ConceptCloud 2.0: Visualisation and Exploration of Geolocation-Rich Semi-Structured Data Sets

Tiaan du Toit[1], Joshua Berndt[2], Katarina Britz[1], and Bernd Fischer[2]

Information Science[1] & Computer Science[2],
Centre for AI Research, Stellenbosch University, South Africa
tiaandutoit@gmail.com

Abstract. ConceptCloud is a flexible interactive tool for exploring, visualising, and analysing semi-structured data sets. It uses a combination of an intuitive tag cloud visualisation with an underlying concept lattice to provide a formal structure for navigation through a data set. ConceptCloud 2.0 extends the tool with an integrated map view to exploit the geolocation aspect of data. The tool's implementation of exploratory search does not require prior knowledge of the structure of the data or compromise on scalability, and provides seamless navigation through the tag cloud and the map viewer.

1 Introduction

Semi-structured data such as product reviews or event logs contains embedded meta information that describe semantic elements, but does not conform to the formal structure of a data model. Many such data sets incorporate some geolocation aspect, for example the location of the vineyard for wine reviews, or the accident location for traffic data. This paper describes a map extension to ConceptCloud, a visualisation and exploration tool for semi-structured data sets.

ConceptCloud uses a formal concept lattice generated from the input data as the underlying navigation structure [2, 3]. The data is presented in an interactive *tag cloud*, providing the user with both an intuitive representation of the data set and allowing for interactive navigation through the data. Tag clouds are a representation of the number of occurrences of the attributes and objects in the data set, wherein the size of each tag displayed is based on the frequency of that tag within the data set. Navigation is achieved by tag selection and deselection, removing the confines of predefined search paths.

The software allows the user to iteratively select an attribute or object tag in the tag cloud, and the tool adjusts the tag cloud to display all other tags attached to objects possessing the selected attribute tag(s). This is achieved by maintaining a focus concept from which a tag cloud is created.

Formally, the *focus concept* $c := \langle O, A \rangle$ is the concept whose extent is the set of objects that share the set of currently selected attributes, $F \subset A$, within the tag cloud. The focus concept can be further refined by iteratively adding elements to F. When an additional attribute is added to F, the focus concept is updated by computing the meet of the current focus concept c and the concept introduced by the additional attribute. This strategy obviates the need to compute an entire concept lattice or implication base, which is not feasible for large data sets.

The explorative search process corresponds to the process of stepping through a concept lattice, wherein the selection of an attribute moves the user to the point in the lattice where all linked objects contain that attribute. As more attributes are selected, the user moves further down the lattice. The reverse of this process is also possible. ConceptCloud has been used in many domains as an effective method for knowledge discovery using tag cloud visualisation and navigation [4, 5].

With the rapid increase in always-on, embedded GPS devices, geolocation rich data is becoming more prevalent. This abundance presents a new opportunity for knowledge discovery by exploring the geolocation aspect of the data, but it also demands a different visualisation approach [8] — clearly, textually displaying latitude and longitude in a tag cloud is not optimal.

Historically, maps have most commonly been used to visualise geolocation information. However, while maps are immediately useful for visualising the geolocation aspect of data, fully integrating maps into an interactive tag cloud explorer requires that the exploration also be driven from the map. Thus an important function of this tool is *to make the exploration of the data work bidirectionally,* i.e., to update the map when changes are made in the tag cloud and vice versa. Moreover, the rate at which such data is being generated is increasing, resulting in ever larger sets. This requires that *the visualisation and exploration tool be highly scalable in order to process large data sets.* With maps providing a time-tested method of exploring the geolocation aspect of data, and tag clouds providing an effective method for facilitating knowledge discovery and data visualisation for semi-structured data, the overall goal of the ConceptCloud 2.0 extension described here is therefore to merge these two proven methods into an integrated and scalable system.

2 User Perspective

ConceptCloud 2.0 generates map pins based on the objects' geolocation attribute, where available, and uses the Google Maps JavaScript API[1] to render them on a fully interactive map. However, to prevent the generated pins from occluding each other, ConceptCloud 2.0 clusters them automatically within a shifting geolocational tolerance, based on the current zoom level of the map, and displays a count of the pins in each cluster.

[1] https://developers.google.com/maps/documentation

Fig. 1 (Left) shows this view for an example crime data set which contains, for each crime, its category (e.g., *Theft*) or sub-category (e.g., *Stock theft*) and the name and location of the police station where the crime was reported.

