Autonomic approach to information discovery in crowd sourced data

Final Report

CM0343 - Individual Project

Abstract. This report documents progression from the the interim report and provides conclusions on experimenting with mechanisms to facilitate autonomous information discovery. This includes finalised detailing of the: purpose, approach, design choices and implementation of the project components. Evaluations are made on the approaches, design choices and implementations of these components in terms of suitability and satisfying the initial plan aims and objectives. Concluding with reflections on the main contributions of the project towards suggesting autonomous capabilities for information discovery; future work and uses relevant to each project component individually and finally a reflection on my personal learning.

Student Number: 1029310
Student Name: Liam Turner
Lecturer: Dr S Allen, Dr J Shao

Date: May 1, 2013
Hours Spent on this Exercise: 400
Special Provision:

Acknowledgements

With thanks to Dr S. Allen as project supervisor.
With thanks to the experiment participants for providing qualitative feedback.

Table of Contents

List of Figures

1 Introduction

The purpose of this report is to document progression of the project from the interim report towards finalised details of approaches, design, implementation and experimentation. Additionally to also provide a detailed evaluation of the these finalised details and discuss the main contributions of the project. The structure of the report as a whole and for each relevant section will follow the flow of: a brief outline of the contents; followed by specific details of a given topic and finally an evaluation or reflection on that given topic. Discussion will be split up where components of the project are related but do not have a strict association, this being the core tagging algorithm, the API platform and the experiments performed. The scope of the project does not cover all possible implementations of each component and therefore any conclusions regarding autonomous information discovery from the group of implementations used cannot be a complete proof. For example the concepts of: emulating human cognition for creating tags; using an API for deployment and using ranking and interfaced semantic content as information discovery processes are a subset of possible methods. However, by detailing and evaluating each method individually with results from experiments, suggestions can be made regarding the effect of specific approach and design aspects; which can indicate the suitability of the concepts behind each method. The contributions of the project are subsequently suggestions on the feasibility of the concepts used for autonomous information discovery and direction towards future work.

The initial sections will supplement the content of the interim report to produce a finalised outline of the project deliverables. This includes detailing any missing aspects, changes made and extended work to the: scope, approach, design, implementation and tools and methods used. Focus will then concentrate on evaluating the project deliverables against requirements and the results from performing testing and experiments. Followed by evaluating the adopted project management methodologies of planning and control. The final sections of the report will comprise of: ideas for future work; an overall conclusion to the main contributions of the project and a reflection on my personal learning as a result of undertaking this project.

2 Project Details and Deliverables

2.1 Project Description

The project focuses on experimental research towards solving the problem of enabling information discovery in large crowd-sourced textual data repositories with a method which is autonomous in both use and deployment. The core scope towards a solution was experimenting with tagging text by using only what could be deduced from the words using natural language processing; in order to emulate human cognitive input without any direct human influence. Following this, to encase a solution into a standard driven structure for autonomous access and to test the suitability in information discovery scenarios. For example, with the tags produced as either the base or a part of, a further process such as ranking content for desirability.

2.2 Core Tagging Mechanism

This section aims to provide finalised details of the core tagging algorithm. As much information regarding this was documented in the interim report, the content is aimed to supplement the content of the interim report and focuses on: briefly justifying the reasoning of emulating human cognition; a recap and expanded justifications of the approach; tweaks to the algorithm and finalised details of the tools and methods used to implement and test. Any content not covered in this section should be assumed still applicable from details in the interim report.

2.2.1 Approach and Scope

The inspiration of the approach comes from combining features of trained tagging models from machine-learning and the human driven tagging convention of hash tagging (Twitter, Inc 2013c). Tagging models which are trained using machine learning is a method of classifying text computationally. However, the method is arguably not autonomous when used with crowd-sourced repositories such as Twitter due to the large volume of data moving at a fast pace; which causes degradation in accuracy when the data used to train the model becomes outdated and decreasingly relevant. Hash tagging is a human driven tagging system adopted on a subset of crowd-sourced text repositories, typically social networks such as Twitter. The concept stems from how humans naturally wish this data to be tagged; a method which could do this computationally is therefore appropriate. A consideration to take into account is that the thought process that humans take in tagging data may not be universal and individuals may tag text differently.

The approach taken in this project focuses on investigating into deducing word classifications (e.g. Noun) computationally using natural language processing methods. These word classifications would be determined from a machine learning trained model, the potential for degradation in accuracy would be reduced as the classifications of words in a language are unlikely to change. It should be considered that words may be appended to a language, but this is at a much slower rate than the flow of data on Twitter. This provides increased potential for autonomous use in comparison to trained models based upon a set of Tweets. The intention was then to examine the extent in which the set of tags produced in this manner matched tags produced by humans. It was hypothesised that if a set of word classification patterns could be determined which matched those which humans deduce naturally, then the information discovery processes it would influence would also be more natural. This would then produce more desirable content than enforcing that tags follow some convention (e.g. must be a person, object, location or event etc).

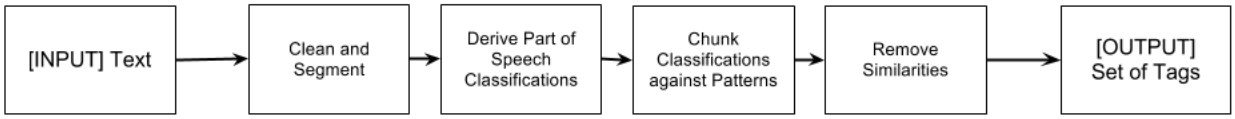
The approach taken with word classifications was to use a model trained using the part-of-speech tags of a renowned text corpus; where the word classifications could be assumed as correct. The shortfall of potential incompleteness with this was aimed to be overcome by falling back to a default classification (e.g.

noun), if another part-of-speech tag denoting otherwise could not be found. Finally, a source for providing test data for evaluation was needed and Twitter was chosen due to it's large amount of data and popularity in other social computing research [1].

2.2.2 Design and Algorithms

The majority of the design for the tagging algorithm has not changed from the interim report. The algorithm structure used remained the prototype 'Brown Parser' algorithm:

Fig. 1: Expanded overview of experiment environment



Minor tweaks and changes have been implemented, namely extensions to the pattern set for part of speech chunking; to include verbs and proper nouns as tags. An expanded view with pseudo-code of the final tagging algorithm can be found in the Appendix, Figure 6. An additional note regarding design choices is that the natural language processing methods used in the tagging algorithm are enhancing programming syntax specific. This enables the algorithm to be implemented in any environment, enhancing the flexibility of its use. Furthermore, the methods used are a collection of existing algorithms chosen for their stability and overall acceptance in other natural language processing works. Deriving my own natural language processing algorithms was seen as not being cost-effective in terms of time and resources required. These algorithms would form a stable foundation to focus on building the novel aspects of the human emulating tagging algorithm on.

2.3 Applications of the Tagging Mechanism in Information Discovery

This section aims to provide details regarding applying the core tagging algorithm into an autonomous platform to then build information discovery applications on. The beginning of this shift in focus was documented in the interim report, this section aims to supplement this by documenting the final details of approach, design, implementation and tools and methods used.

2.3.1 Purpose and Approach

Following the successfulness of the tag precision in the prototype tests of the tagging algorithm (documented in the interim report), focused shifted towards trying to examine the extent of the suitability of the algorithm in information discovery scenarios; i.e. as part of information discovery applications. A system was implemented to enable delivery of the tags and content based on the tags (e.g. ranking). Methods for interacting with this system and the output it produced were wrapped with standardised web service protocols to create an application layer interface (API). The approach was taken that this API would serve as a scalable development platform, with the scope of its use in this project as a test bed environment for the numerous information discovery applications.

