which we needed primarily descriptive data mining methods and experts have superior predictive models for disaster spread, we did not look into this tool.

Classification Modeling This analysis tool allows the creation of classification trees, which are another predictive data mining method. As before, we chose not to further investigate this tool since it does not seem applicable to disaster management scenarios.

As it turns out, none of the available analysis tools have any immediately apparent application in disaster management. Coupled with the general lack of user friendliness of the tools' interfaces, we found Spotfire severely lacking in data analysis. This is definitely an area in which a little improvement to user-friendliness would make a remarkable difference.

QlikView

The program supports grouping data. The groups can be hierarchical or cyclic. However, there is no built-in functionality to cluster data based on textual content and geographical distance. As mentioned above, the CWM extension could combine nearby data into clusters. Unfortunately, unlike single data points, the clusters have no details on demand in their tooltips.

Since QlikView is a general visualisation tool, it only knows semantics in the way a database does: data types and relationships between columns. Therefore no further analysis of the quality or importance of the tweets was possible with native functionality. The user has to implement it himself, if necessary.

Tableau

Tableau also supports grouping data. Applying this feature to the tasks of our study, it could be used, for example, to combine multiple keywords that have identical or similar meaning into a category and examine as a whole (the "Earthquake" category has "earth quake", "earthquake"; the *Consequence* category could contain "power plant", "road block", etc).

Tableau doesn't have any semantic analysis tools.

The detection of relationships between messages (answering messages, repostings of messages or so-called "retweets") is possible. To answer a message, people usually have to include the recipient's username in the message, prefixing by an "@" sign. Thanks to this specific format, a calculated field can be created to extract the username. A separate table containing information about user ID and username must be used to detect which message a user is answering to and therefore create a link between the two messages. The same technique could be used to detect "retweets" and eliminate them to improve data quality.

Tableau is able to visualise such "trails" between records over time. A sample workbook included with the Tableau Desktop installation offers this visualisation of a storm's progresses: Figure 3.25 shows the storm's tracks for the for Atlantic basin. The colour indicates the category of the storm. The lines' thickness indicates the total energy of the storm.

Unfortunately, the visualisation of this kind would look very complicated for a large amount of data and in our case, the sample data is insufficient for the task. For those reasons, we decided to continue to evaluate the tool in other aspects.

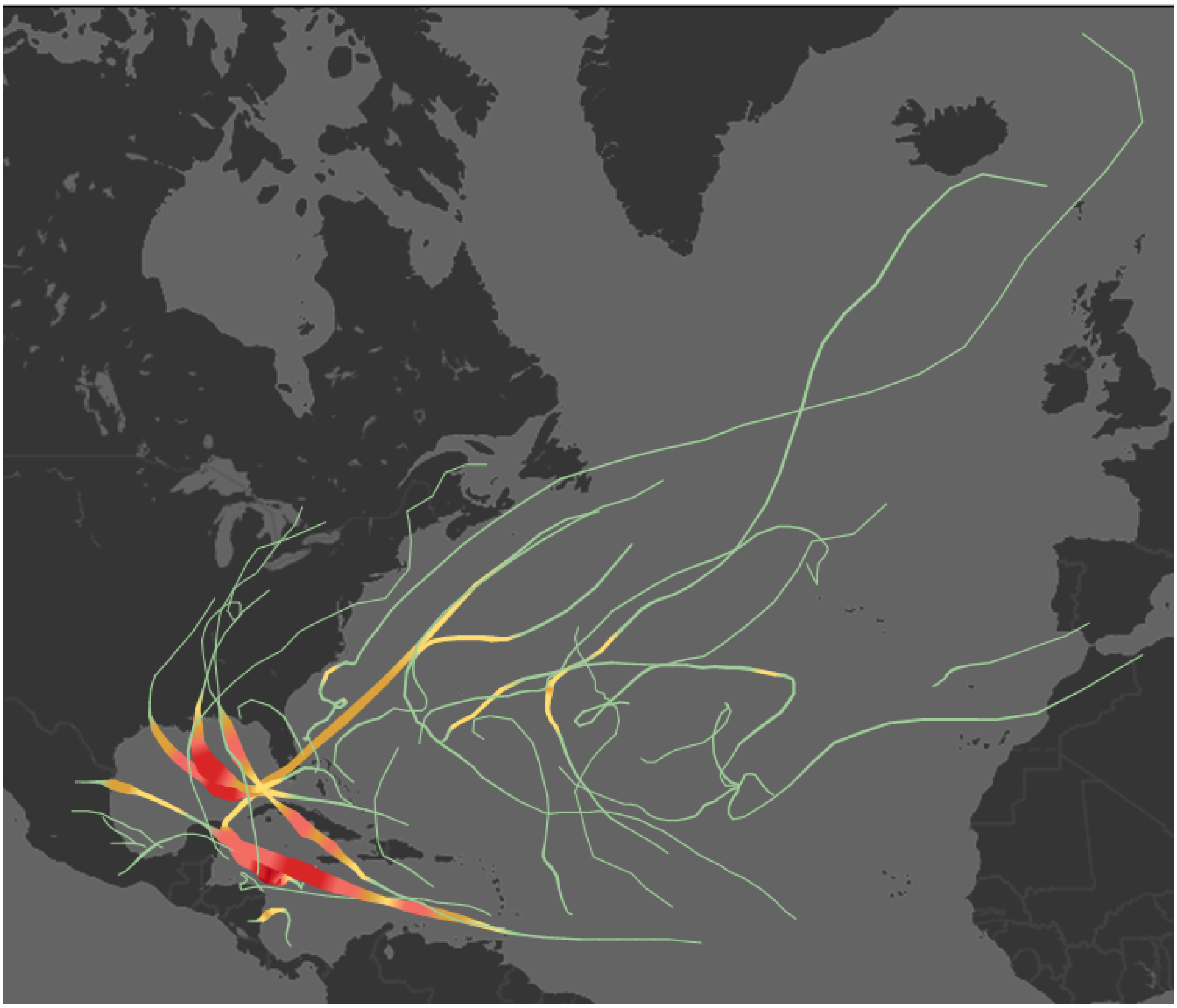


Figure 3.25: Storm tracks through the Atlantic basin.

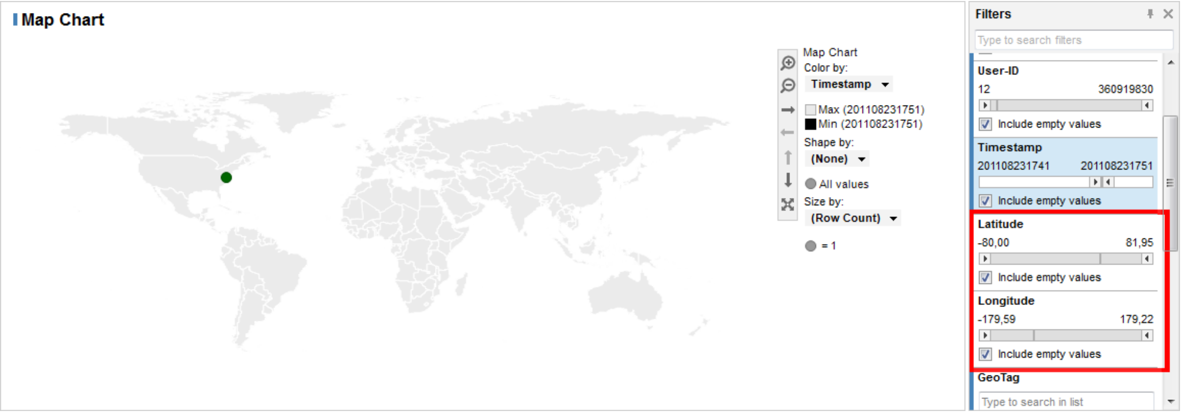
3.3.5 Exploring the Earthquake

As mentioned above (3.2.1 Tool Selection), our sample data set contains messages regarding an earthquake. This disaster serves as a benchmark to help us evaluate how easy and intuitive the disaster discovery process is with a specific tool.

Spotfire

To find the earthquake, all we had to do was to use the *Text Filter* to search for the term "*quake". This concentrates the search on the English speaking regions of the world, which isn't problematic in our scenario, since the disaster mainly impacted Americans. Using this method had the disadvantage that misspellings and related messages not containing the search term were ignored. Since the result set was still sufficient to comfortably explore the earthquake given Spotfire's limitations, we decided to proceed.

After a lengthy initial rendering, we used the *Range Filter* on the "Timestamp" column to display a time span of several hours. We moved this window across the time axis until we found a high concentration of messages posted in North America. To get a step-by-step visualisation of the spread of the earthquake-related messages, we first narrowed our *Range Filter* down to ten minutes. Secondly, we moved the time range so that the first earthquake-related message was visible, as shown in Figure 3.26. As Spotfire was able to render this amount of data in real-time, we then stepped through the data, shifting the time window minute by minute. Figures 3.27 through 3.33 illustrate the spread of messages over the whole world.