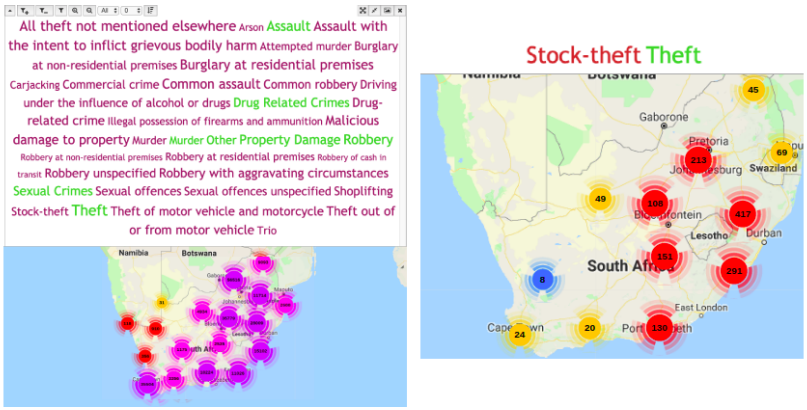

Fig. 1. Left: Interface applied to Crime Data; Right: Data filtered for Stock-theft

The pins then function the same way as the tags in the tag cloud. The user can thus explore the data by selecting an attribute in the tag cloud, which will update the cloud as before [4] but will also update the map view to display only the objects (resp. their pins) with that selected (focus) attribute. Fig. 1 (Right) shows the interface after the user has selected the *Stock theft* sub-category from the tag cloud. The tag cloud now contains only the selected *Stock theft* tag and the map viewer only the (clustered) pins corresponding to the *Stock theft* objects. (The *Theft* crime category tag is still visible since *Stock theft* belongs to this category). Alternatively, the user can select an individual marker which will "drill down" the map view, i.e., decrease the map's scale, redraw and re-cluster the pins, and update the tag cloud to reflect only the objects that are still represented on the map view.

The pin clusters mentioned above also have a formal correspondence in the concept lattice in the form of *biclusters*. Ordinarily, a bicluster is defined as a pair (A,B) of inclusion-maximal sets of objects and attributes such that almost all objects in A have almost all attributes in B. This technique has been implemented and applied successfully to mine numeric data sets using triadic formal concept analysis [7]. In our case a bicluster emerges as an element of the concept lattice by forcing the inclusion of all geolocation attributes from the object in a pin cluster to all other objects in this pin cluster. These naturally occurring biclusters can be mined to view common trends where possible.

Fig. 2 shows the new tag cloud generated from biclusters from the central Cape Town area. The user can generate this bicluster tag cloud by right-clicking either a single or multiple pin clusters as well as individual pins. This bicluster

tag cloud corresponds to a smaller lattice created from the original lattice, with its own focus concept, which can be navigated through independently of the main lattice.

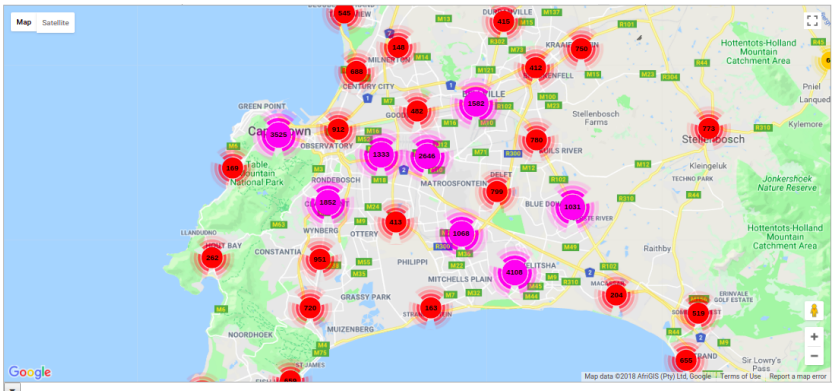

Fig. 2. Central Cape Town Bicluster

3 Implementation

ConceptCloud 2.0 adopts a client-server architecture. Its server is written in Java and uses the Play! framework[2] along with a PostgreSQL database to generate the concept lattice. The amount and nature of pre-processing that is required in order for ConceptCloud to consume data depends on the source. Typically some standard data cleanup and NLP techniques are applied, such as stemming and the removal of stop words. ConceptCloud accepts a JSON file of objects consisting of attributes only. Along with this, the user needs to specify the attribute types present in the object, in particular, the geolocation attribute which must be of format "lat, long".

