Online Appendix – Additional Technical Details and Discussion

To make the application web-based and to facilitate simple access and deployment on a variety of platforms, we chose to develop the prototype based on Adobe Flex technology. Flex is free and open source software that is extends Adobe's Flash player technology. Both the use and the API of Adobe Flash player are free. The flash player is also platform independent and is widely use in web applications. Flash based software can be embedded in any web page and is supported by all the major web browsers. Flex code can be also be use in Adobe's AIR to create Rich Client Application that does not depend on a web browser and runs as a native desktop application. An AIR based application can be deployed and updated via a simple web-based installer. The prototype (without the pilot data) including the source code and instructions for populating the application with your own data are available from our web site [URL will be provided once manuscript is published]. The web site provides an installer that will download and install the software on the user's machine. The software runs without modifications on Windows, Mac OSX and Linux platforms. The web page will also check if the software is already installed on the user's machine in which case it will inform the user if a new version is available.

Additional technical details about tag layout approaches:

We implemented several approaches for determining the layout of the tags in two-dimensional space. After the tags are laid out, the user can freely drag tags around the screen in order to manually fine tune the clustering, form hypothesis or resolve overlaps. The random layout maximizes use of screen space by distributing the tags uniformly on the screen but at the price of removing the clustering feature. We decided to avoid force-based layout algorithms, such as Spring Graph Layout, as they tend to converge slowly causing the tags to oscillate and distract the user.

We explored a Principal Component Analysis (PCA) based approach, in which we consider each tag as an *n*-dimensional vector, where *n* is the number of raw data items, with the *j*-th entry equal to 1 if the tag is associated with the *j*-th data item. PCA employs an orthogonal transformation to convert a set of observations of possibly correlated variables (the tags) into a set of values of uncorrelated variables called principle components. We use PCA to project the tags from the $\Re^n$ to $\Re^2$ (screen space) by projecting the tags on the first two principal components. Our aim was to use PCA to project the tags such that tags having an underlying association in some abstract sense (i.e. similar values along the two major principal components) would end up close to each other on the screen. This approach proved inadequate in two respects. Tags that are not strongly correlated with either of the two major principle components tend to cluster around the origin (small x,y values). This apparent clustering is misleading because it is due, not to the *existence* of strong correlations between the tags, but rather to their *lack* of correlation to something else, i.e. the two major principle components. The second problem stems from the inherent property of the PCA projection which positions tags that do have strong correlation to the two major principle components in a circular fashion far away from the center. Tags that have similar correlation to one principle component but do not have similar correlation to the second principle component end up far apart on the screen, e.g. the normalized points (1,0) and (1,1).

After several iterations of PCA-based layouts, we turned our attention to a Multi-Dimensional Scaling (MDS) approach, which is used more often in information visualization, and is based on the similarities (or dissimilarities) between the tags. MDS methods provide various projections from $\Re^n$ to $\Re^2$ that preserve some notion of distance between the points (tags). We use various distance functions between tags using Euclidean distance ("classical") and correlation functions (Pearson, Geometric, or Centered Geometric). The MDS layout performs well both with respect the clustering of the tags and with respect to the computational time required to calculate the tag coordinates.

Proposed overall architecture and dataflow for system implemented in production

A general schematic for a possible EpiCanvas system implementation is depicted in Figure 1. A *Data Repository Semantic* (DRS) is used to encapsulate a reference to a given database and the semantic associated with it. Numerous DRS can exist that represent disparate sources. The *Data Tagging* is an offline process that attaches tags to the raw data items in the data sources. The tagging is done by a collection of independent software tools (TagBots). A TagBot can define new tags, which are then stored in the Tag Library, and can reuse tags contributed by others. The Tag Library is part of a single knowledge base for EpiCanvas. A TagBot stores information about the tagged data in a many-to-many form (tag id, data item) in the EpiCanvas repository.

Users interact with the system via the EpiCanvas display. Multiple users can work independently using multiple EpiCanvas displays. When started, the EpiCanvas display interrogates the knowledge base and load the definitions of all the known types of tags. The display can then fetch and display tags for a given period of time.

The modular architecture separates the data sources, the tagging process, the knowledge base, the tagged data repository and the display. Numerous data sources can be used independently and be scanned by independent TagBots. There is no dependency on any particular data source or the TagBot. New data sources and new TagBots (which provide new semantics) can be added at any time independently of any other component in the system. There is only one Tagged Repository and one Knowledge Base, both of which provide fixed sources of information for the EpiCanvas display. *Multiple users can run independent displays and interact with the system.* The display in turn interacts only with the Tagged Repository and the Knowledge Base and is separated from any changes, additions or deletions of data sources and TagBots.

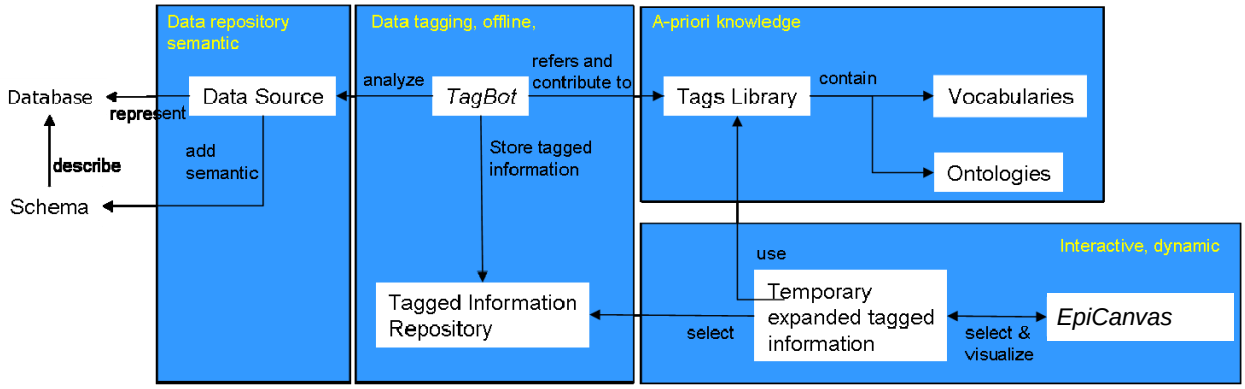

Figure 1. An overview of the various parts of a possible complete tagging system to support the EpiCanvas approach.