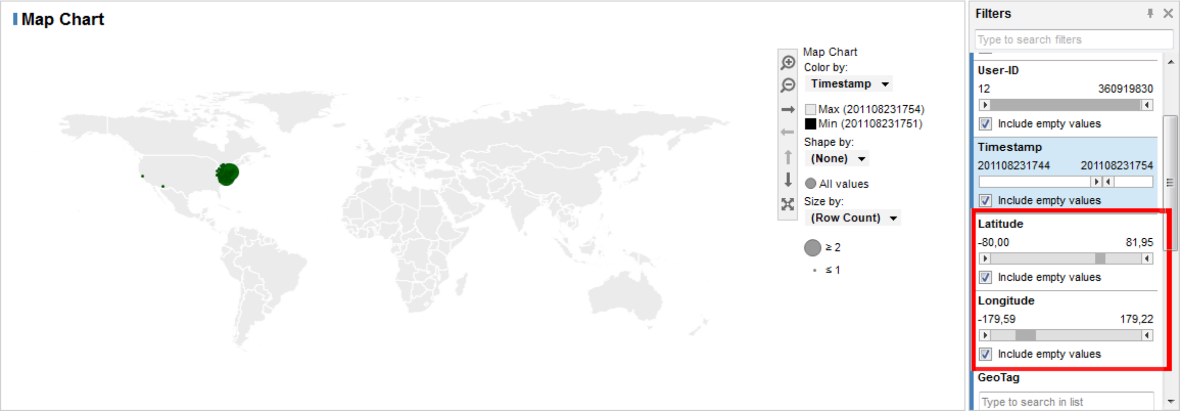


Map Chart

Map Chart
Color by:
Timestamp
□ Max (201108231751)
■ Min (201108231751)
Shape by:
(None)
● All values
Size by:
(Row Count)
● = 1

Filters
Type to search filters
User-ID
12 360919830
☑ Include empty values
Timestamp
201108231741 201108231751
☑ Include empty values
Latitude
-80,00 81,95
☑ Include empty values
Longitude
-179,59 179,22
☑ Include empty values
GeoTag
Type to search in list

Figure 3.26: Spotfire's map of tweets about an earthquake (17:41 - 17:51 UTC)



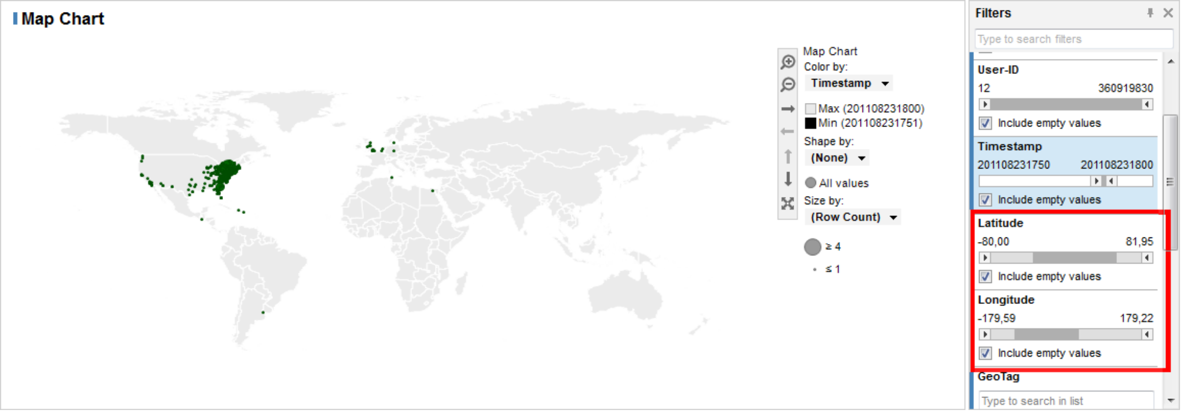
Map Chart

Map Chart
Color by:
Timestamp
□ Max (201108231754)
■ Min (201108231751)
Shape by:
(None)
● All values
Size by:
(Row Count)
● ≥ 2
• ≤ 1

Filters
Type to search filters
User-ID
12 360919830
☑ Include empty values
Timestamp
201108231744 201108231754
☑ Include empty values
Latitude
-80,00 81,95
☑ Include empty values
Longitude
-179,59 179,22
☑ Include empty values
GeoTag
Type to search in list

Figure 3.27: Spotfire's map of tweets about an earthquake (17:44 - 17:54 UTC)

One useful Spotfire feature is apparent in these figures: the highlighting of the currently visible data range. Figures 3.26, 3.26 and 3.28 demonstrate this in the "Latitude" and "Longitude" columns. The light grey area represents the data that is included by the particular filter, but



Map Chart

Map Chart
Color by: Timestamp
□ Max (201108231800)
■ Min (201108231751)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 4
• ≤ 1

Filters

User-ID: 12 – 360919830
☑ Include empty values

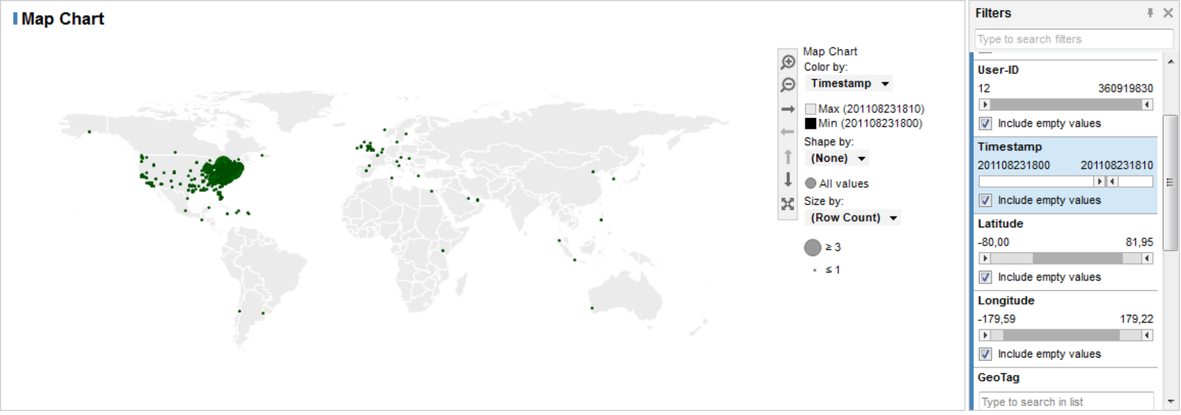
Timestamp: 201108231750 – 201108231800
☑ Include empty values

Latitude: -80,00 – 81,95
☑ Include empty values

Longitude: -179,59 – 179,22
☑ Include empty values

GeoTag

Figure 3.28: Spotfire's map of tweets about an earthquake (17:50 - 18:00 UTC)



Map Chart

Map Chart
Color by: Timestamp
□ Max (201108231810)
■ Min (201108231800)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 3
• ≤ 1

Filters

User-ID: 12 – 360919830
☑ Include empty values

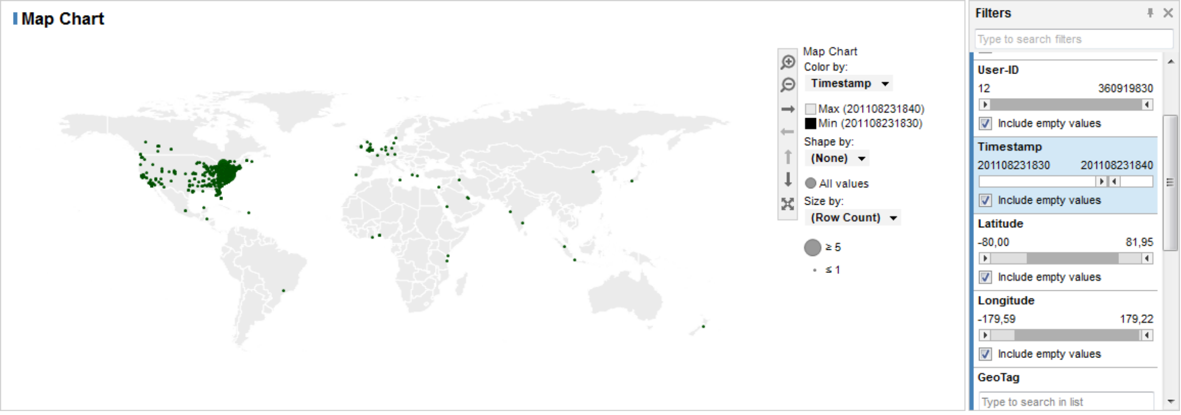
Timestamp: 201108231800 – 201108231810
☑ Include empty values

Latitude: -80,00 – 81,95
☑ Include empty values

Longitude: -179,59 – 179,22
☑ Include empty values

GeoTag

Figure 3.29: Spotfire's map of tweets about an earthquake (18:00 - 18:10 UTC)



Map Chart

Map Chart
Color by: Timestamp
□ Max (201108231840)
■ Min (201108231830)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 5
• ≤ 1

