Geosoftware II, WiSe 2019/20
Web platform for the integration of weather and social media data during extreme weather events

Christian Knoth, Daniel Nüst, Edzer Pebesma

October 23, 2019

Introduction

Geosoftware II provides a challenge to groups of geoinformatics students to apply their technological and conceptual skills as well as their teamwork capabilities. Students cooperate to solve a realistic and relevant spatial issue with computer science methods. This semester, the challenge lies in creating a web platform for integrating real world weather data and social media data from different sources into a coherent visualisation in an extreme weather scenario.

The teachers take the role of the *customer*, i.e. an emergency response agency. The customer publishes an invitation to bid[1] with a problem statement and a set of requirements for solving the problem. The student groups are the *contractor*, i.e. the architects, designers, and developers of a software system to advance the state-of-the-art in disaster response. Each group bids with their own approach to the project, implements it, and presents it to the customer. Groups are encouraged to name their group and product for easy recognition, including GitHub/GitLab organisation, logo, etc.

The final project grade is determined both by the main result, i.e. the developed software, and the process of reaching it, e.g. the collaboration process, communication, group work, or presentation.

Veranstaltungsnummer im Vorlesungsverzeichnis: 148765

[1] https://en.wikipedia.org/wiki/Invitation_for_bid

Invitation to bid (Lastenheft)

Project title

Web platform for integration of weather and social media data during extreme weather events

Problem statement

The amount of openly available geospatial data, generated by various sources and in various forms, is growing dramatically. A lot of this data is highly dynamic, i.e., it is provided not only in high volumes but also in (near) real-time. The heterogeneous structure and dynamic availability of this data creates new technical challenges, but also opportunities for diverse applications, such as emergency response. In crisis situations due to extreme weather events, reports in social media can be a source of information for decision makers, e.g., in emergency management. Specifically, geotagged messages or images, i.e. posts with a coordinate or textual location, from social media can augment established information sources, provide more detailed insights in areas with only few first-hand reports from other channels, help establishing an overview of the situation, and might even open channels for bidirectional communication. The major challenge is to extract only relevant and authentic reports from the vast amount of information available, to correctly interpret these reports, and to draw the right conclusions from them.

To help us tackle these challenges, the project should develop a web platform that intersects geotagged information from social media with real-time weather data. By jointly visualizing and analyzing all the available data, we expect to be able to quickly extract the most relevant information from diverse sources. This way we can increase the actuality and detail of our situation reports and ensure to adopt the most suitable measures to respond to emergencies. The main use case for our application is an extreme weather event.

State-of-the-art

The advent of new information and communication technologies has had a huge impact on the way emergency response in crisis situations is organized. For example, eye-witnesses can support relief operations by issuing situation reports and help requests via SMS, social media, email and other means of online communication. Localizing and mapping these first-hand reports including time stamps has been as part of relieve campaigns in various types of disasters, e.g. conflicts (Okolloh, 2009), tsunamis (Acar and Muraki, 2011) or earthquakes (Heinzelman and Waters, 2010; Hester et al., 2010).

The integration of social media and additional geospatial data, specifically remote sensing, has been studied in the contexts of monitoring natural disasters, monitoring of urban environments, land cover change or environmental issues

(Hultquist et al., 2015; Li et al., 2017). Most of these studies use remote sensing sources such as LIDAR (light detection and ranging) data, SAR (synthetic aperture radar), or optical satellite sensors with comparably low temporal resolution (Liu et al., 2017). Social media is often used to complement this data with more up-to-date information.

In contrast to these projects, the specific challenge in our application scenario is to integrate data from different, *highly dynamic* sources, where social media but also remote sensing data (e.g., rain radar) is available in near real-time. A stable, performant integration with a very simple workflow for one specific disaster scenario outweighs other potential features.

Project goals

This project will integrate data from two very different but potentially complementary sources. Remotely sensed weather data and social media data differ in scale, temporal resolution, data formats, usage rights, and many more properties. The goal is to provide a system where a civil protection expert can quickly inspect the combined information streams in a high pressure situation to derive appropriate actions. This situation requires the user's interaction to be very straightforward, the information to be up-to-date, and the source data to be transparent.

Concrete requirements and the process to fulfil this goal are detailed in the remainder of this document.

Requirements

Functional requirements

The main functionality of the web-based platform is the selection of an area of interest and subsequent display of precipitation and social media data for that area. The interaction is a two step process and should require minimal user interaction. The realisation requires a server and a client-side (browser) component. The implementation may be built upon existing state-of-the-art solutions or a new software project.