The geolocation attribute of the objects are used to display the marker objects in the map viewer, although they also appear in the tag cloud as latitude and longitude pairs. When considering the underlying context table, each

[2] https://www.playframework.com/

lat-long pair is an attribute, with each object having at most one geolocation attribute.

The map and marker objects in the browser-based client are populated by a single specialised server call. The size of the visible map in the Google Map viewer is dependent on the size of the viewer window and the zoom level. Using the dimensions of this viewer, we then calculate the visible radius, and use this radius together with the zoom level and centre coordinates of the map viewer to only retrieve the map pins that are visible to the user. We keep track of movement in the map viewer and make additional server calls whenever the map is moved to a new location.

ConceptCloud allows the user to pre-configure the data attributes they wish to appear in each map pin object, allowing for a far smaller, and therefore more responsive server call to create each marker. The server returns a set of objects, containing at least an identifier and the geolocation, from which a Google Maps marker object is created and populated with the pre-configured attributes of that object. If desired and available, the user can populate a tool-tip text window of the marker with additional meta-data and links to external resources involving the object.

In order to maintain a single focus information retrieval navigation algorithm [6], Boolean disjunctive selection is used when dealing with biclusters. This technique involves modifying the underlying context table of the lattice and generating a new temporary concept lattice of only the selected objects from the data set. These are either from objects in the bicluster, or objects selected by the user, or multiple biclusters selected by the user.

Computing the disjunction of two or more objects involves assigning a new meta-tag to the selected objects, and in doing so, generating a new temporary lattice on the fly. This new lattice consisting of the merged objects becomes the subject of the new tag cloud window, and the user is free to explore the desired objects without introducing concept broadening [6].

4 Conclusion

ConceptCloud 2.0 functions effectively as a data exploration and visualisation tool. A user study [1] of an earlier version of ConceptCloud showed the effectiveness of the tag cloud navigation for investigating a rich text-based data set, but another study is still required to determine the intuitiveness and ease of use of this latest iteration of the software.

In our testing it became apparent that the Google Maps API could be a limiting factor, since it adds marker objects to the map individually before being clustered. This process is highly memory intensive, and the web browser becomes unresponsive when exceeding 250 000 markers. With the previously discussed front-end and back-end optimisation implemented, these limitations were overcome. The ConceptCloud software itself proved to be highly scalable, processing a full 2.5 million object data set without any error or decrease in performance. With regard to future work, research is being done around abstracting

the map viewer, essentially a 2D Cartesian plane, to support other metric space visualisations.

References

1. Dunaiski, M., Greene, G.J., Fischer, B.: Exploratory search of academic publication and citation data using interactive tag cloud visualizations. Scientometrics **110**(3), 1539–1571 (Mar 2017). https://doi.org/10.1007/s11192-016-2236-3, https://doi.org/10.1007/s11192-016-2236-3
2. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer Science & Business Media (2012)
3. Greene, G.J., Fischer, B.: Interactive tag cloud visualization of software version control repositories. In: Software Visualization (VISSOFT), 2015 IEEE 3rd Working Conference on. pp. 56–65 (Sept 2015). https://doi.org/10.1109/VISSOFT.2015.7332415
4. Greene, G.J., Fischer, B.: Interactive tag cloud visualization of software version control repositories. In: Software Visualization (VISSOFT), 2015 IEEE 3rd Working Conference on. pp. 56–65 (Sept 2015). https://doi.org/10.1109/VISSOFT.2015.7332415
5. Greene, G.J., Fischer, B.: Conceptcloud: A tagcloud browser for software archives. In: Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. pp. 759–762. FSE 2014, ACM, New York, NY, USA (2014)
6. Greene, G.J., Fischer, B.: Single-focus broadening navigation in concept lattices. In: CDUD@CLA (2016)
7. Kaytoue, M., Kuznetsov, S.O., Macko, J., Meira, W., Napoli, A.: Mining biclusters of similar values with triadic concept analysis. arXiv preprint arXiv:1111.3270 (2011)
8. Kisilevich, S., Mansmann, F., Keim, D.: P-dbscan: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geotagged photos. In: Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research #38; Application. pp. 38:1–38:4. COM.Geo '10, ACM, New York, NY, USA (2010)