Filters

User-ID: 12 – 360919830
☑ Include empty values

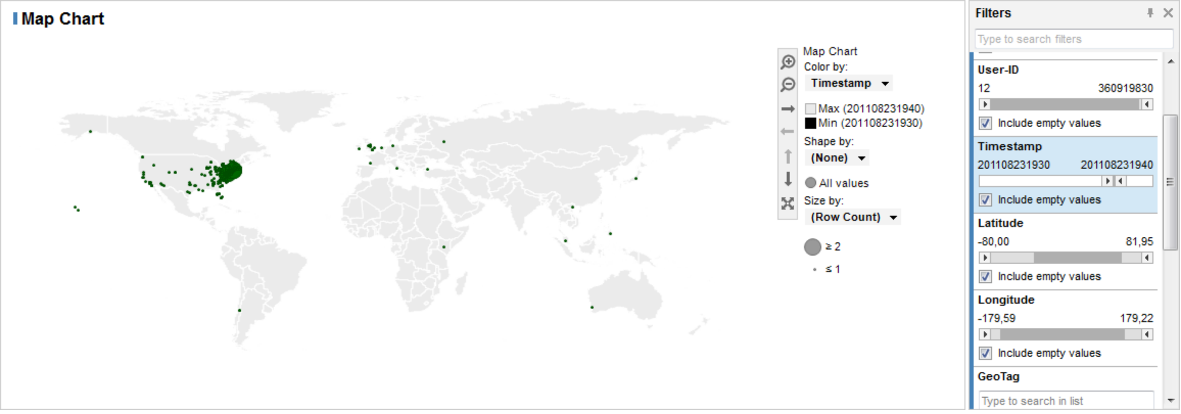
Timestamp: 201108231830 – 201108231840
☑ Include empty values

Latitude: -80,00 – 81,95
☑ Include empty values

Longitude: -179,59 – 179,22
☑ Include empty values

GeoTag

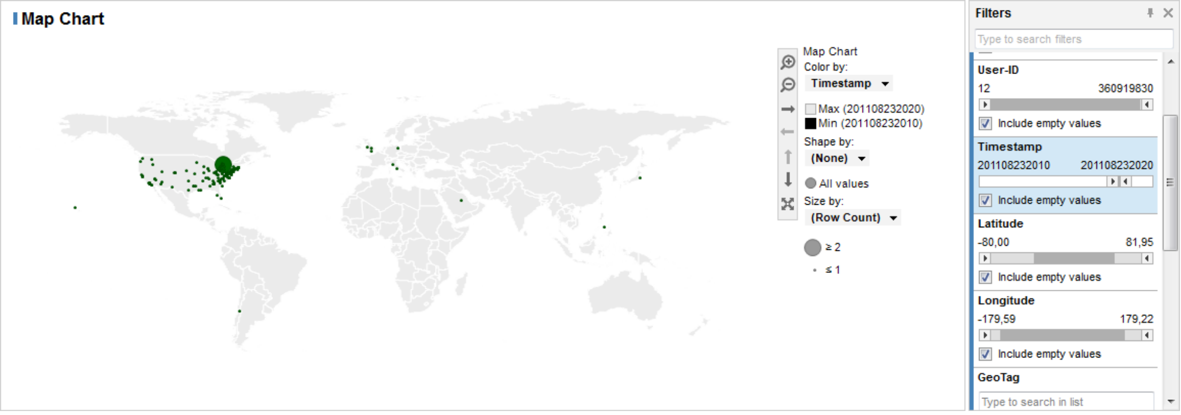
Figure 3.30: Spotfire's map of tweets about an earthquake (18:30 - 18:40 UTC)


Map Chart

Map Chart
Color by: Timestamp
□ Max (201108231940)
■ Min (201108231930)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 2
• ≤ 1

Filters
Type to search filters
User-ID: 12 – 360919830
☑ Include empty values
Timestamp: 201108231930 – 201108231940
☑ Include empty values
Latitude: -80,00 – 81,95
☑ Include empty values
Longitude: -179,59 – 179,22
☑ Include empty values
GeoTag
Type to search in list

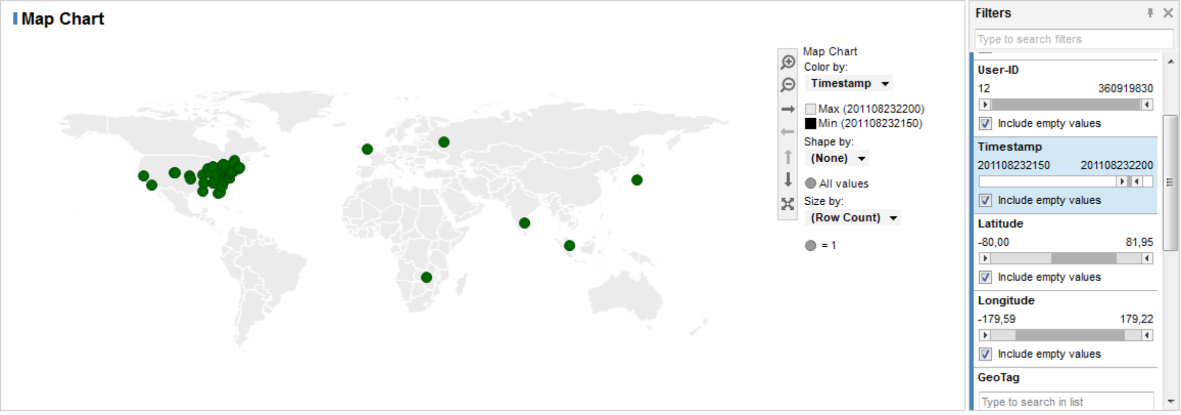
Figure 3.31: Spotfire's map of tweets about an earthquake (19:30 - 19:40 UTC)


Map Chart

Map Chart
Color by: Timestamp
□ Max (201108232020)
■ Min (201108232010)
Shape by: (None)
● All values
Size by: (Row Count)
● ≥ 2
• ≤ 1

Filters
Type to search filters
User-ID: 12 – 360919830
☑ Include empty values
Timestamp: 201108232010 – 201108232020
☑ Include empty values
Latitude: -80,00 – 81,95
☑ Include empty values
Longitude: -179,59 – 179,22
☑ Include empty values
GeoTag
Type to search in list

Figure 3.32: Spotfire's map of tweets about an earthquake (20:10 - 20:20 UTC)


Map Chart

Map Chart
Color by: Timestamp
□ Max (201108232200)
■ Min (201108232150)
Shape by: (None)
● All values
Size by: (Row Count)
● = 1

Filters
Type to search filters
User-ID: 12 – 360919830
☑ Include empty values
Timestamp: 201108232150 – 201108232200
☑ Include empty values
Latitude: -80,00 – 81,95
☑ Include empty values
Longitude: -179,59 – 179,22
☑ Include empty values
GeoTag
Type to search in list

Figure 3.33: Spotfire's map of tweets about an earthquake (21:50 - 22:00 UTC)

excluded by another one. The dark grey area represents the currently visible data and white areas are excluded by the filter.

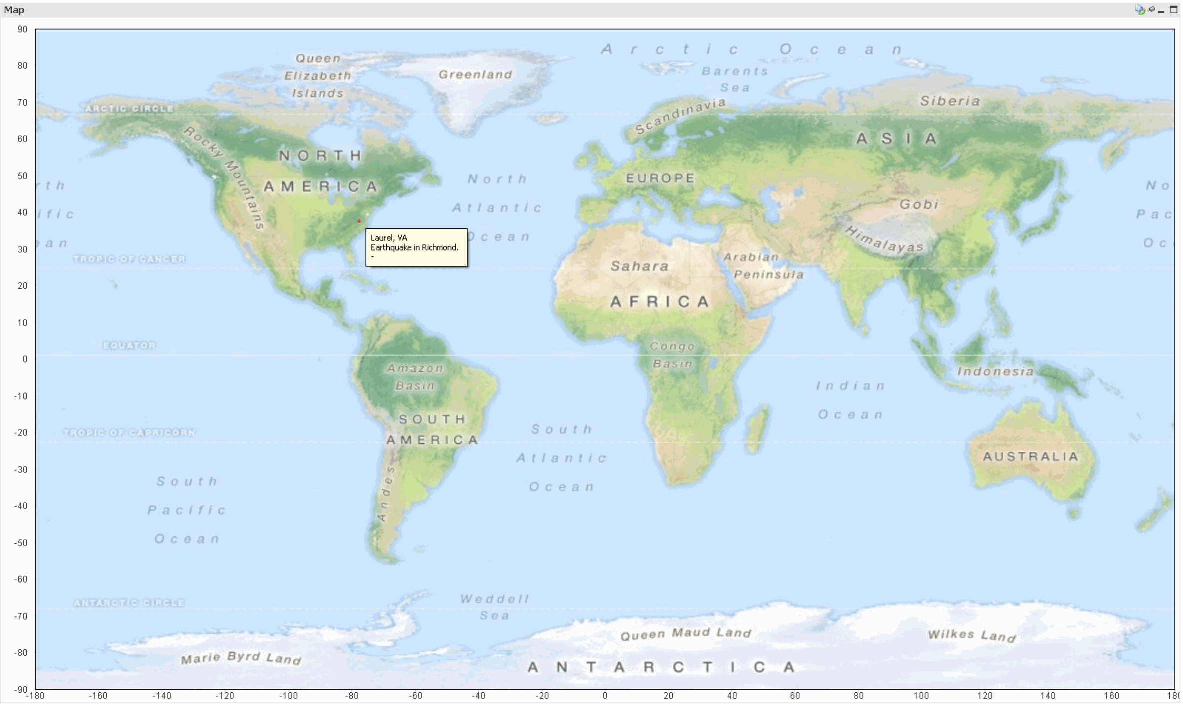
Figure 3.27 illustrates how many Twitter users reacted to the event within the first four minutes. This obvious spike in the use of keywords ending in “quake” demonstrates how event detection in Twitter [17, 30, 27, 18] works. After 22:00, the spread and amount of new messages posted stayed relatively constant to the end of the data set’s time frame. This illustrates how Twitter users kept discussing the event even after its end. Since the completion of events does not coincide with the end of their discussion, social networks are useful for event discovery and the observation of their development. They cannot be used to discover the time an event ends.