The potential use of extracted tags from text (e.g. a Tweet) was broad; therefore the approach taken was to build applications where the functionality can be described as information discovery processes. Tags were to be used to influence the output of these processes; in this case, influence Tweet ranking and interfacing external content APIs, such as: news, video and image content. It was envisaged that this would

[1] Based upon a Google Scholar search result of 1,270 papers with 'Twitter' in the title, published in 2012. [Accessed: 27 April 2012]

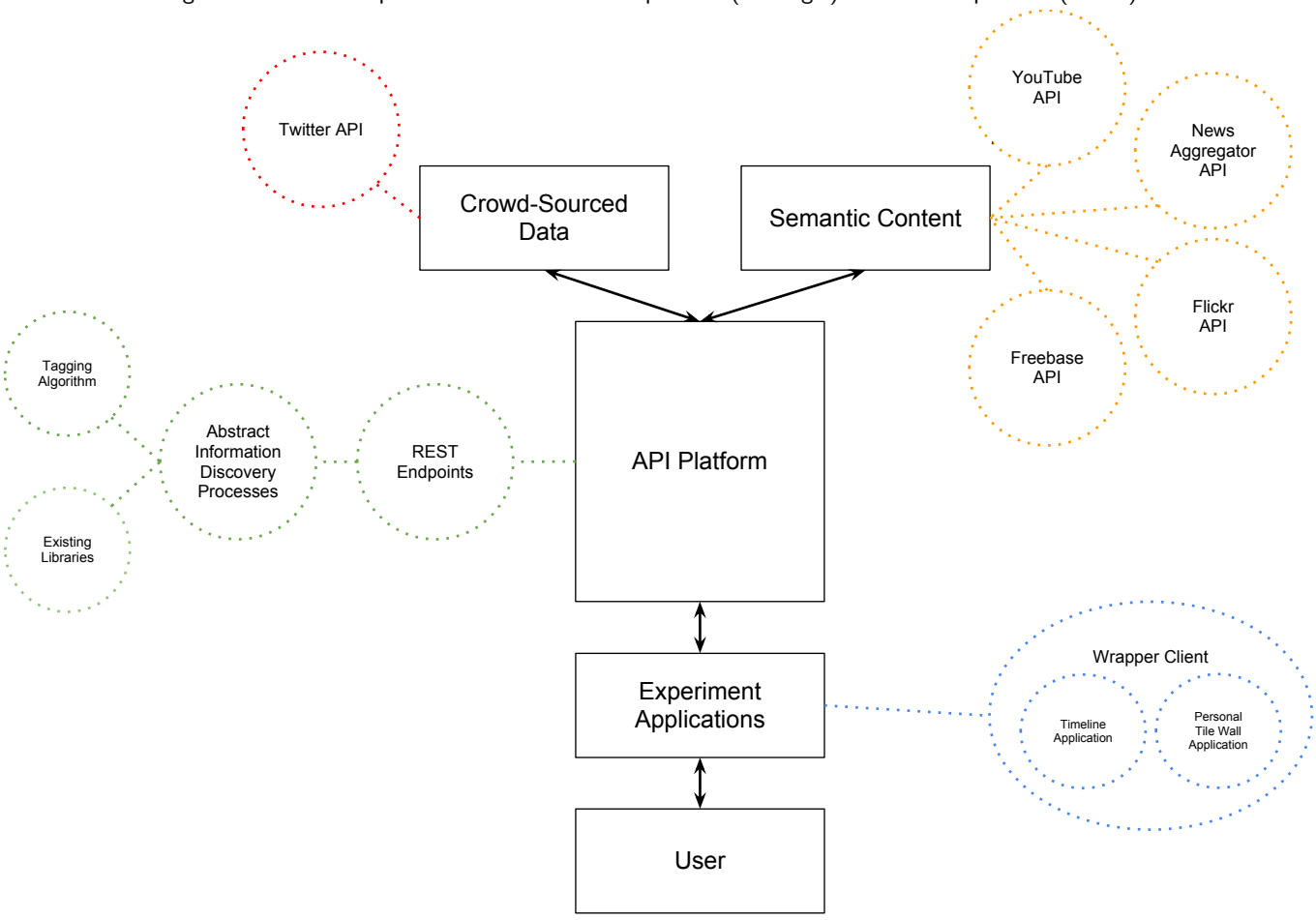
allow for the scalability of the API to be demonstrated with the individual applications building upon it with more specific implementations. These implementations would integrate functionality of the API with their own independent application logic much like most '3rd party' applications.

By developing different types of applications using the API with as many tests as time would allow; the aim was to increase the prospective conclusiveness in the evaluation of the suitability of using an API as an autonomous deployment method for tag influenced information discovery processes. This approach would also provide an opportunity to satisfy less applicable aims and objectives from the initial plan as a result of changed deliverables. Most notably those regarding the creation of a consumer focused system following a data backbone >API >client architecture; by mimicking consumer focus in the applications built on the experiment platform.

2.3.2 API Platform and Experiment Application Design

The design of the experiment environment remains the same as that initially designed in the interim report and follows a structure of four parts: A data backbone for test data (Twitter API); several external API services to pull content of a variety of information mediums (see Appendix, Figure 7); the API platform and client applications for users to use for information discovery. A visualised overview of the environment can be found below, Figure 2.

Fig. 2: Overview of experiment environment components (rectangle) and sub-components (circles).



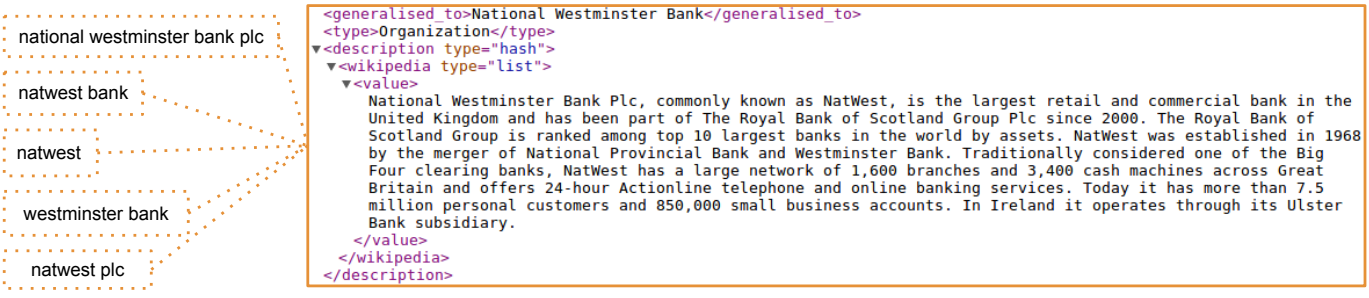
These components would be loosely coupled with minimal specific dependencies between each, for example, a different test data API to Twitter could be integrated without affecting existing implementation or an applications use of the platform would not dictate use by another application.The decision was taken to implement abstract implementations of the complex processes (ranking and endpoint interfacing) as endpoints of the API so that any configuration was not specific to any application built on top of it, see Appendix Figures 14 and 15.

A drawback of connecting to an external APIs is that they require specific implementation as the input parameters and output structure vary. For example, the HTTP protocols of the REST API architecture and syntax for input/output are standardised, however the structure of the input and output content is not standardised. This forces the encoding and decoding of input and output to have explicit implementations for each external API used. To minimise this issue, the implementations were designed in the API platform to follow a strict inheritance structure for methods and variables; limiting specific implementations of an external API to inside generic methods. This produces a translation effect, where once an explicit implementation is defined in the API platform, the methods of interacting with it are the same for interacting with any API under the same inheritance parent nodes (object-oriented super types).

The infrastructure of the API platform would be different internally and externally to facilitate these features. Internally the API would follow conventions of a white-box framework, heavily object-oriented inheritance based in a tree structure. This is to allow for modularity and subsequent expansion of branches (e.g. ranking, semantic information and authentication) as well as leaf-nodes for individual implementations (e.g. test data backbones (Twitter), semantic interfacing APIs (YouTube, Flickr etc). Externally, from the perspective of an application building upon the API, the API would resemble more of a black-box structure, similar to a library of callable methods, but language independent aside from HTTP protocols.

