Project ICS2205/2230
Web Intelligence

Charlie Abela, Joel Azzopardi

November 13, 2017

This document contains the details for the ICS 2205 project that will involve the discovery of features from a corpus of emails, and its visualisation. This project is meant to bring together all the components which have been dealt with during ICS 2205, namely web technologies, graph and network theory as well as textual analysis and text mining.

This is a group project and while discussions between groups are considered as healthy, the final deliverables need to be that produced by your own group and not plagiarised in any way. This project is marked out of 100 (however it is equivalent to 40% of the global mark for this unit).

The **deadline** for this project is **12:00pm Friday 12th January 2017**. Deliverables and attached plagiarism form must be uploaded on VLE. Projects submitted late will be penalised or may not be accepted.

1 Background

Nowadays, one can find various data sets that have been released to the public with the purpose of allowing people to discover and visualise interesting nuggets of information from these datasets.

For example, on 31st August 2015, the US State Department released a considerable number of emails that were the source of controversy for Hillary Clinton since she was using a non-government email server when she was secretary of state. One can download such emails from [1]. Moreover, one may also find various demos that visualise interesting knowledge nuggets extracted from this dataset[2].

[1] `https://www.kaggle.com/c/hillary-clinton-emails/data`
[2] `https://www.kaggle.com/c/hillary-clinton-emails/scripts`

As a further example of data visualisation, one may refer to the site MovieGalaxies[3] that provides insight into the world of movies by displaying the interactions between the actors participating in that movie, as a network. The visualisation used, distinguishes between the various actors and the importance of their roles within a movie, by displaying nodes with different colours and sizes. Furthermore, various metrics are computed, such as the betweenness centrality and degree for each node, and the clustering coefficient, diameter and path length for the whole network.

2 Specifications

In this project, you will be required to perform the automatic analysis of a smaller email corpus called the **ConThread-BC3 Corpus**[4]. This dataset consists of 40 conversation threads with a total of 261 emails. You will need to analyse the interactions between the different senders/recipients as well as the content of the emails themselves.

You will need to implement a web-page application that will display

i. a keyword-cloud based on keywords extracted from the emails.

ii. a graph of the senders/recipients network and allow for some analysis to be performed on the graph and the visualisation of the results.

3 Team Setup

You are to work in teams of 2 people each. You can decide with whom you want to team up. However, if you do not manage to team up yourselves, then we will form the teams, and such team setups will be final.

The work within each team has to be distributed fairly, and in the documentation you will need to describe how the work was distributed and who was responsible for which part of the project. The mark given to each team member is determined based on the quality of that member's contribution to the team's overall project. Marks assigned to different members of the same team *may* vary.

4 Deliverables

The following is information about the deliverables that you will need to present. Each component of this project is individually marked.

[3] `http://moviegalaxies.com/`

[4] `http://humanities.uva.nl/~dehghani/conthread-bc3-v1-0-conversation-threads-annotated-bc3-email-da`

The email dataset may be downloaded from [5]/

4.1 D1: Server Setup and Web Application Development

For this deliverable, you are expected to create a Web-based front-end for your application that integrates together deliverables D2 and D3, described further down.
You are expected to:

i. install a Web server on your machine to host your Web application.

ii. for the client-side, the part which handles the user interaction with the application and for displaying the keyword-cloud/network, it is recommended to use HTML for the structure of the Web page(s), CSS for styling and JavaScript for additional dynamic functionality or for implementing the algorithms. However, you could also use PHP on the server-side to implement the algorithms. The choice is yours.

This part of the project is being allocated **10 marks**.

4.2 D2: Text Analysis

This deliverable involves the extraction and creation of the keyword-cloud.

You are expected to:

i. Extract the text from each email.

ii. Perform lexical analyses to extract the separate words, and to fold them all to lower case.

iii. Use a standard stop-word list for English to filter out the stop words.

iv. Use an implementation of Porter's stammer to reduce terms to their stems (note that you may find a ready-made implementation provided that you reference its source).

v. Calculate term weights using TF.IDF. All emails between any 2 distinct senders/recipients should be considered as a single document. In this way, the important words that characterise a particular interaction will be given high weights.

vi. Use the highest-weighted $n\%$ of the terms to build the keyword-cloud. This keyword-cloud will show what concepts the 2 senders/recipients generally talk about. n can be determined arbitrarily so that the keyword-cloud does not contain neither too much nor too few words.