QlikView

QlikView allowed exploring the earthquake, too. After the messages were constrained to those including the text “quake”, a range filter was used on the timestamp to try and find the first record. For this the range was repeatedly decreased from the high and the low end. This allowed for the discovery of an approximate time at which users started to write massively about an earthquake. The search was then continued by selecting ever smaller ranges and finally resulted in a single record at 17:51 (Figure 3.34).

After the start time of the event was identified, we proceeded with the analysis of the spread of the news by selecting a time range of several minutes and moving it forward (Figure 3.35 - 3.38).

After finding the first message and viewing the spread of the news on the map, we created a stacked mountain chart to see the progression over time (Figure 3.39).



Laurel, VA
Earthquake in Richmond.

Figure 3.34: QlikView showing the map at the beginning of the earthquake (17:51).

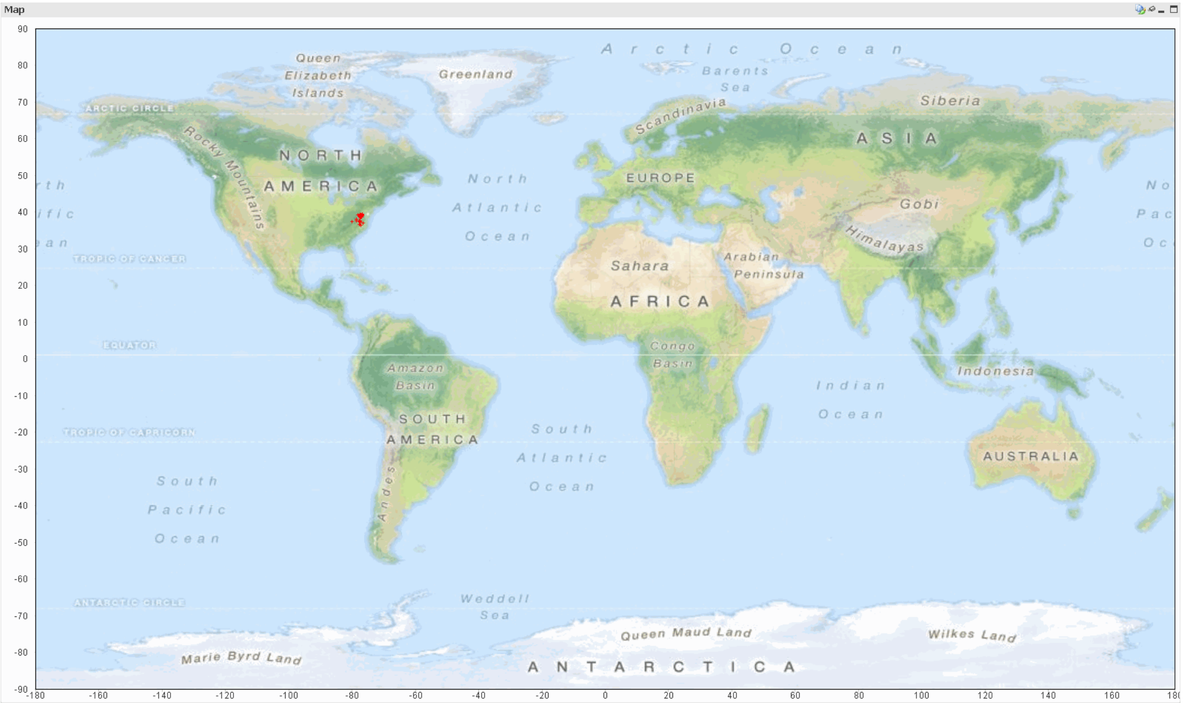


Figure 3.35: QlikView's map with Twitter messages of the earthquake (17:52).

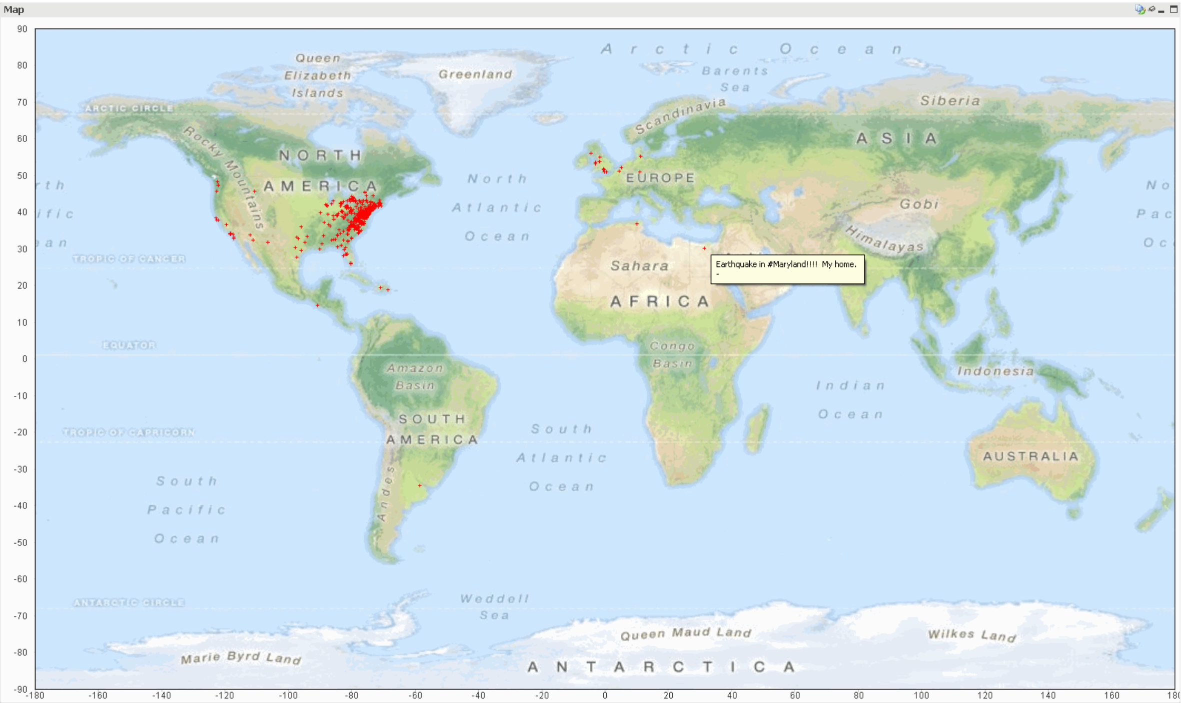

Earthquake in #Maryland!!!! My home.

Figure 3.36: QlikView's map with Twitter messages of the earthquake (17:53 - 18:00).

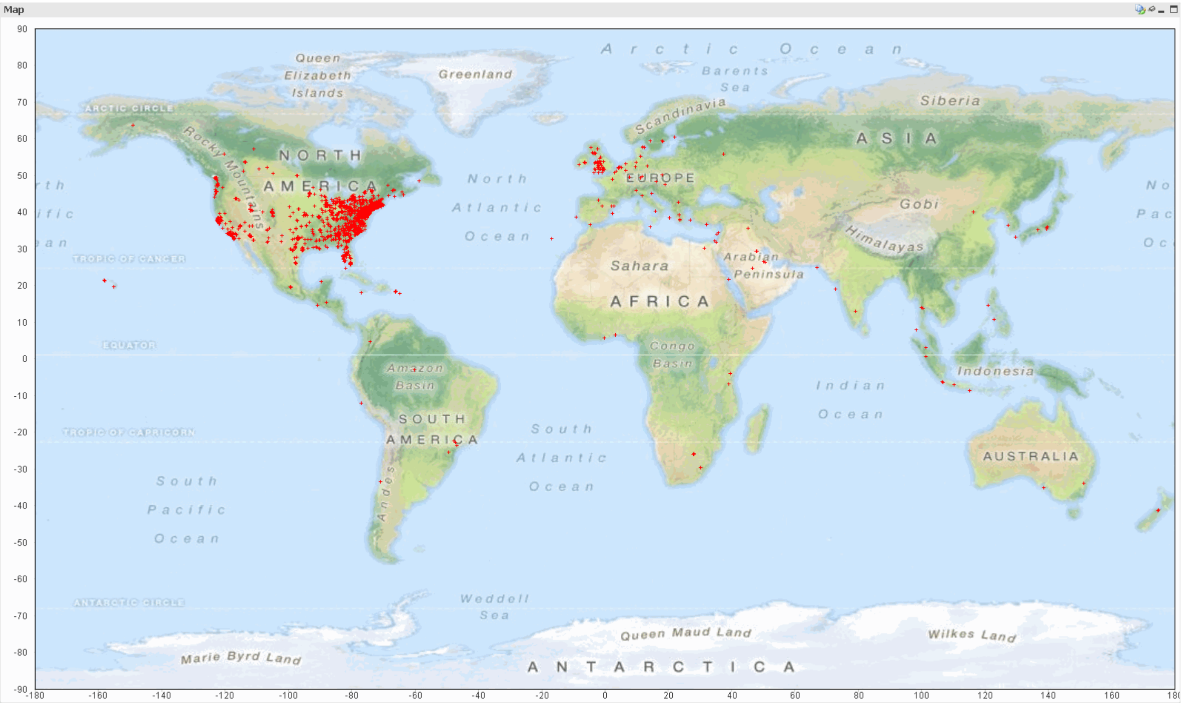


Figure 3.37: QlikView's map with Twitter messages of the earthquake (18:16 - 19:00).