In terms of the external APIs used in the project, those proposed in the interim report remain largely unchanged. The changes were the choices of APIs to provide semantic content with focus honed in on four core content mediums: entity descriptions, images, videos and news stories. One API was chosen for each content medium and the choice of service was based upon: the size of the user base; flexibility in search parameters and comprising of crowd-sourced data where possible. These choices were: the YouTube API for videos; Flickr API for Images; a community made Yahoo Pipe for aggregated news and the Freebase API for entity descriptions. Further details of the uses and quotas of these APIs can be found in the Appendix, Figure 10. The aggregated news pipe was chosen as it sourced from numerous different global news needs and enabled both 'AND' and 'OR' to be used between tags for flexibility in result relevance. The Freebase API was chosen due to the extensiveness of its database, having crowd-sourced data contributions and being flexible with input parameters for tags through generalisations and alias' (see below, Figure 3).

Fig. 3: Example visualisation of different tags referencing the same entity using the Freebase API.



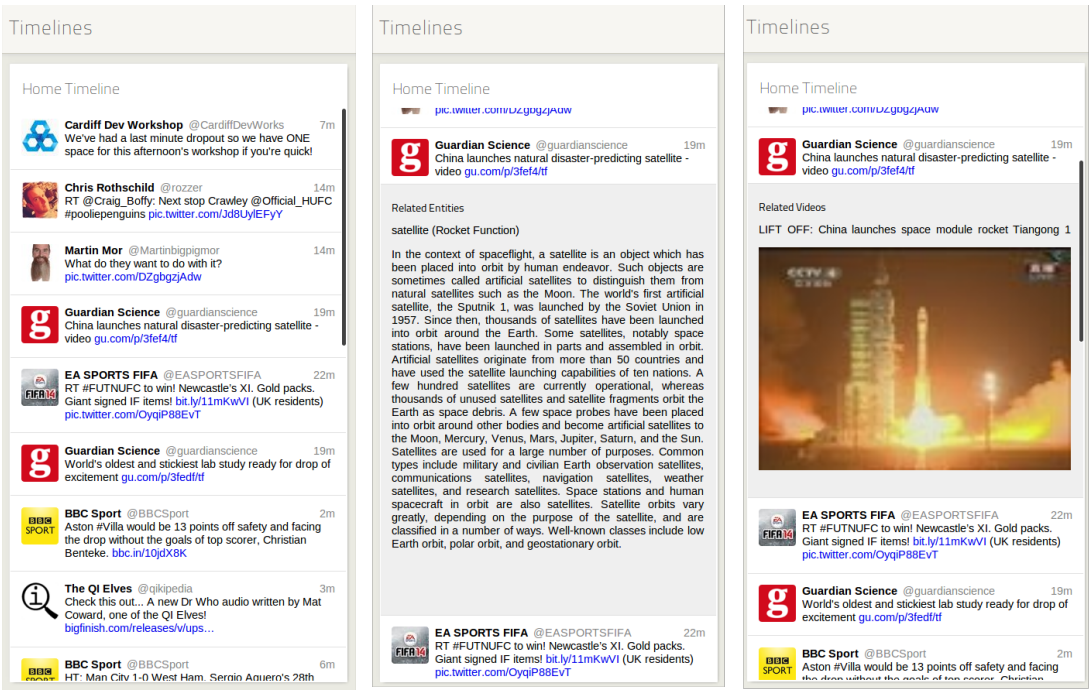
The use of each external API was controlled through a white-list of acceptable endpoints (see Appendix, Figure 7). The white-list can be changed at any time for any API used but for the scope of the project these white lists were populated with endpoints that provide the primary medium associated with it. For example, search endpoints rather than account operations. An exception to this was authentication endpoints of Twitter as per-user authentication was needed to retrieve personal Tweets. For the scope of the project, Twitter authentication was designed to be the only personal data service implemented. However the design of the authentication internally allows for authentication to be scalable to other services easily. For example if the platform was to be used outside of the project scope and another social network was implemented for the personal text data; all authentication credentials accounts would be tied together so that signing in with one would allow for access to the other, if credentials for both were registered at some point (see visualisation in the Appendix, Figure 16. Similarly a user could be created without any attached accounts and these could be registered when desired. This capability could also be extended for the semantic APIs, for example registering a Flickr or YouTube account would allow for personalised content to retrieved in addition to public content from searches.

Finally in terms of experiment applications, two independent applications were designed to provide the capability of testing the effectiveness of tagging influenced information discovery processes. This design choice was taken so that likewise to experimenting with using multiple complex processes, multiple applications of these would further reduce the possibility of biased or narrow results in evaluation. These two applications were independent but for the purpose of the experiments were placed together under a single user interface wrapper for usability; more details of which can be found in the Appendix, figure 17.

Application 1

The first application is primarily a Tweet ranking application which displays Tweets in a time line ranked by multiple interchangeable values, including calculated: interest, integrity and time; specific details of the calculations are documented in the interim report. Related semantic information would be provided for each individual Tweet by clicking the Tweet area.

Fig. 4: Screenshot of the time line application with demonstration ranked home time line data and expandable semantic content.

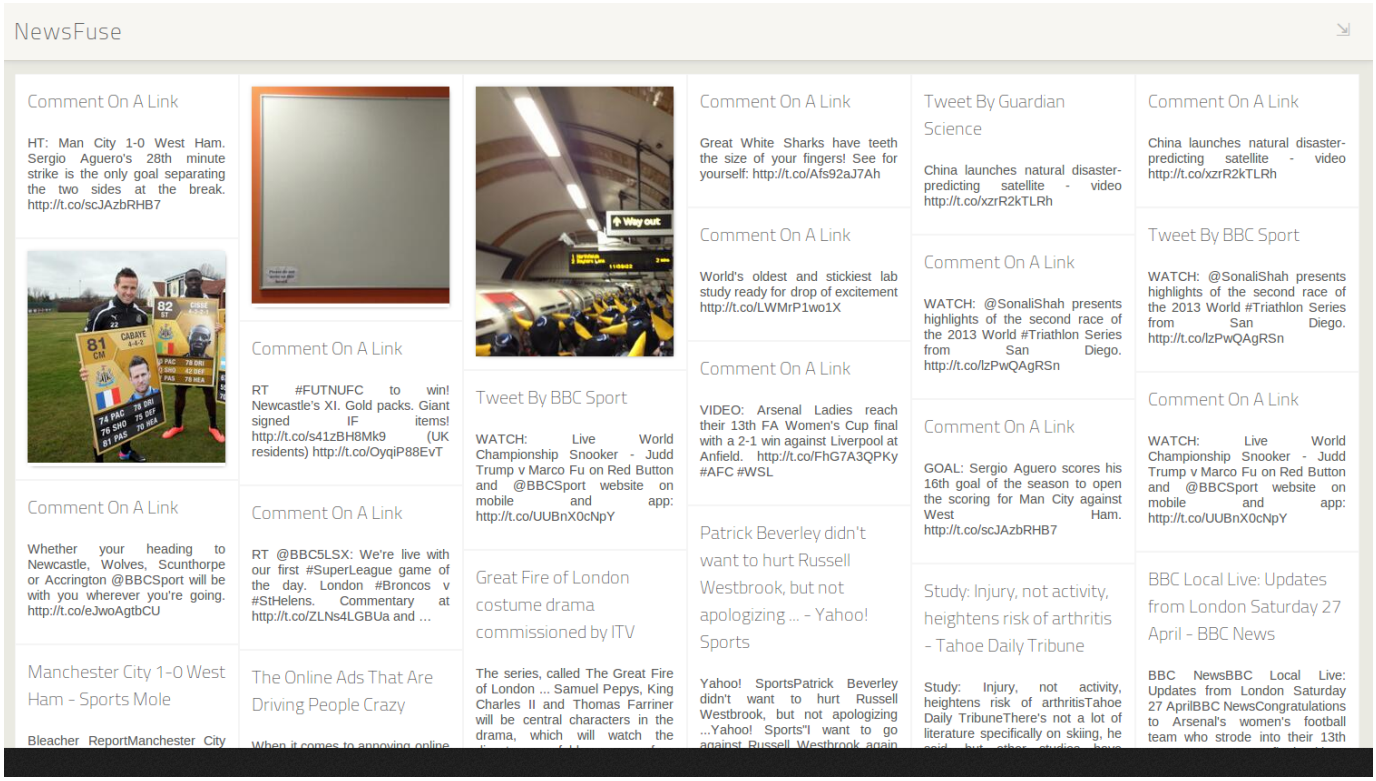


Further details of this application can be found in the Appendices, Figure 18; a visual demonstration of the implementation can be found above, Figure 4.