[5] `https://www.cs.ubc.ca/cs-research/lci/research-groups/natural-language-processing/bc3/download.html`

vii. The TF.IDF weights should also be used to build keyword-clouds for each correspondent. This keyword-cloud will show what a correspondent generally talks about to all other correspondent overall.

This part of the project is being allocated **40 marks**.

4.3 D3: Graph Analysis and Visualisation

This component focuses on the analysis and visualisation of the email corpus as a graph where the nodes represent the correspondents (senders/recipients), and each edge represents the correspondence between any 2 correspondents.

You are expected to:

i. transform the provided dataset into a suitable representation for analysis and visualisation: typical representations include, GraphML[6], Gexf[7] and JSON[8]. **Note** that some third party libraries available to create the visualisation work with a subset of such representations;

ii. compute the following graph metrics:

   (a) number of nodes and edges in the network;

   (b) diamater;

   (c) average path length.

iii. provide a visualisation that shows the following features:

   (a) The correspondents' activity – the node size should be proportional to the number of emails sent/received by the corresponding sender/recipient. In this way, one can immediately recognize the most active correspondents.

   (b) The level of interaction between any 2 correspondents – the edge width should be indicative of the number of emails passed between the correspondents represented by the surrounding edges.

iv. link the visualisation of the network with the keyword-clouds from D2 above. The user should be able to interact with the nodes and edges (e.g. by clicking on them) to open the keyword cloud that is associated with the particular node/edge.

For this part of the project, **40 marks** are being allocated.

[6] `http://graphml.graphdrawing.org/`
[7] `http://gexf.net/format/`
[8] `http://www.json.org/`

4.4 D4: Documentation

This deliverable must be clearly written, marked up (figures and tables) and checked for spelling and grammar

This document needs to include a **one-pager** with *future work*. In this one pager provide answers to the following questions (not necessarily in the enlisted order of the questions):

i. How can this work be extended? Provide 2 possible improvements.

ii. Where and how do you envision the use of this work?

The rest of the documentation must include:

i. Descriptions of all the project's components, including aspects related to their design and implementation.

ii. The use of any third party libraries/tools has to be documented.

Moreover, the documentation must contain a clear statement of which parts were completed, and which not.

Last but not least, the documentation should contain a brief description of how the work was distributed amongst the different members of the teams.

The complete deliverable will be a bounded document, whose length must not exceed 20 pages.

The total marks allocated for this part of the project is **10 marks**.

4.5 Possible Resources

Find below a list of suggested libraries (and related information) that can be used to handle this component:

- *D3.js:*[9]: is a JavaScript library for manipulating documents based on data.
- *Javascript InfoVis Toolkit (JIT):*[10] provides tools for creating Interactive Data Visualizations for the Web.
- *Sigma.js:*[11] is an open-source lightweight JavaScript library to draw graphs, using the HTML canvas element. The issue here is that

[9] `http://d3js.org/`
[10] `http://philogb.github.io/jit/index.html`
[11] `http://sigmajs.org/`

Sigma.js requires the data to be in particular formats, such as for example the **gdf** format, which is built like a coma separated file (CSV), as well as the **gefx** through which its possible to combine with toolkits such as *Gephi* [12].

- *WordCloud2.js:*[13] is a JavaScript library that can generate and display keyword-clouds.

4.6 Summary of deliverables

D1: Server Setup and Web Application Development	10 marks
D2: Text Analysis	40 marks
D3: Graph Analysis and Visualisation	40 marks
D4: Documentation	10 marks

You will need to submit the following on VLE:

i. A PDF file containing D4 which also includes a plagiarism form, duly filled-in

ii. A zip or archive file containing all the deliverables.

5 Final Remarks

Final suggestion: if you have difficulties do not hesitate to contact us.
Good luck!!

[12] `https://gephi.github.io/`
[13] `http://timdream.org/wordcloud2.js`