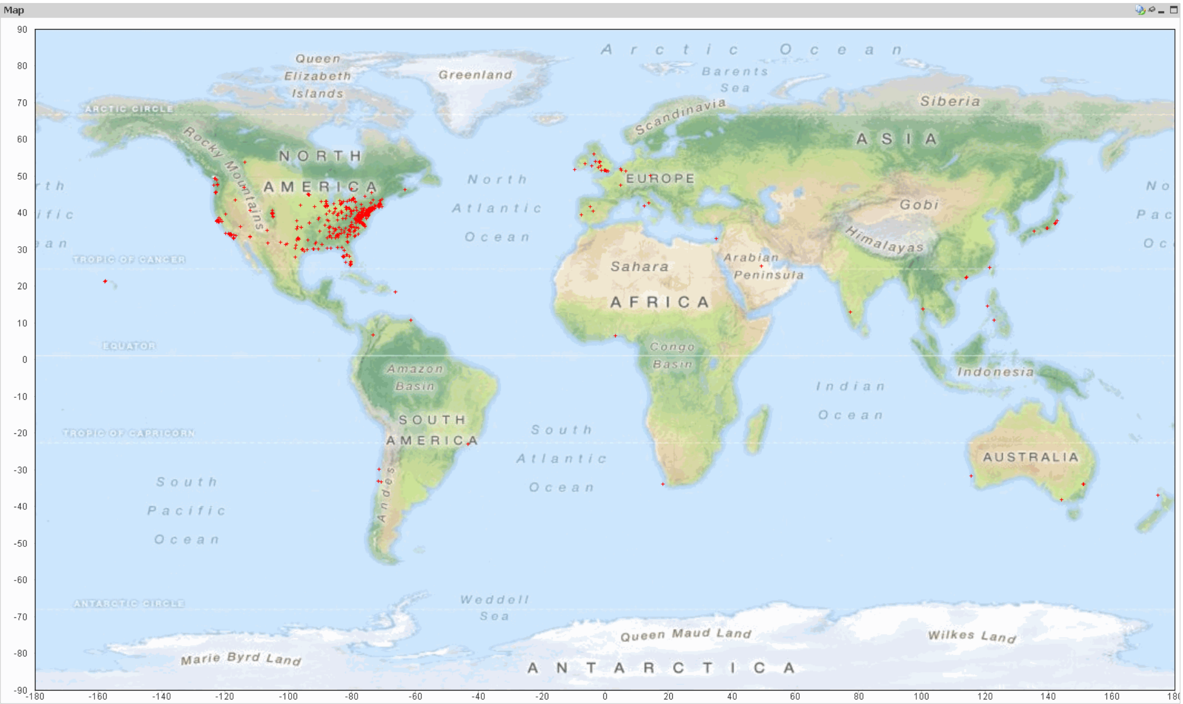


Figure 3.38: QlikView's map with Twitter messages of the earthquake (22:01 - 23:59).

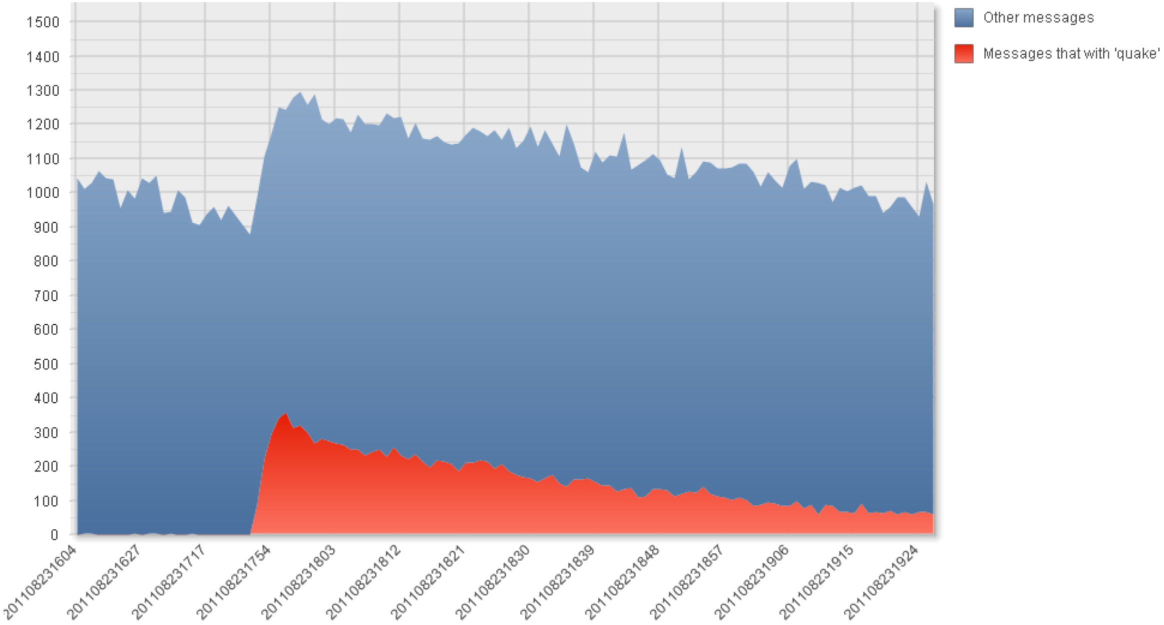


Figure 3.39: QlikView's stacked mountain chart (red - earthquake, blue - other messages).

Tableau

We were able to explore the earthquake data with Tableau in the same way as with the other applications. The tooltip provides quick information about an individual data record. Multiple graphical representations of records can be selected by mouse and their underlying data can be shown, either summarised or in their original form.

Theoretically, if a parameter field could be created containing every important keyword in a message (which are, in case of Twitter, called "hashtags" and usually prefixed by a "#" sign), it could be use as the label for a data point. This would be a great visual assistance to possibly discover other important information related to the crisis. This technique is undoubtedly not suitable for a large region or regions with high message density.

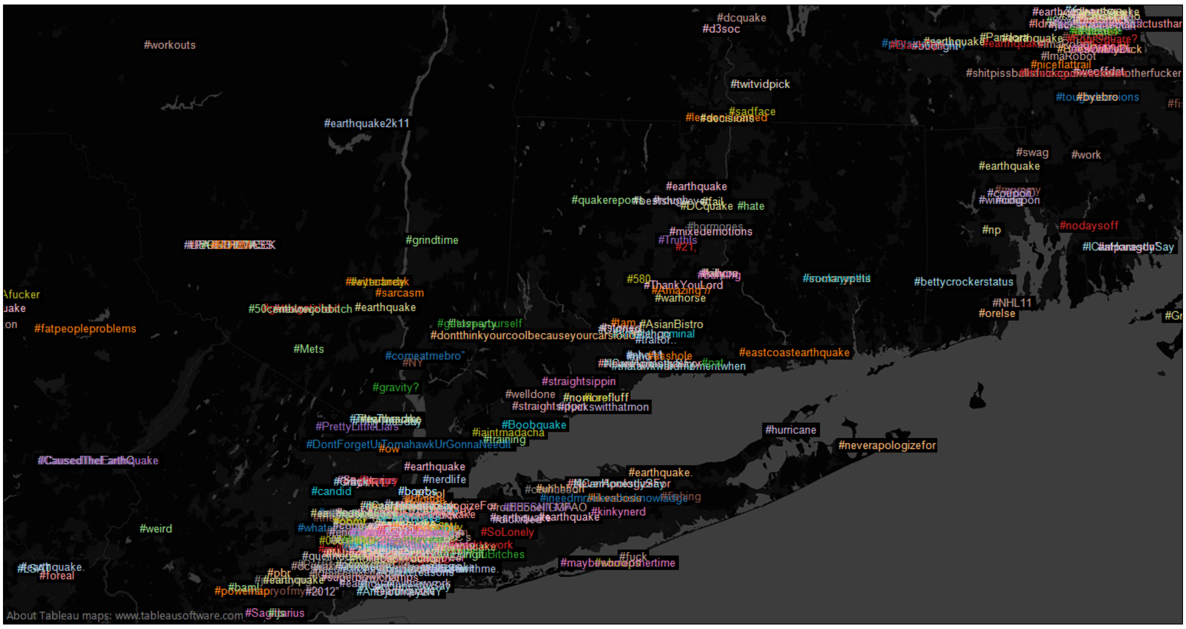


Figure 3.40: Using hashtag as labels (first hashtag only) - USA's east coast between 20:00 and 21:00.