Application 2

The second application aimed to merge Tweet content and semantic content and produce a tiled wall of recommended information that the user would find interesting and relevant. These tiles are not ranked and were instead derived from influence from the user's social graph of Twitter connections. This would differ from the first application by examining the successfulness of the information discovery based solely on the user and their Twitter connections, rather than on the relevance of related semantic content for a Tweet. This approach was influenced from comments from the prototype human keyword tagging process documented in the interim report, where some participants felt instinctively pressured into finding information to satisfy the experiment instructions. The potential for this effect to occur arguably exists in the task of commenting on the relevancy of relevant semantic content of the first application. This second application would reduce the potential for this in effecting the results by having no similarities to the interface of Twitter and no task on commenting on specific pieces of content. More comprehensive detail regarding this applications used for experimentation can be found in the Appendices, Figure 19; a visual demonstration of the implementation can be found below, Figure 5.

Fig. 5: Screenshot of the personalised tile wall application with a demonstration tile data.



A final design choice of the applications (as well as the wrapper) was that they were designed to be as flexible as possible in terms of user interface to match consistency in autonomous deployment. This includes having adaptive designs based upon screen pixel width for use on mobile devices and not using images in the design with graphical elements fully rendered in the browser, for network traffic efficiency and scalable sizes in the adaptive design.

2.3.3 Tools and Methods Used

The policy was taken that if implementations existed for a particular relevant standard or aesthetic feature license-free, then it would be used so that focus could be concentrated on delivering novel features and effective evaluation. These existing implementations would also be more stable in terms of testing in addition to saving unnecessary development time.

Creating an autonomous API platform for the experiment environment was achieved by using existing standards for structure and data encoding. This includes using the REST software architecture to structure the platform (Fielding 2000); likewise to the external APIs that this platform would use to gather text and semantic data from. This would provide both internal benefit (simpler implementation than alternatives such as SOAP) and externally, by having the same style of interaction (using HTTP GET/POST requests) as other APIs that are most widely used (DuVander 2010). The encoding formats for output encoding were JSON or XML.

The tools and methods used to implement this structure: Python, the Django framework (Django Software Foundation 2012) and the TastyPie framework (Lindsley 2012) remain the same as those planned in the interim report. These technologies remained appropriate throughout implementation, particularly with the seamless integration of: pure Python libraries (see Appendix, Figure 9); HTTP and REST protocols; POSTGRESQL database data (by means of object relational models (ORM)) and output from external APIs (non-ORM data sources). The model-view-controller methodology (Reenskaug, 1979) provided a modular means of development for suitable: project management scheduling and testing. These features could have been achieved with a combination of other technologies, however, those chosen were not a disadvantage and provided the ability to satisfy the project requirements, continual development and my personal aims and objectives from the initial plan.

However, the tools and methods used for the client applications have changed from that stated in the interim report. The same requirements were still satisfied, however, the technologies to accomplish this changed from using the intended backbone.js framework (Ashkenas 2012) to a combination of JavaScript libraries (see Appendices Figure 8). This was due to issues in integrating the platform output into the data model structure of backbone.js which caused the time plan allocations for the creation of these applications to be stretched. The change was appropriate as it provided a means of satisfying the requirements within the time plan allocations (such as AJAX interaction between the applications and the API platform) with only the internal layout of the application logic changing. In addition to the planned application tools and methods stated in the interim report, additional technologies were deemed necessary following design of the lower level application logic. These include HTML5 features (W3C 2013a) such as 'localStorage' for the ability to store and retrieve data securely across the experiment environment application wrapper (e.g. Twitter credentials). Other open-source JavaScript libraries were used for some design elements of the client applications (see Appendix Figure 8); this was appropriate as recreating these for aesthetic features was not seen as being cost effective (as per the stated existing library policy).

3 Testing and Evaluation of the Project Deliverables

The purpose of this section is to perform a critical appraisal on the project deliverables. Firstly, an evaluation will be performed to review each component in regards to features (i.e. What can it do, what it can't do) and the technical challenges overcome. Following this, the methodologies undertaken to test the functionality and stability will be discussed. Finally, details will be provided of the experiments performed to analyse the effectiveness of the deliverable components in terms of suitability for information discovery and autonomous capabilities.

3.1 Evaluation of the Core Tagging Mechanism

This section provides a review of the tagging algorithm against the relevant aims and objectives of the initial plan. Additionally, evaluations will be made on the design choices, technical challenges and known limitations of the algorithm in terms of functionality.

3.1.1 Review against Relevant Requirements

Due to the project undergoing several shifts in deliverable focus, the aims and objectives outlined in the initial plan have become less directly applicable. However, as each iteration in the project deliverables either integrated or was influenced by the previous, comparisons can be made to an extent. In regards to the tagging algorithm, the following objective from the initial plan is relevant:

"At a conceptual level, the primary aim and objective of the project is to create a system capable of providing users with a personalised and sustainable means to quickly gain knowledge and discover information based upon real-time collated data."

The tagging algorithm has autonomous capabilities so that it could be sustainable in any environment, including processing data in real-time, i.e. recent Tweets from the Twitter API. It is arguable that the tagging algorithm partially satisfies the aim of an information discovery system by being an underlying mechanism for doing so. With the focus shifted away from a single information discovery system, the algorithm can and has been used in processing such as ranking and external API interfacing to create multiple information discovery systems for experiments, those of which are comparable with the objective.

3.1.2 Reflection on Design Choices and Tools Used

On reflection, the primary approach towards design of only using what could be deduced from computational linguistics to create tags was appropriate. The outcome of doing so was unknown (thus the purpose of the project), however, the concepts of tagging text and natural language processing of linguistics are established areas of research. By building on existing research of these areas as part of the approach to autonomous information discovery, the suitability was somewhat predictable from the uses in other works. The advantage of this was that the scope was able to planned relative to a time plan with less risk of not meeting the aims and objectives than a completely new method of creating tags. It is my opinion that this also gave the scope a level of completeness which helped drive the focus of balancing milestones with deliverables.

The design choices of the natural language processing algorithms and components (such as the training data corpus) used were taken with my perception of applicability. These algorithms were deemed suitable as they were able to achieve the objectives in creating a tagging algorithm based upon linguistics. It should be noted that comparisons were not made due to time restrictions between similar algorithms (e.g. chinking and chunking) and therefore it can be suggested that the chosen natural language processing algorithms

are justifiably suitable but potentially not optimal. The tool used for implementation of the natural language processing algorithms (the natural language tool kit Python library (NLTK Project 2012)) is suitable as the choice of algorithms were implementation and language independent. This enabled the choice for implementation to be focused on parameters such as: speed to further satisfy the initial plan aims and objectives; stability to reduce testing workload and ease of integration. The only potential limitation in the implementation choice was there was the extent of parameter configuration in the algorithms. For example, the amount of text corpora that were available and compatible for part of speech classification with the library was limited, which reduced the choices available for the training data. This resulted in a suitable but potentially incomplete choice with the Brown corpus, due to its age in comparison to the intended test data (Tweets).

Lastly, as the prototype classifier was deemed fit for full implementation, the decision was taken to the compile the trained part of speech classifier for speed efficiency. This is justifiable within the project scope, however, if experimentations into other corpora were to be performed then the time taken to recompile may affect this justification. Overall, I believe that the use of Python and the NLTK (NLTK Project 2012) is justifiable as a suitable implementation tool, however, future work could be performed in experimenting with configuration to fully assess optimal use.

