Analyzing Twitter Data using Snap!

Andreas Grillenberger, *andreas.grillenberger@fau.de*
Computer Science Education Research Group, Universität Erlangen-Nürnberg

Ralf Romeike, *ralf.romeike@fau.de*
Computer Science Education Research Group, Universität Erlangen-Nürnberg

Abstract

In this paper, we present a software tool which enables students to discover and learn about data stream systems (DSS) in a constructionist way. As DSS are part of the ongoing developments and innovations in data management, which are often summarized by the term *Big Data*, this tool shows in an exemplary way how these complex topics can be incorporated into school teaching. Therefore, we extended the block-based programming environment *Snap!*, so that it supports analyzing the Twitter data stream even without having any pre-knowledge on data stream analysis.

Keywords

data stream system; data analysis; block-based; software tool; big data

Introduction

Today, everyone is generating large amounts of data and is always confronted with the results and implications of data analysis. Not only smartphones and websites are producing and processing data continuously, but data are also relevant to many other parts of the daily life. Despite their relevance for everyone (cf. Grillenberger & Romeike 2015), these topics seem hard to understand because of their complexity. Nevertheless, various ideas that are central to these topics can be identified: for example, causality or correlation as analysis paradigm, the analysis methods used, (in particular categorization, clustering and association analysis), consistency, redundancy and parallelization.

For discussing these concepts at school, appropriate software tools are required in order to let students gather their own experiences with these systems. While in other teaching contexts, often professional tools are also used for teaching purposes, typical tools in the context of “Big Data” are too complex for providing them to students without extensive further support. Instead, finding ways for reducing the complexity is important for allowing students to build up own knowledge on such systems in a constructionist way.

Hence, we prepared the topic *data stream systems* in a way, that students can discover and understand the efficiency of this modern data analysis approach, the accompanying threats but in particular also the chances and possibilities that are opened up by these systems on their own. The software tool presented in the following supports the design of own data analyses without pre-knowledge on data stream systems, the underlying concepts and ideas or the tool itself. In this way, the presented approach is prototypical for incorporating the complex and innovative topics in data management / Big Data for (secondary) computing education.

The *Snap!* Data Stream Extension

For enabling students to conduct own data analyses, we extended the block-based programming environment *Snap! (Harvey and Mönig 2015)* in order to support working with and processing of data streams. We decided for using *Snap!* because of its low barriers to entry, but also because it provides diverse possibilities and is easily extendable. Hence, the students can not only use the provided functionalities, but also extend them on their own in order to provide additional possibilities. As data stream source, we selected Twitter, as this social network provides a huge

amount of data easily accessible via a programming interface. Also, these data can be used without data privacy risks, as all the accessed data are also available via Twitter's public website; the students can hence not only access the tweets text, but also e. g. data on the user, followers or location data.

All new implemented blocks are based on original *Snap!* blocks only. Hence, they can be used in the original *Snap!* installation and most of its derivatives. Also, students can take a look behind the implementation and customize it, as the blocks may be edited as all other own blocks. For restricting the creative openness as less as possible, we allow accessing all the attributes provided by Twitter for each tweet. Thus, students have much creative freedom when analyzing the Twitter stream.

For visualizing the analysis results in an easy and clear way, we also implemented two visualizations: a bar chart (cf. fig. Figure 1), which is e. g. suitable for showing the results of categorization tasks, and a map (cf. fig. Figure 2) that can be used for visualizing spatial data. Both visualizations can also be used as example implementation for additional visualizations, which can also be implemented by the students on their own. In fig. Figure 3, we show an example program for categorizing all the incoming tweets by language; this code generates the image shown in fig. Figure 1.

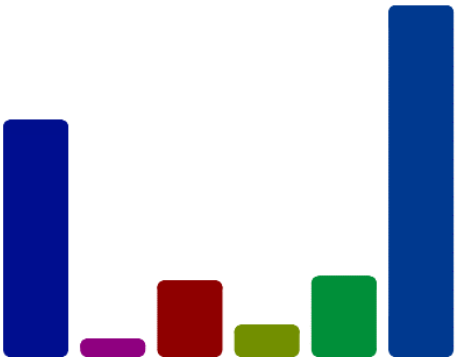

Figure 2. Bar chart visualization

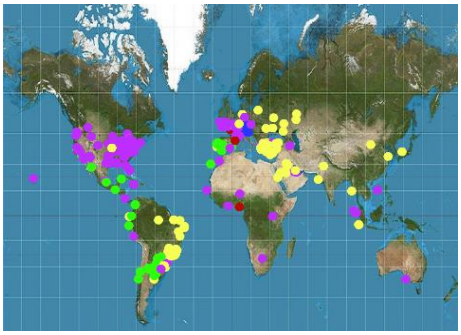

Figure 3. Map visualization
(map image ©2011 Strebe, CC-BY-SA 3.0)

Conclusions

With the simple-to-understand programming environment *Snap!* and by restricting to the basic ideas and concepts, we can break down the complex topic data stream systems to a level suitable for students. The extensibility of *Snap!* provides the advantage that despite the low entry level, also complex analyses are possible. This is further supported by basing on the real and infinite data source Twitter. The relevance and attractiveness of topics like data stream system can be further emphasized by incorporating other innovations: e. g. with the upcoming *Internet of Things (IoT)*, huge amounts of data are generated and analyzed—and it can be assumed that data stream systems will also have high importance there. By combining the described tool with for example objects created in physical computing projects, the *IoT* can be reconstructed in small-scale.

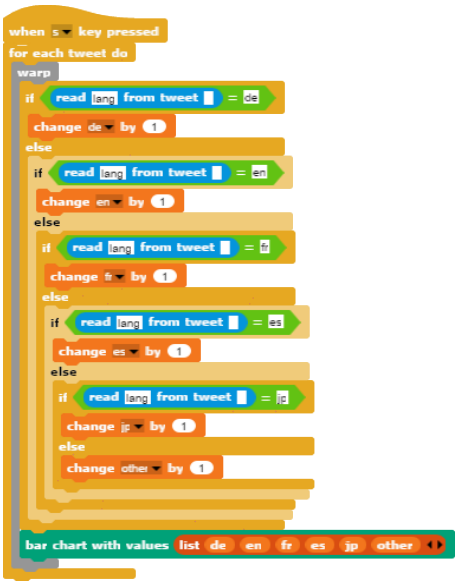

Figure 1. Bar chart visualization

References

Grillenberger, Andreas; Romeike, Ralf (2015) *Analyzing the Twitter Data Stream Using the Snap! Learning Environment*. In Brodnik, Andrej; Vahrenhold, Jan (Eds.) Informatics in Schools. Curricula, Competences, and Competitions. Springer International Publishing.

Harvey, Brian; Mönig, Jens (2015) *Snap! Programming Language*. http://snap.berkeley.edu