Another theory on the topic of hashtag discovery is formed at the end of our study. Using a calculated field to extract the first and usually the most important hashtag, a visualisation could be created to show how often a hashtag is used in a specific time period. As a result, anomalies can be identified if they are popular enough. This method has one upside: if an important event has a significantly smaller amount of messages than other topics, it could possibly be located.

Even though creating additional calculated fields for the secondary hashtags is achievable, the attempt to extract all hashtags in a message is, on the other hand, another problem.

Since the *COUNT* function of SQL and Tableau only accepts one field as data source, if a keyword exists in two hashtag columns (calculated fields), the function cannot register its appearance correctly. The problem has to be approached from another angle.

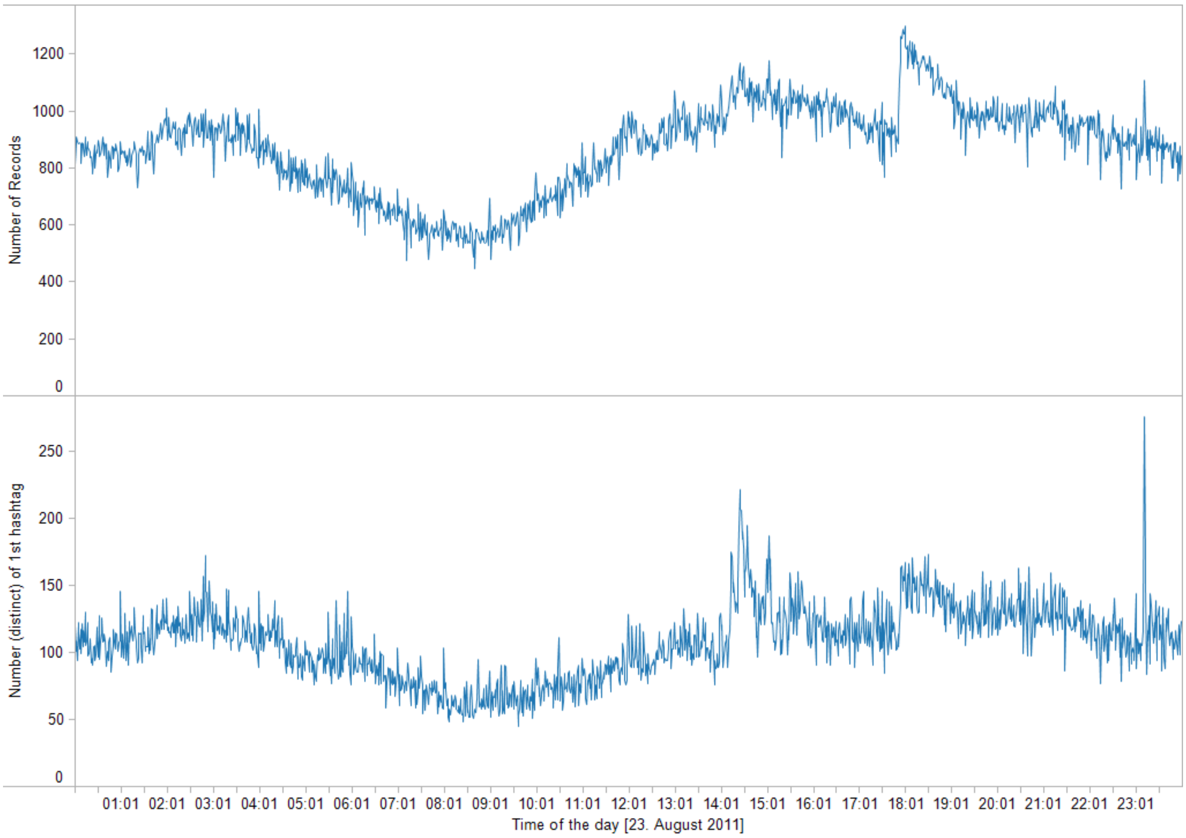


Figure 3.41: Number of messages and distinct hashtags over the hours (first hashtag only).

Obviously, another table containing only the hashtags in one single column is desired. This is when the data import and export features of Tableau shine. The table with two hashtag columns, to choose a simplistic example, can be exported to a data file (Tableau supports exporting to Microsoft Access or a CSV file through the clipboard). A new table can then be created by importing the exported data, but initially only the first hashtag column should be imported. The same data is imported again to that newly created table, this time with only the second hashtag column. After the table with all the hashtags' appearances is constructed, visualising it is within Tableau's ability.

This method is certainly not the great, since it requires difficult manual work, which other software tools could do much quicker and easier. Nevertheless, theory shows that Tableau, or more generally, generic visualisation tools could be used in an out-of-the-box manner to overcome the posed difficulties.

3.3.6 Performance

During our evaluation, we often registered performance related problems. This motivated us to look more closely at this issue.

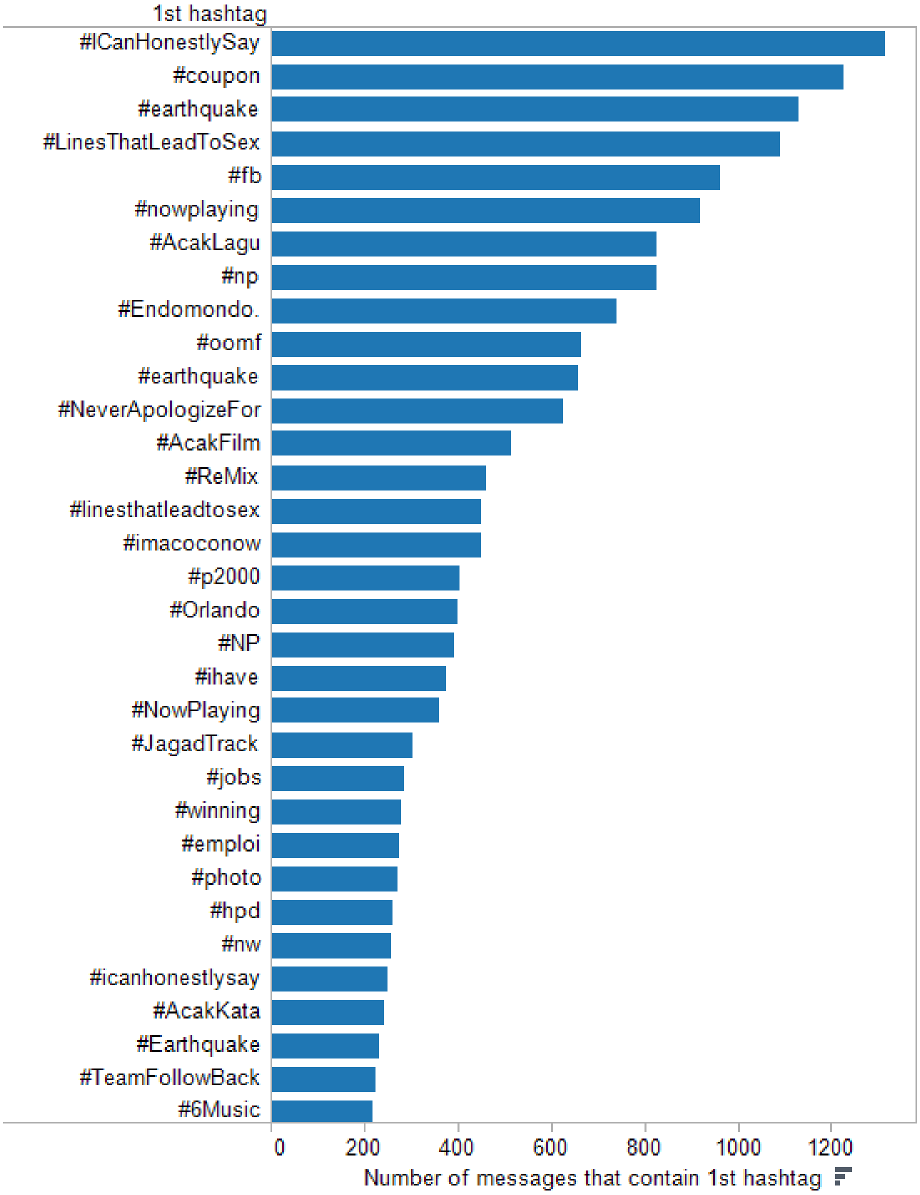


Figure 3.42: Number of messages that contain a hashtag (first hashtag only).

Number of records	**Spotfire**	**QlikView**	**Tableau**
1000	<1	<1	<1
10000	2	<1	3
25000	5	1	5
50000	10	2	8
100000	22	3	15
1000000	240	20	150

Table 3.5: Combined filtering and rendering times in seconds.

One major issue that all evaluated tools shared, was the inability to use more than one thread for their calculations. This has the effect that on modern multi-core and multi-processor computers, only one processor core is used. As a result, a lot of potential processing power is wasted. QlikView is able to use one thread per visualisation, which somewhat alleviates this when working on multiple large visualisations in parallel.