3.1.3 Unforeseen Development Challenges

The use of an existing well renowned Python library to provide stable implementation of the relevant natural language processing methods provided straightforward development with minimal stability risks. Unforeseen technical challenges in the development were minor as a result and any that occurred revolved around configuring the library calls at the necessary times with the necessary configuration. The reasons for this were a lack of previous experience in using the library. Any other unforeseen development challenges revolved around boosting efficiency in the implementation, such as compiling the part of speech classifier into a static model so that the model did not have to be trained against the text corpus each time the algorithm script was executed.

3.1.4 Known Limitations and Persisting Issues

Outside of tests in experimentations, unit tests were initially performed exclusively on the algorithm to test the functionality and stability. Overall the algorithm was deemed suitable for use in information discovery experimentation, however, this testing highlighted limitations in its current state and issues which effect the ability to be autonomous. These issues include the incapability to process irregularly formatted text with the correct interpretation. More specifically, problems such as irregular capitalisation (e.g. 'A Tweet With A Capitalised Character At The Start Of Each Word Is Not Necessarily A Set Of Nouns' or a company which uses an adjective as it's name should be a tag) and spelling errors potentially being classified incorrectly which causes incorrect tags.

The exact impact this would have is unpredictable as it would correlate to the occurrence of these issues in the input data. However, in the context of this project, crowd-sourced data from sources such as Twitter are arguably likely to have these issues if the data was taken from a random set of users. This is due to a large component of Twitter being casual social interaction where formatting of content is unregulated. With this project experimenting with personalised information discovery it is not feasible to predict with a close degree of accuracy the language structure of the Tweets being tagged for a particular user. Therefore it was decided that this be explored practically through feedback in the later experiments.

Another potential limitation the use of the Brown corpus (Francis and Kucera 1979) for part of speech training data. Any negative conclusions of results may be as a result of the training data in this corpus

and not necessarily the algorithm itself. Measures were taken to reduce this possibility, these being using a renown corpus and performing initial tests against human derived keywords (as detailed in the interim report). Although any negative effects from the used corpus were suggested to be minimal to none, it was not proved and therefore should not be completely ignored in reviewing the experiment conclusions.

Finally, it was observed following the results of the human against algorithm precision/recall tests (detailed in the interim report) that some tags were similar but with differences in words perceived as relevant either side of a core word. For example, given a string with words represented by W X Y Z, one person may tag 'X Y' as a tag, another 'W X Y' or 'X Y Z'; with word 'X' mutually agreed as part of a tag. Project deliverable focus shifted from exploring this occurrence further, but it could be suggested that the patterns used to computationally determine a tag may or may not be complete with this remaining unexplored. It could equally be said that having a complete set of matching patterns would reduce the recall in computed tags and that an incomplete set which computes a subset of similar variations (i.e. only 'X Y' and 'X Y Z' would produce better performance. Therefore the impact of this issue cannot be determined but could be explored in future work.

3.2 Evaluating the Application of the Tagging Solution

This section provides a review of the API platform and the applications built on top of it against the relevant aims and objectives of the initial plan. Additionally, evaluations will be made on the design choices, technical challenges and known limitations of these components in terms of functionality.

3.2.1 Review against Relevant Requirements

Likewise to reviewing the tagging algorithm, direct comparisons can not be made between the final project deliverables and the aims and objectives outlined in the initial plan. However, three objectives are relevant to the environment created for the information discovery experiments. The first two being:

"Develop a web service platform as an API for developers that would produce pages of semantic information from a data API backbone(s); with a similar concept to the Google Maps API, a platform on top of the raw data. Be flexible in terms of API endpoints and standards to be compatible for use with as many technologies as possible in client applications. Be sustainable for deprecation or introduction of new data backbones."

"Assess the suitability of building an API centric system revolving around real-time use of external services with text processing; including any notable advantages and disadvantages."

The API platform features of the environment used for conducting the experiments satisfies the criteria of these objectives overall. It also has the fundamentals for use as a standalone web service by following REST architecture standard; i.e. it could be deployed in the manner of this objective with minimal infrastructure changes. The endpoints of the API extend beyond semantic information and returns both original data from external semantic sources as well as additional data derived from it, see Appendix Figures 10 through 15. This coupled with modularised endpoints, where each operation of a process to reach the final output is also made available where possible through a separate endpoint or input parameter, facilitates an open platform to build on top. The flexibility of endpoints is further enhanced by the ability to chain together data request endpoints if that data is to be the input of an endpoint parameter; an example request which reduces three requests into one could be: rank Tweets from my home time-line using the keyword precision of my time-line, a global search for term T and against the set of terms 'X, Y and Z'.

The output of the platform can be in either XML or JSON format with the ability to expand or deprecate output formats interdependently. As the internal structure of the API is inheritance based, new endpoints, authentication methods (sign in via service X), data backbones (e.g. Twitter) and semantic data sources (e.g. news, videos) can be introduced and expanded with minimal cost in development resources. Instances of connections to these data sources are created by an implementation of the factory object oriented design pattern. The data source for test data was limited to Twitter in this project, however, the combination of inheritance and composition (through the factory design pattern) would allow for any other REST API (Fielding 2000) to be implemented using the same templates. This could include other social networks or crowd-sourced data repositories of other mediums, such as the YouTube API and then implement logic for enabling video ranking as well as Tweet ranking.

However, the inability to adapt automatically and be sustainable for deprecation of any external data source is arguably a shortfall which prevents the platform being fully autonomous. As the libraries used are only applicable to the current URL set of that service, if these were to change, connection to the relevant endpoints would fail. The implementation is designed to fail gracefully by not stopping execution of any other endpoints in the request unless they rely on the irretrievable data. Additionally, valid output in the chosen format is always returned to state a failure has occurred so this can be handled by the application using it. On reflection the implementation of backup services (such as Instagram as a photo backup for Flickr) wherever would provide better fall-backs if deployed as a standalone web service beyond the this project. Ultimately, in terms of use as a test environment for the experiments, I believe that the current state provided the necessary suitability to undertake at least the chosen experiments.

The third relevant objective from the initial plan is:

"Provide functionality for a measurable means of sustainable or seamless information discovery such as: attempting to predict desired information by integrating pre-emptive search aspects with related insights. In addition, use the authenticated user's social graph to produce recommendations of similar interests; based upon what is popular with friends."

This objective is related to the applications of the experiment environment. The applications build upon the API platform with independent application logic which provide information discovery features relevant to those in this objective. For example, the time-line application aims to predict interesting Tweets containing desirable information and place these at the top of the feed. Additionally it provides features relevant to pre-emptive search by enabling in-line: entity descriptions, images, videos and news stories relevant to the Tweet content; i.e. removing the time and actions required of using other services such as search engines to retrieve this information. Furthermore, tags extracted from Twitter endpoints related to the applications user's social graph can and were used in the ranking of Tweets in the time line used in the experiment instances of this application. The cosine similarity between Tweets is a variable used in the ranking calculation and provides the ability for Tweets with popular content to rise in the ranking. The other personalised tile wall application is strongly focused on social graph information discovery, where the content is based upon Tweets from the user's personal and home time lines. Both applications have the ability to provide functionality sustainably and seamlessly adapt input data (e.g. any Twitter endpoint that returns a list of Tweets) without changes to the user experience; a requirement necessary for both valid experimenting and possible standalone deployment for consumers.

Overall it can be suggested that the objective is therefore conformed to, however, it is arguable whether or not the objective is completely satisfied due to the ambiguity in the terms used. Namely, the reference to the 'social graph' being broad enough to include Twitter connections, but not necessarily be limited to Twitter

solely. An additional potential shortfall in meeting the objective fully is that neither of the applications take into consideration varying closeness in relationships of Twitter connections; i.e. is the Twitter connection mutual where both accounts are 'following' each other. Therefore the term 'friends' can also be said to be conformed to, but potentially not fully satisfied.