Data and Processing Weather data and social media data integration and processing happen in a web service.

- Precipitation data for the whole of Germany based on radar observations
- The latest available precipitation data is used by default
- Geocoded social media data from Twitter
- Retrieved data is cached in the server component for matching user actions and re-used for a suitable amount of time (e.g. until new data is available, or configurable for social media)

- Data source and data age (timestamp) are clearly shown in the user interface
- Based on the current map extent, the area of a currently ongoing heavy precipitation event is automatically determined and shown on the map
- All data processing happens in the server component
- Processing steps must be communicated to the user (e.g. "downloading, processing data, preparing visualisation")

API and Configuration A well-defined API is used for communication between user interface and web service, which is easily configurable for system administrators and allows third parties to integrate functionality in independent user interfaces or systems.

- All features are exposed in a well-defined web API
- API follows the RESTful paradigm unless an existing standardised API is extended (used standards pending acceptance by the customer)
- API endpoints accept and return valid JSON in responses, including error messages
- API uses appropriate HTTP status codes and HTTP headers
- All configuration is possible via plain text files, e.g. in YAML format
- Changes in the configuration are active after restarting the service
- API exposes all data in a single coordinate reference system

User interface The user interacts with the system using a graphical web browser-based user interface.

- The language of the user interface is English
- The default map location on opening the application is configurable
- Precipitation data and areas of extreme events are shown on an interactive tiled web map
- Map supports for pan and zoom interaction with both buttons and computer mouse (Desktop computer)
- Locations of social media posts are shown on the same map
- The positional accuracy of data is transparently communicated to the user
- The author/account and text/image content of social media posts can be inspected without leaving the system

- The shown social media posts can be filtered client-side with a text search
- Map supports at least two background layers: street data, satellite imagery
- The area for which social media data are queried can be interactively configured
- The UI uses only the project API for data retrieval and processing (may use other APIs for background data)
- The URL is updated after user interactions (map location, text filter) and works as a permalink[2]
- Visual progress reporting
- User documentation is accessible from the UI

Bonus feature One bonus feature from the following list or own ideas for features must be completed. *Only one contractor may implement each feature (first come first serve).*

- an additional geospatial data layer, e.g. elevation models, is used in modelling and visualisation
- animation of data with the last 10 time steps (play/pause functionality, animation download/export)
- real-time updating of the map with social media posts and asynchronous notifications in the UI for new events (browser notification and message (e.g. Mattermost chat, email, messenger))
- user-controlled area delineation algorithm (thresholds, maximum size, (un)-populated areas, ...) and an advanced visualisation of these data properties
- mobile device support (responsive UI, geolocation, touch interaction), tested for a well-defined percentage of mobile devices as used today
- support for historic data, including a visual comparison of two points in time
- summary statistics (e.g. aggregations, descriptive statistics, spatial statistics, text-mining) about extracted weather data and social media data
- integration of at least one additional social media channel as data source (e.g., Instagram)

Bonus feature decisions can be submitted any time but must be submitted at least *one month before final delivery.* The bonus feature must be described in the bid or a change request. The bonus feature must be described in the project and user documentation. It underlies the same non-functional requirements as regular features.

[2] https://en.wikipedia.org/wiki/Permalink

Non-functional requirements

Training and Demonstration For training and demonstration purposes, the system must support at least two scenarios based on historic data, i.e. a fixed time stamp (not "now") and fake or recorded data snapshots. These scenarios should be accessible via a permalink and the time stamp must be configurable, so that the platform may show a "regular" view first and then after a refresh an event is happening.

Maintainability The developed software must be published as open source software under an OSI-approved license according to the license requirements. The contractor must ensure license compatibility of other used or extended software.

An established build and packaging workflows as well as dependency management must be applied, e.g. Gradle for Java, npm for JavaScript, R packages for R, or PyPI packages for Python. If more than one programming language is used in a component then different build and packaging steps may be scripted with a Makefile.

Documentation on building, installing and configuring the system must be provided in Markdown-formatted documentation files using the appropriate markup for lists, links, etc.

User friendliness The system must support intuitive use to the extend that targeted group can use its features without further documentation. Common practices for web-based user interfaces and interaction paradigms should be applied. The user interface and visualisation must be suitable for color-blind users. The system must be tested and fully functional with browsers representing at least 80% of current users.