Spotfire

Spotfire was very fast with simple rendering tasks like changing the shapes of all the markers on a map chart, even when the whole data set was displayed. Doing calculations like filtering was very slow, as it seemingly iterated over the whole displayed data set. As a result, filtering to a very small amount of displayed data points resulted in a massive speed up. As can be seen in Table 3.5, Spotfire's performance was linearly related to the amount of data displayed.

While this is quite good for business intelligence scenarios, social media produces hundreds of millions of data points. Using the data from Table 3.5 to project the time it would take Spotfire to filter and render 100 million messages results in an estimated 6h 40m on the reference machine. This kind of time frame is prohibitive in disaster management scenarios and makes preprocessing a necessary step to make this program a viable alternative to specific disaster management tools.

QlikView

Since QlikView's extensions run inside Microsoft's Internet Explorer, we did not use them in our performance test, because it would not reflect the work QlikTech has done on their product. In the subsequent tests two data representations were used QlikView was quite fast with the application of filters and the subsequent rendering (see Table 3.5). However, the memory consumption was very high: out of the 12 GiB of memory on the reference machine, 7.6 GiB were allocated during the interactions that caused the rendering processes. The amount then gradually decreased while the program was idle. At those levels of memory usage, the program itself or the operating system would sometimes warn about the lack of memory. While Windows' warning did not disturb the work on the project, the map would not render any more after QlikView's warning and thus required an application restart.

Another issue was the locked up user interface. When there were big data sets – e.g. 10^6 Twitter messages – the user interface became unresponsive when the mouse hovered over the map chart. This indicates an inefficient representation of the drawn shapes that causes excessive calculations in response to mouse move events. The issue probably could have been avoided easily, if the hit test was only performed after the mouse hovered over the same spot for a configurable amount of time.

Tableau

Tableau's performance is good in general. The program functions with a high level of responsiveness most of the time, since the progress of time consuming tasks such as loading or filtering data is always shown. Such tasks are also interruptible.

Other jobs, such as zooming, panning or changing the graphical appearance of objects, execute much faster.

The only performance problem we encountered is caused by the misinterpreted coordinate values as long integers, which are significantly greater or less than the normal longitude and latitude values. It took Tableau an unusually long time of more than twenty minutes to visualise the data on a map.

The operation, which employed approximately only twenty percent of the CPU power, a normal amount of RAM and very little hard disk activities, had to be manually aborted after the twenty minutes mark. With a smaller data sample, Tableau was able to create a graphical visualisation, where all the invalid coordinate values were placed on the borders of the map.

Tableau normally doesn't fully utilise the computer's resources. The CPU and RAM usage is acceptable. Visualising data stored on a solid state disk didn't make a big different in performance compared to conventional hard drive.

Similar tasks, such as filtering 25,000 data records, don't always execute in the same amount of time, although the differences are not big. The performance test (Table 3.5) is provided as a reference. The performance of Tableau seems to increase linearly.

4 Toolkits for Individual Development

Harger and Crossno [19] examine a number of open source visualisation toolkits. These toolkits can be used to visualise arbitrary data, in accordance with their specific capabilities. One potential use of these toolkits is to implement new, individual solutions to visualisation problems like disaster management. For this reason, we will give a quick overview of the toolkits' abilities and a quick impression of their quality.

Axiis *Axiis*[1] supports basic visualisation and no analysis techniques. It is developed in ActionScript and based on Flash.

birdeye *birdeye*[2] has comprehensive visualisation technique and very basic analysis support. It is developed in ActionScript and based on Flash.

ESTAT *ESTAT*[3] supports very basic visualisation and no analysis techniques. It is developed in Java and based on Swing. Starting it proved difficult as the supplied Java WebStart application was missing resources.

GeoVista Studio *GeoVista Studio*[4] supports basic visualisation and no analysis techniques. It is developed in Java and based on Swing.

Gephi *Gephi*[5] supports basic visualisation and basic analysis techniques. It is developed in Java and based on Swing.

Google Visualization API *Google Visualization API*[6] supports basic visualisation and no analysis techniques. It is web-based and developed in JavaScript.

GraphViz *GraphViz*[7] is used to create graphs and not display them. Is developed in C and doesn't have a graphical interface. It does not support any relevant visualisation and analysis techniques.

Improvise *Improvise*[8] supports basic visualisation and no analysis techniques. It is developed in Java and based on Swing.

[1] http://www.axiis.org/
[2] http://code.google.com/p/birdeye/
[3] http://www.geovista.psu.edu/ESTAT/
[4] http://www.geovistastudio.psu.edu/jsp/index.jsp
[5] http://gephi.org/
[6] http://code.google.com/apis/charttools/index.html
[7] http://www.graphviz.org/
[8] http://www.cs.ou.edu/~weaver/improvise/index.html

infovis Toolkit The *infovis Toolkit's*[9] latest stable release is from 2006, with the latest alpha release bein from 2010. This leads me to conclude that it is not actively developed. It supports most basic visualisation techniques, notably excluding map-based visualisations, and almost no analysis techniques. The *infovis Toolkit* is developed in Java and based on Swing. Attempting to execute the supplied JARs leads to many exceptions being thrown, and the sample data wasn't loaded due to a loading error that was not further elaborated on.

JIT *JIT*[10] is web-based and developed in JavaScript. It does not include any analysis techniques, but incorporates most basic visualisation techniques. It is missing map-based visualisations. *JIT* provides some interactive example visualisations, which can be used as templates for individual implementations. It does not have the capability to load external data and relies on the user to load and prepare the data for its use.

JFreeChart *JFreeChart*[11] is actively developed and widely used. It is developed in Java and based on Swing. It includes some very basic graph implementations and no analysis techniques at all. A user manual is available for a "donation". This user manual is well-written and includes a lot of useful examples.

JGraphX *JGraphX*[12] is not officially released as a separate component and thus only available by cloning the GIT repository. It is developed in Java and based on Swing. It includes some basic visualisation and analysis techniques.

JUNG The latest release of *JUNG*[13] is from 2010, leading to the conclusion that it is not actively developed. It is developed in Java and based on Swing and SWT. Attempting to start the supplied example application leads to a crash due to a segmentation violation. Whether this crash is located in the toolkit, the example application or SWT is unclear. *JUNG* supports very basic graph implementations and several analysis techniques, among them clustering.

Mondrian *Mondrian*[14] contains a sample executable that is able to load our sample data set. It cannot handle its size well and takes a very long time to display any visualisations. It is developed in Java and based on Swing. Despite the rather conservative Java default settings, *Mondrian* did not experience problems due to a lack of RAM with our data set. It supports basic visualisation techniques as well as one map visualisation, and no analysis techniques.

NetworkX *NetworkX*[15] is implemented in Python and uses matplotlib as graphical frontend. It is distributed as a Python Egg. It supports almost no visualisation techniques and several basic analysis techniques,

[9] http://ivtk.sourceforge.net/
[10] http://thejit.org/
[11] http://www.jfree.org/jfreechart/
[12] http://www.jgraph.com/
[13] http://jung.sourceforge.net/
[14] http://rosuda.org/software/Mondrian/
[15] http://networkx.lanl.gov/

Prefuse The latest release of *Prefuse*[16] is a beta release from 2007. This leads to the conclusion that it is not actively developed. It is developed in Java and based on Swing. *Prefuse* supports some basic and advanced visualisation and almost no analysis techniques.

Flare *Prefuse Flare's*[17] latest release is an alpha release from 2009, indicating stalled development. It is developed in ActionScript and based on Flash. It supports basic visualisation techniques and some analysis techniques, including clustering.

Protovis *Protovis*[18] is not developed anymore – it has been succeeded by *d3js*. It is web-based and developed in JavaScript. It has comprehensive visualisation capabilities and knows no analysis techniques.

R *R*[19] is designed around a programming language and extensible. It supports basic visualisation and advanced analysis techniques.

Titan *Titan*[20] has comprehensive support for visualisation and analysis techniques. It can be used with C, Java, Python and TCL and uses Qt and TK as graphical interface.

Tulip *Tulip*[21] has comprehensive support for visualisation and and basic support for analysis techniques. It is developed in C++ and based on Qt.

VisAD *VisAD*[22] supports basic visualisation and no analysis techniques. It is developed in Java and based on Swing.

WilmaScope *WilmaScope*[23] supports almost no relevant visualisation or analysis techniques. It is developed in Java and based on Swing.

Zest *Zest*[24] supports basic visualisation and no analysis techniques. It is developed in Java and based on Eclipse.

Most of these toolkits do not contain comprehensive implementations of visualisation or analysis techniques. It may therefore be necessary to combine several or extend one of them to create a comprehensive visualisation tool.

To give a better overview of the analysis and visualisation technique support, we have included Table 4.1.