3.2.2 Reflection on Design Choices and Tools Used

Overall the design choices of API platform functionality, can be reflected upon as effective as they were suitable in facilitating the initial plan aims and objectives. I believe that the choices of external data sources are suitable overall for each type of content; on reflection limiting the text data source (e.g. Twitter) and sources for semantic data (e.g. Flickr) to a single source may have lead to a higher risk for bias in the experimentation. The presence of bias could not be proved or disproved in the project as the scope for the semantic content focused on relevancy rather than integrity and the data was crowd-sourced where possible. If other data sources of the same mediums (e.g. images, news etc.) were introduced into the design then the risk would arguably be significantly reduced. In contrast, the scope of the project should be considered and it is arguable that adding extra data sources would have taken time which would have been taken out of the experiments.

Similarly, the choice of using crowd-sourced data sources is also debatable. The use arguably provides a suitable set of data possible for testing for being autonomous due to the large volumne of data and frequent updates typically. It is also easily accessible through similar REST structured APIS and it also reduces the risk of bias as the data is contributed by many individuals. However, this does raise concerns for questionable integrity of the information. This was foreseen and for Tweets this was combated with integrity checks in processing the data post-retrieval during ranking. For the semantic information, the scope of the project did not include integrity checks only checks for relevance through matching tag precision, although this could be implemented in the future.

A final point of reviewing the chosen data sources is the choice of using Twitter as the primary data source for content to rank and/or get related semantic content from. On reflection, Twitter changed its policy of quota limits (Singletary 2013) in calling several of its endpoints which reduced the speed of both testing and experiments of features using retrieved Twitter data. It could be suggested that the choice of using Twitter was too heavily influenced on adopted use in other social media research [2]. rather than practical justifications. On the other hand, when relying on sources outside of personal control, this is always a possibility and the same could have occurred for any other source. It should also be noted that testing and experiments were still able to be performed, albeit at a smaller scale than previously envisaged. Overall, it is my opinion that the design choices for the data sources are suitable in regards to the project scope and in providing the capability for valid experiment conclusions.

In regards to design choices for the applications and information discovery processes that were used in the experiments, it is my opinion that the choices are justifiable given the project scope and time available for development. Those chosen were typical information discovery processes and the use of multiple processes and applications demonstrate suitability with minimised potential for bias. The design of these applications also satisfied the initial plan objectives and enabled experiments to be performed with them. If more time was available, more applications using different combinations of the platform endpoints (based upon more information discovery processes) would further reduce the bias and increase the validity of the results.

[2] Based upon a Google Scholar search result of 1,270 papers with 'Twitter' in the title, published in 2012. [Accessed: 27 April 2012]

3.2.3 Unforeseen Development Challenges

In terms of development of infrastructure and functionality for the API platform, unforeseen challenges were minimal in terms of severity in impacting the time plan. This can be suggested to be due to the design choices and tools used being intended for maximum flexibility and built around standardised protocols and conventions. Some minor challenges which had an unforeseen impact on development time were as a result of changes in the final deliverables requiring parts of the platform code needing to be re-factored; i.e. changed in some way for some benefit (e.g. efficiency) and then reintegrated without any negative impacts on other unchanged parts of the code.

In regards to unforeseen development challenges of the experiment applications, some of the tools to facilitate design choices made in the interim report had to be changed, including the choice of framework. As these choices only effected the internal structure of the application clients, I believe that the decision to change to a framework which provided quicker prototyping but less of a strict structure is justifiable given the small scale of its use in experiment context. In hindsight, more conclusive research should have been performed, perhaps with the inclusion of small prototypes, to test the suitability of the best framework for the applications; as this did have an unforeseen negative effect on the development time.

3.2.4 Known Limitations and Persisting Issues

Even with the test environment (platform and applications) being suitable overall in terms of satisfying requirements, design choices and tools used; some limitations can be highlighted. Firstly, the infrastructure of the API platform allows for simple expansion due to following an object oriented inheritance tree structure, but does not have capabilities for full composition in its current state. This is limiting as the internal mechanism of the platform is therefore not fully autonomous, i.e. expansion requires explicit definitions for each service. It cannot be concluded whether this would be possible due to the content structure of REST API endpoints not being standardised; but could be explored. The effect of this limitation would only be a persisting issue to internal development, which in the context of its use in the scope of this project, would not have any effect on user perception in experiments and subsequently no effect on the validation of the experiment conclusions.

Another persisting issue, is problems in the reliability of external data sources which could be considered the core components to this project. A related issue to the composition limitation, the unpredictability in both API call quotas (see Appendix, Figure 7) and deprecation of endpoints from the vendors is a persistent issue. This issue did not directly affect the validity of results, but quota changes by Twitter during the projects life-cycle (Singletary 2013) did limit the scale of participation in the experiments and the time taken to perform them. If assurances of stability cannot be made, then it is arguable that the platform for information discovery deployment cannot be fully autonomous. The severity of this problem could be reduced with some existing functionality of the API platform. For example, the inheritance structure aids in reducing the development resources needed to change to an alternative source. This could also be implemented beforehand and either have the platform return data from multiple sources; or have the alternative source be used in the event of an endpoint connection failure. This would create a chain of fall-backs so that deprecation would not call for immediate action as a result of downtime.

Additionally, Twitter is also unpredictable in changes to display requirements of the content it provides. The current implementations of the experiment applications do not satisfy these requirements fully in the their current state and any changes enforced by Twitter at any time would also have to be abided to. As this project was intended for academic research and not for profit in a public domain, e.g. a public consumer based service, this is not an issue. This would need to be considered if the context of the API platforms use were to be change beyond the scope of this project.

Additionally in terms of potential limitations that would affect the validation of any conclusions made, the choice of one source per content medium (e.g. videos) could be limiting in additional ways to potential bias. This limitation is the capabilities of the internal mechanisms of those source APIs e.g. ability to decode input and search efficiently. Finally, the specific formulas used for Tweet ranking were derived based upon perception and are unsupported due to lack of an existing formula approach to both Tweet integrity and interest; however, the variables used in the formula were influenced from other academic research (Uysal and Croft 2011). This presents potential uncertainty on having a negative impact on the validity of analysis and conclusion. This issue influenced the design choice of using two different information discovery to mitigate this occurrence, however, should be considered in reviewing the conclusions of the experiments.

3.3 Testing and Experimentation

This section aims to detail and evaluate the testing of the project deliverables, including functionality testing on the code base, as well as the experiments which aimed to produce findings regarding the ability for autonomous information discovery using crowd-sourced data. Testing relevant to the tagging algorithm is not present as the details of the prototype testing remain relevant from the interim report.

3.3.1 Code Base Testing

The code base for the experiment environment, which includes both the API platform and applications, was built in a small modules in an inheritance tree structure. This enabled testing to be performed through individual unit tests as each module was developed. This was the primary testing approach taken due to not having finalised deliverables for the majority of the development time; which required ongoing stability to avoid the prospect of having unstable code when deliverables were finalised. Attempts were made to improve the balance of code quality control and testing through the development methodology of: prototype module logic; test the suitability and then re-factor the code if necessary for better space or runtime efficiency. Where time was not available for a full analysis of the causes for a bug, a fix was injected into the code so that the issue was resolved. The fix was then marked with a code comment indicating that the issue may need to be reviewed to produce a more elegant solution to the issue if time was available. To ensure that negative effects on the experiment results were minimised, testing priority focused on functionality relevant to the scope of this project; rather than on long term features beyond their use in this project (e.g. the process to add other data sources). Lastly, final checks were performed on the experiment applications prior to experimentation to check for any problems in the interaction of all modules together; by performing the same tasks that were to be asked to participants.

On reflection, this approach does have disadvantages over more extensive methods such as producing detailed test cases, however these methods have a significantly overhead in creating test cases for every module and scenario. It is my opinion that the time taken to integrate a more extensive testing strategy would have been more than what was available and the amount of time left for experimentation would have been minimal. Moreover, by prioritising testing towards implementation relevant to the project scope over implementation primarily for future work, I believe the amount of testing performed was suitable to enable the project deliverables to be met. Nevertheless, the testing methodology could be reviewed in any extended work, due to uncertainty in the code stability beyond the scope of this project. In terms of benefit as a result of unit testing, it can justified as successful as any issues found during the final experiment application checks were only minor aesthetic issues.