Performance In general, a user interaction must result in a display change within 1 second to allow users to stay focused on their current train of thought; complex page contents may be loaded asynchronously; interactive visualisations must react within 0.1 seconds to give a user the impression of direct manipulation.[3] The full workflow of area selection, data retrieval, and visualisation must not take longer than 20 seconds.

The contractor provides a test script to evaluate the performance using at least 5 different HTTP operations or views with an appropriate number of repetitions.

Deployment Docker must be used to ensure easy deployment. Used images must be based on Dockerfiles and all base images hosted on Docker Hub (or comparable).

[3]Source: http://www.nngroup.com/articles/powers-of-10-time-scales-in-ux/

Project management The contractor shall apply an agile development process, e.g. oriented at Scrum. The customer must be given access to the contractor's online task management to observe the progress. The contractor may include the customer in regular meetings for updates and feedback on the progress.

Deliverables and deadlines

Bid The bid (Pflichtenheft[4]) must be submitted to the customer via WWU's Learnweb no later than **October 31st, 14:00 CET**. It must comprise an implementation plan for all requirements of the invitation to bid (Lastenheft) as well as a schedule for the implementation as a single PDF document. The bid must include *user stories* realising all requirements. The bid document may be English or German. Its content should follow common standards for content and structure of bids.

Changes The bid is a binding agreement between customer and contractor. All changes to the bid after first acceptance require a written confirmation (email) by the other party. The contractor may direct *change requests* to the customer via email to both `daniel.nuest@uni-muenster.de` and `christian.knoth@uni-muenster.de`. A change request includes a short summary of the changes in the email body and an updated version of the bid as an attachment.

Final delivery The final delivery must be submitted one day in advance of the final presentation, on **January 28th 2020, 23:59 CET**, and contain the following items:

1. project report
   (a) a single PDF document submitted via Learnweb
   (b) cover page contents: team members, team name (optional), references to all parts of the delivery (code, documentation, etc.)
   (c) screen-shots of all relevant UI components and workflow steps
   (d) describes all requirements/user stories from the bid
2. commented source code in a single code repository on zivgitlab, GitLab.com, or GitHub.com; delivery in more than one repository must be formally accepted by the customer
3. ready-to-use Docker images (if multiple images with docker-compose configuration) and Dockerfiles

[4] https://de.wikipedia.org/wiki/Pflichtenheft, https://wiki.induux.de/Pflichtenheft, and http://www.infrasoft.at/downloads/Anleitung_zum_Pflichtenheft.pdf

4. installation and maintenance documentation (e.g. online Wiki or included in repository)

5. integrated user documentation (i.e. within the regular UI)

6. API test suite (e.g. as documented collection of `curl` requests in a Jupyter/R Markdown notebook, or a Postman/SoapUI Open Source project)

Acceptance criteria The acceptance criteria encompass the fulfilment of all functional and non-functional requirements as described in the accepted bid, and the complete and on-time submission of all deliverables.

References

Acar, A. and Muraki, Y. (2011). Twitter for crisis communication: Lessons learned from Japan's tsunami disaster. *Int. J. Web Based Commun.*, 7(3).

Heinzelman, J. and Waters, C. (2010). Crowdsourcing Crisis Information in Disaster-Affected Haiti. *United States Institute of Peace Special Report*, 252.

Hester, V., Shaw, A., and Biewald, L. (2010). Scalable crisis relief: Crowdsourced sms translation and categorization with mission 4636. In *Proceedings of the First ACM Symposium on computing for development, London, UK, 17-19 December 2010*, New York, NY, USA. ACM.

Hultquist, C., Simpson, M., Cervone, G., and Huang, Q. (2015). Using nightlight remote sensing imagery and twitter datato study power outages. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop*, Seattle, WA, USA. ACM.

Li, J., Benediktsson, J. A., Zhang, B., Yang, T., and Plaza, A. (2017). Spatial technology and social media in remote sensing: A survey. *Proceedings of the IEEE*, 105(10):1855–1864.

Liu, X., He, J., Yao, Y., Zhang, J., Liang, H., Wang, H., and Hong, Y. (2017). Classifying urban land use by integrating remote sensing and social media data. *International Journal of Geographical Information Science*, 31(8):1675–1696.

Okolloh, O. (2009). Ushahidi or 'testimony': Web 2.0 tools for crowdsourcing crisis information. *Participatory Learning and Action*, 59(1):65–70.