[16] http://prefuse.org/
[17] http://flare.prefuse.org/
[18] http://mbostock.github.com/protovis/
[19] http://www.r-project.org/
[20] http://titan.sandia.gov
[21] http://tulip.labri.fr/TulipDrupal/
[22] http://www.ssec.wisc.edu/~billh/visad.html
[23] http://wilma.sourceforge.net/
[24] http://www.eclipse.org/gef/zest/

Toolkit	**Visualisation Support**	**Analysis Support**
Axiis	basic	none
birdeye	comprehensive	very basic
ESTAT	very basic	none
GeoVista Studio	basic	none
Gephi	very basic	basic
Google Visualization API	basic	none
GraphViz	none	none
Improvise	basic	none
infovis Toolkit	basic	very basic
JIT	basic	none
JGraphX	basic	basic
JFreeChart	very basic	none
JUNG	very basic	some advanced
Mondrian	basic	none
NetworkX	very basic	some advanced
Prefuse	advanced	very basic
Flare	basic	basic
Protovis	comprehensive	none
R	basic	advanced
Titan	comprehensive	comprehensive
Tulip	comprehensive	basic
VisAD	basic	none
WilmaScope	none	none
Zest	basic	none

Table 4.1: Functionality of visualisation tookits.

5 Conclusion

In this study, we explored the information needs of disaster analysts during an ongoing event and evaluated some generic visual analytics tools in regard to their ability to provide this information. We consistently found that these tools have their strengths in after-the-fact analyses, while they lack real-time analysis capabilities. This makes them unsuitable for disaster management scenarios. While Zhang et al. [31] speculate that these capabilities will see development in the near future, the current deficiencies make specific disaster management tools necessary for the foreseeable future.

The only way to make the evaluated tools usable with a data set of hundreds of millions to billions of tweets is to reduce its size with preprocessing. This preprocessing step has to be executed independently of the tools and may still yield large volumes of data for generic disaster discovery. The ability to use multiple processor cores for visualisations would somewhat alleviate this issue, but so far isn't available.

Of the evaluated tools, no clear winner can be chosen. The feature completion is measured by the features needed by disaster analysts as presented in the tables in chapter 3.3 Tool Comparison. The feature completion of the three main programs in this study is very close without considering extensions. With them, QlikView scores best in this regard, while Spotfire is slightly ahead of Tableau. QlikView's extensions have the disadvantage of only being able to visualise data sets of up to 10,000 records, which renders them less useful to disaster management.

While Spotfire is the most user-friendly, it is also the slowest. QlikView is fastest, albeit less user friendly. Tableau scores between the other two programs both in user-friendliness and speed.

All tools handled the visual representation of geographical information well. While QlikView needed extensions to perform well at this task, both Spotfire and Tableau managed with integrated functionality. The disaster discovery process was the same in all the programs: none of them offered any intuitively available hint when and where an event might be found. They all managed to quickly find a known disaster using a simple keyword search and visually present it in a suitable fashion.

Both QlikView and Spotfire are extensible, but only QlikView has an active community that makes use of this. QlikView's use of Internet Explorer as an extension engine has its advantages and disadvantages. The main advantage is the ease with which an extension can be created – the main disadvantage is the performance of *JavaScript*, which causes the limit of 10,000 usable data records. Overall, better and native plugin support would benefit all evaluated programs.

As a result, our overall ranking of the evaluated tools is:

1. QlikView
2. Tableau
3. Spotfire

The main influence on this ranking is speed. While feature completion is very relevant, the differences in the programs' available features is small enough to not impact this ranking at all.

In conclusion, generic visual analytics tools are currently not fit to replace specifically developed tools like SensePlace2, TwitInfo, TwitterReporter or Twitcident. They may, however, become suitable replacements in the future, as they absorb modern and more specialised visualisation and analysis techniques (see Zhang et al. [31]).

Bibliography

[1] 2 - FEMA Disaster Analysis.qvw. URL: http://qonnections.geoqlik.com/GeoQlik/proxy/QvAjaxZfc/opendoc.htm?document=2%20-%20FEMA%20Disaster%20Analysis.qvw&host=QVS@aslw0026&anonymous=true [cited 2013-02-27]. (Cited on page 49)

[2] Business Discovery: Business Intelligence For Everyone | QlikView. URL: http://www.qlikview.com/ [cited 2013-02-14]. (Cited on page 23)

[3] Facebook. URL: http://www.facebook.com [cited 2013-02-14]. (Cited on page 9)

[4] GeoQlik, the mapping extension for QlikView. URL: http://www.geoqlik.com [cited 2013-02-14]. (Cited on page 49)

[5] Google Maps. URL: http://maps.google.com [cited 2013-02-14]. (Cited on page 47)

[6] Jigsaw homepage. URL: http://www.cc.gatech.edu/gvu/ii/jigsaw/ [cited 2013-02-14]. (Cited on pages 28 and 31)

[7] JMP Software - Data Analysis - Statistics - Six Sigma - DOE. URL: http://www.jmp.com/ [cited 2013-02-14]. (Cited on page 27)

[8] QlikCommunity: Another Google Maps Extension. URL: http://community.qlikview.com/thread/36301 [cited 2013-02-14]. (Cited on page 48)

[9] QlikCommunity: OpenStreetMap Extension Object. URL: http://community.qlikview.com/thread/60652 [cited 2013-02-14]. (Cited on page 48)

[10] Spotfire homepage. URL: http://spotfire.tibco.com [cited 2013-02-14]. (Cited on page 21)

[11] Twitter. URL: http://www.twitter.com [cited 2013-02-14]. (Cited on page 9)

[12] Wikipedia: 2011 Virginia earthquake. URL: http://en.wikipedia.org/wiki/2011_Virginia_earthquake [cited 2013-02-27]. (Cited on page 19)

[13] Wikipedia: Heat map. URL: http://en.wikipedia.org/wiki/Heat_map [cited 2013-02-14]. (Cited on page 47)

[14] Wikipedia: Tableau Software. URL: http://en.wikipedia.org/wiki/Tableau_Software [cited 2013-03-01]. (Cited on page 25)

[15] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Semantics + filtering + search = twitcident. exploring information in social web streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, HT '12, pages 285–294, New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2309996.2310043, doi:10.1145/2309996.2310043. (Cited on page 11)

[16] Tom Brown. Tableau Extracts – What / Why / How etc. URL: http://www.theinformationlab.co.uk/2011/01/20/tableau-extracts-what-why-how-etc/ [cited 2013-02-28]. (Cited on page 36)

[17] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, MDMKDD '10, pages 4:1–4:10, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1814245.1814249, doi:10.1145/1814245.1814249. (Cited on pages 9, 11 and 58)

[18] Junghoon Chae, Dennis Thom, Harald Bosch, Yun Jang, Ross Maciejewski, David S Ebert, and Thomas Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition, 2012. URL: http://rmaciejewski.faculty.asu.edu/papers/2012/Chae_SocialMedia.pdf. (Cited on pages 12 and 58)

[19] John R. Harger and Patricia J. Crossno. Comparison of open-source visual analytics toolkits. pages 82940E–82940E–10, 2012. URL: http://dx.doi.org/10.1117/12.911901, doi:10.1117/12.911901. (Cited on pages 11 and 67)

[20] A.M. MacEachren, A. Jaiswal, A.C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. Senseplace2: Geotwitter analytics support for situational awareness. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 181 –190, oct. 2011. doi:10.1109/VAST.2011.6102456. (Cited on pages 9 and 11)

[21] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1978942.1978975, doi:10.1145/1978942.1978975. (Cited on page 11)

[22] Brett Meyer., Kevin Bryan, Yamara Santos, and Beomjin Kim. Twitterreporter: Breaking news detection and visualization through the geo-tagged twitter network. In *Proceedings of the 26th International Conference on Computers and Their Applications (CATA-2011)*, 2011. (Cited on page 11)

[23] Michael J. Paul and Mark Dredze. A model for mining public health topics from twitter. Technical report, Johns Hopkins University, 2011. URL: http://www.cs.jhu.edu/~mdredze/publications/2011.tech.twitter_health.pdf. (Cited on pages 9 and 12)

[24] Karl Popper. Tableau Custom Geocoding outside the US and UK. URL: http://www.clearlyandsimply.com/clearly_and_simply/2010/10/

tableau-custom-geocoding-outside-the-us-and-uk.html [cited 2013-03-01]. (Cited on page 50)

[25] Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models, 2010. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1528. (Cited on page 12)

[26] Tableau Software. Tableau Desktop. URL: http://www.tableausoftware.com/products/desktop [cited 2013-03-01]. (Cited on page 27)

[27] D. Thom, H. Bosch, S. Koch, M. Worner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 41 –48, 28 2012-march 2 2012. doi:10.1109/PacificVis.2012.6183572. (Cited on pages 9, 12 and 58)

[28] United States Search and Rescue Task Force. Disaster Intensity Scales. URL: http://www.ussartf.org/disaster_intensity_scales.htm [cited 2013-03-01]. (Cited on page 14)

[29] Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '10, pages 1079–1088, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1753326.1753486, doi:10.1145/1753326.1753486. (Cited on pages 11 and 14)

[30] Jianshu Weng and Bu-Sung Lee. Event detection in twitter, 2011. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2767. (Cited on pages 9, 12 and 58)

[31] Leishi Zhang, Andreas Stoffel, Michael Behrisch, Sebastian Mittelstädt, Tobias Schreck, René Pompl, Stefan Weber, Holger Last, and Daniel Keim. Visual analytics for the big data era – a comparative review of state-of-the-art commercial systems. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 173 – 182, oct. 2012. (Cited on pages 11, 20, 21, 22, 71 and 72)

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschift

Ort, Datum, Unterschift

Ort, Datum, Unterschift