3.3.2 Experimenting with Autonomous Information Discovery

This section provides details regarding the experiments performed using the functionality of the applications built on top of the API platform. Reflections will also be made on the outcomes against relevant

aims and objectives documented in the initial plan. The assessment performed through the experiments was qualitative and consisted of gathering feedback from participants performing various tasks with the applications. Qualitative assessment was chosen as the effectiveness of information discovery was assumed to be largely subjective. The experiments were performed locally by participants, due to constraints causing the set-up costs of crowd-source experiments to become infeasible. These constraints were time constraints in developing the applications to fully conform to Twitter's display requirements (Twitter, Inc 2012), configuring suitable global hosting hardware and configuring the environment to guarantee full task completion and appropriate feedback. Additionally the quota limits on the external APIs used (Twitter, Inc 2013a; see Appendix, Figure 7) act as a speed bottleneck on collecting results which would have to be closely monitored to ensure that limitations were not exceeded during an experiment with a remote participant; in which I did not have the control to do so. Local participants would minimise these issues to only the speed bottleneck, which could be managed with more control of experiment execution.

A task-centred approach was intended to provide an element of consistency to the qualitative assessment approach. The tasks performed would be application specific but with the aim to cover as many information discovery contexts or scenarios as possible, including personalised recommendations, social graph recommendations and related semantic information. Feedback was then aggregated together and analysed to suggest the suitability of: the tagging algorithm; the API platform and the experiment applications in being a suitable component for autonomous information discovery. Where a component was not completely suitable, reasons in relevance to particular approach and design aspects would attempt to be concluded.

For the time line application, the user would firstly be asked to comment on the appropriateness of the ranking order of Tweets from their personal home time line. With precision scores of keywords having a significant impact on the ranking, it was envisaged that this would give a first indication on keywords effecting information discovery of Tweet content. The purpose of this task was to suggest whether ranking, taking influence from tags, can highlight Tweets with desirable interest to the user. A secondary task would be asked of users for this application, by clicking on a desirable Tweet the Tweet area would expand to present the user with options on finding relevant semantic content to that Tweet. The user would be asked to comment on the relevancy of the suggested content in regards to their perceived context of the Tweet; if there was no relevancy could be found, to also suggest why if possible. The purpose of this task was to investigate the capabilities of finding related content about an interesting Tweet, by interfacing external data sources with the computed tags. The task for the tiled wall application was to comment on how interesting and/or relevant the content on the wall overall was to the user. This task was intended to experiment with the extent in which content from a user's social graph is also desirable to a user and whether any patterns could be suggested from the results as to why. Instructions on completing these tasks for each application were to perform them in any order or simultaneously so that the use of the applications would be as natural to the user as possible to increase validity of the results.

The result set consisted of feedback from 12 Twitter accounts due to time restrictions and infeasibility of large-scale crowd-sourcing for participants. The demographics of the participants were predominantly students but also consisted of 2 from professional backgrounds; ages varied between 20 and 55 and levels of Twitter experience varied from new users to frequent users. Other notable experiment parameters were: the experiments were performed at varying times of day, each account could only be used once; the number of Twitter accounts being followed to draw data from varied from at least 10 up to a few hundred and the type of Twitter accounts being followed varied between those with mostly friends and mostly enterprise or service accounts (e.g. news). Further supplementary information of the experiments can be found in Appendix, Figure 20. The experiments aimed to retrieve as many results as possible so that if any trends existed, they could at least be noticeable if not able to be proved.

From analysing the collated experiment results, conclusions can be made on the suitability of using the human emulating tagging mechanism as part of the two processes related to information discovery; ranking and interfacing with external data sources. In addition, the conclusions from this analysis are relevant to the following initial plan aims and objectives:

"Give indication as to the effectiveness of using text categorisation and ranking mechanisms to extract semantics from micro-blogging posts (e.g. Tweets); particularly with relevance to authenticity and suitability with formal and informal language expressions."

"Analyse psychological aspects relating to efficient information discovery, particularly exploring to what extent cognition can be optimised through implementable system mechanics."

Due to the relatively small result set, statistics will not be used in the conclusions and are intended to be initial observations. Nevertheless, trends were found in the initial observations and given the array of demographics in the participants, these trends should not be disregarded. The observations are grouped under two categories. Firstly, those which can be attributed to the specific approaches and design choices of this project and secondly, observations surrounding the viability of autonomous information discovery as a concept.

Implementation Specific Conclusions

Firstly, in regards to Tweet ranking in the time line application, the feedback gave little indication of any strong trend towards the extent of effectiveness in identifying Tweets with more interesting and/or integrity. However, indications were seen from some users who noticed some elements of intended outcomes such as grouping of similar Tweets and Tweets from more reputable sources and friends appearing higher than those not on similar content. On reflection, the results did not produce a strong trend to conclude from, however, it cannot be ruled out that this was potentially because of ineffective formulas in calculating the interest and integrity ranks, rather than the influence of the tagging algorithm being negative. Therefore, the design choice to experiment with two complex processes could be said to be concluded as being justified so that these results can be added to.

In regards to interfacing the tags with external data sources in order to retrieve related semantic content for both applications, the feedback did suggest a strong trend. Related semantic content was typically relevant to Tweets where content could be found. The primary cause of any issues was incorrect interpretations on tags where the term refers to multiple definitions or uses; see example in the Appendix, Figure 21. This arguably suggests that, tags being extracted are suitable and relevant for the most part and the inability to encode that tag with correct context so that it can be decoded is a concluding issue from the feedback; i.e. tags cannot be interfaced to external services to a degree which is predictable and stable. Participants were able to identify in most cases for text content (entities and news) why the incorrect content was being shown. For image and video content, participants found it more difficult on average to see how the context was misinterpreted. It cannot be concluded if this occurrence was due to the individual APIs used for the content but could prompt for further investigation.

Additionally, in regards to the quantity of relevant semantic content, a trend showed that participants which found the least amount of relevant semantic content were those where the Tweets primarily consisted of personal accounts (as opposed to business accounts). With the Tweets typically consisting of conversations which contained less tags with distinct entities than those Tweets which were single statements such as media article titles. For the personalised tile wall application with strong social graph influence, a weak

trend was that those participants where their home time line at the time of the experiment consisted of Tweets mostly from accounts of people considered friends beyond Twitter found suggested content on the tile wall application more desirable. Again, the trend prompts for potential further investigation as this could have been affected by: the types of personalised tiles chosen (detailed in the Appendix, Figure 19) and the negative effects of the misinterpretations of context.

Concept Specific Conclusions

A third conclusion from the results is a wider issue beyond the specific implementations of this project, regarding interfacing with external data sources. This being whether information discovery with the process of interfacing with semantic content sources is possible to be fully autonomous, given the dependency on numerous third parties for content and the features in the technologies they predominately use (REST). It is my opinion that the highlighted limitation of being unable to universally encode context with tags for mapping to REST API endpoints. is substantial enough to prevent this. This issue is a demonstration of the known limitation of a lack of standards and subsequent unpredictability of the content structure with REST architecture (Fielding 2000) and output formats (XML, JSON among others).

If hypothetically this problem was resolved and context of tags could be encoded 'perfectly' for interfacing against any REST API endpoint (or specific to each), it cannot be guaranteed that the encoding could be equally decoded by each external source given the unpredictability of the internal mechanisms for fetching content and the data structures of the content below the REST architecture layer. It could be suggested that the use of alternative semantic technologies could be explored through open-linked data such as the resource description framework (RDF) (W3C 2004). A drawback of these technologies is that they are not supported by the most consumer popular crowd-sourced data repositories, such as social networks. Additionally, conformance to a structure cannot be guaranteed as these typically follow conventions and not strict standards and therefore still have an element of unpredictability. Lastly, context of the extracted keywords would already need to be known so that it could be mapped to attributes for translation into interpretable queries such as SPARQL (W3C 2013b) or Meta-Query Language (MQL) (Google, Inc 2013c). These experiments have demonstrated that this conceptual issue may only be avoided through either: the reliance on third parties being removed for semantic content or somehow enforcing adoption of standards on the third parties.

