

Abstract for MMPIM

Armand Maree

May 23, 2016

Abstract

Mind Map Personal Information Manager (MMPIM) is a system that collects data from social media sites like Facebook and Gmail and interprets and processes that data then displays the result in the form of a mind map. It analyses the content contained in an email's subject and body, a Facebook post's body or the caption of a picture to determine the topic of this text. The topic is then attached to the mind map and other relevant topics and articles will spawn from that. Relevant topics can include friends and contacts that have the similar topics linked to them, people tagged in the current article and other events in your timeline that correspond to the same topic. MMPIM should also change it's context based on your current location and your Google Calendar. The context you are in should allow you to see information relevant to that context, like friends that have the same event in their current timeline or who are at your location. The root node starts out by showing the relevant topics in your life right now. Like if you received an email about cooking recently then a node labelled "Cooking" will be present. From this node photos and communication (posts, emails, messages, etc) nodes will spawn (that is related to cooking) that will show the most recent of both these nodes. Context changes should only occur upon request of the user. The system should realize that a context change can occur then prompt the user to change the context.

1 Back-end Functionality

In the back end, the system would have a polling service that polls the social media sites for new information about the users. If a polling thread receives new data from the platform they add this new data to a queue to be processed. Several worker threads dequeue data from the queue and uses the natural language processing to determine the topic(s) of the data. This data is then added to a database and the corresponding topic(s) as well. Note that since hashtags are used so frequently, any hashtag in the text should be added as a topic of its own. When a user requests their mind map, the system finds all relevant topics of their life based on frequency and recency. These topics are then returned to the user in the form of JSON objects. Once a user expands one of the nodes a request is sent to the server to retrieve all the relevant articles

corresponding to the topic to the user. These new data is then added to the mind map in the form of nodes. Relevant articles are found on the basis of your own events and the events of your friends.

2 Front-end Functionality

The user starts out with a root node that has relevant topics connected to it including a friends node. All these nodes can be expanded and reveal information related to that topic. The friends node shows contacts you recently communicated with or might want to communicate with now. Expanding one of these contacts will show the different platforms you can interact with this contact, including viewing a profile and sending a message/email. Other topics will expand into relevant articles and those can be expanded again and so forth. Upon expanding an article a user can then interact with that article (comment/reply) in a side panel that will be presented on the right of the screen.

3 Issues

- How many polling threads? One per platform or multiple per platform.
- What data does the polling threads store in the queue?
- Does the client have to manually refresh to see new data or does the system automatically refresh with the new data? Or maybe alert the user of an update.