A final conclusion is the occurrence where in some cases a tag produced similar semantic content where the link between the content and the tag could not be derived by the participant (even taking into consideration multiple definitions), in the cases of both applications. This was initially solely attributed to shortfalls of the data sources used, however, on inspection of the tags during this occurrence the tag was in some cases an expression or figure of speech. The amount of these occurrences in the result set are arguably higher than what I personally would consider as minor occurrences. Therefore further investigation into the appropriateness of the tag deriving approach of emulating human tagging could be performed.

For example, with the tags follow human tagging conventions, these conventions appear to be producing tags which are not always entities (i.e. people, objects, locations, actions etc). As a set, the tags produced appeared to summarise the text rather than categorise the text. This results in the Tweet being relatively legible if reconstructed from the tags, however, semantic content providers typically store data entries categorically (such as the Freebase API). This results in tags which are not categorical by themselves (i.e. not people, objects etc) returning content which is typically irrelevant regardless of context. This suggests an issue with using tags from a human emulating algorithm for information discovery and could be better suited for other environments which do not rely heavily on querying external services; for example, organising text data or part of augmented conversation.

4 Evaluation of the Problem Approach and Project Management

4.1 Problem Approach and Deliverable Focus

This section aims to discuss and reflect upon the decisions taken in terms of project planning and control in approaching the problem and delivering suitable deliverables. This differs from the approaches to design elements, by referring to the decisions taken in regards to focus and scheduling; particularly as a result of testing and the project state in relation to time-plan milestones.

4.1.1 Evolution of the Problem Approach and Deliverable Focus

The most prominent impact on the problem approach and project management was the evolution in the focus on what deliverables and outcomes were to be final at the end of the project life-cycle. This was not unforeseen as the project type intended to be experimental, however, the exact direction and final outcomes were unpredictable. This prompted the influence of the Agile software development manifesto over the traditional waterfall model. Any changes in focus was taken to improve the prospective outcomes and contributions of the project, with each revision integrating or being influenced by the previous deliverable into a perceived improved project focus.

A brief outline of the evolution of the project approach and focus is as follows. Firstly, from the project proposal to the initial plan, this shifted from a single consumer centric application with connections to various external data sources integrated into the application logic, to more of an experimental project in novel mechanisms to facilitate this. Particularly to investigate into improving the quality of tags created from existing text tagging mechanisms such as the Alchemy API (Orchestr8, LLC 2012); which were observed to be suboptimal from use in my previous projects. The intention was then to return to the original consumer application deliverable to provide a test environment for the new mechanisms and then compare these against those existing mechanisms.

Following initial prototyping of the tagging algorithm, the results showed potential benefits in suitability of the new tagging algorithm. Higher precision was produced in an investigation documented in the interim report where the output was tested against how a human tags Tweets and comparison to the Alchemy API (Orchestr8, LLC 2012). The positive results were unexpected as the design choices of the algorithm, such as the chunking patterns, were initial choices taken from perceived best choice from other research. The decision was then taken that the project focus was to be shifted towards experimenting with the suitability of the algorithm when the output was integrated as a part of more complex processes. The choices of which were influenced by the envisaged functionality of the proposal consumer application; these being as part of a ranking operation and interfacing tags with external API input to retrieve related semantic information. The use of two complex processes aimed to increase the validity of any conclusions made regarding the extent of suitability in practical information discovery operations; with the intention to perform experiments with more processes using the tags if time allowed. Refinement to find optimal configuration of the tagging algorithm therefore took lesser precedent. The reasoning was that information discovery related processes would provide an as accurate test environment as possible for practical uses beyond the project scope and subsequently increase the validity of any conclusions. In addition to this, the focus also allowed for some of the initial plan aims and objectives to continue to be applicable to the project which was adapting to different deliverables.

From prototypes towards final implementation, the approach towards experimenting with the effectiveness of the tagging mechanism in practical applications continued. The deliverable focus was expanded to maximise flexibility and scalability by creating an as autonomous infrastructure as possible, through means

of an API. This API would be a platform with abstract implementations of the ranking and semantic information retrieval processes. Several information discovery applications could then be built on top of this platform which would use different combinations and the complex processes alongside independent logic. This was to provide a means to reduce the potential for bias results, to then produce better validity in the conclusions; i.e. multiple implementations of multiple complex processes to comment on the practical suitability of the tags produced by the algorithm.

From implementation to experimentation, the approach and focus shifted for the final time following the initial results of the experiments highlighting conceptual issues. The shift was towards investigating these issues and analysing why they occurred to deduce whether the issues were as a result of design choices. The deliverable focus subsequently changed the overall evaluation and conclusion of the project to more of a reflection on the suitability of emulating human cognition through tags, for use in information discovery processes. Rather than full focus on suggesting the suitability of the theoretical and implementation design choices for deriving human derived tags computationally.

4.1.2 Reflection

The focus towards the project deliverables took many changes which could be reviewed as either a positive or negative. For example, it could be suggested that the amount of focus shifts coupled with the intensity of subsequent changes was reckless in terms of control and potentially indicates a lack of foresight. In contrast, it is arguable that the experimental nature of the project would inevitably cause changes in order to improve the contributions it brings. It is my opinion that the changes in the project focus following analysis of test results are justifiable as the changes allowed for broader contributions from low-level mechanisms through to consumer testable application.

However, several changes were as a result of new or refreshed ideas of previously narrow design choices. Particularly with the design choices for elements of the API platform becoming increasingly flexible; such as changing from one to the ability for many applications and splitting processes to be modular and accept custom configuration for flexibility. It could be said that my mindset was slightly too focused in these instances on satisfying the initial consumer centric application objectives and that the amount of resources spent on re-factoring code for flexibility provided minimal benefit to the new deliverable scope of the project. It should be noted that all cases of code re-factoring provides increased suitability for future work. Therefore, it is my opinion that the lack of foresight in planning the infrastructure design was the only significant slight negative on the project due to increased development time in re-factoring code rather than experimentation; rather than lack of foresight in the final deliverables.

Aside from the overall evolution of the approaches taken in project deliverables, another point for reflection is the choice taken to focus on experimenting with the suitability of the tagging algorithm in information discovery applications, over attempting to optimise it. This choice was arguably the most significant in terms of impacting the project's direction. On reflection of this I believe that either choice would have led to the ability to create a suitable valid conclusion for the project with experimenting with application providing breadth to the main contributions of the project and optimising the tagging algorithm providing depth in capability of emulating human tagging. Therefore it is equally arguably that the reasoning for the choice, being able to satisfy initial plan aims, was a justifiable choice. However, the perceived success of the algorithm was based on a single experiment, it could be argued therefore that more experiments could have been performed to confirm this before a decision was taken either way.

4.2 Evolution of the Time Plan and Project Management

This section aims to reflect on the effectiveness of the chosen methodologies for project management and time-plan strategies since the interim report. Particularly in relation to what elements remained persistent, what had to be changed and reflections on why it was deemed necessary.

4.2.1 What Aspects Remained Persistent

The approaches taken towards project management and control have remained relatively consistent with the general approaches taken throughout the early stages of the project. Likewise to the project management approach up to the interim report, the overall strategy was influenced from elements of the Agile manifesto (Highsmith and Fowler 2001). It should be noted that the adoption of Agile was not strict. The nature of the work shifted from the interim report to primarily implementation focused, however, the strategy to allow for flexibility and task-based rapid development remained relevant and beneficial.

More specifically to project organisation and task management, this was achieved through continuing regular weekly meetings with the project supervisor where possible to update with: findings from the previous week; resulting proposals for short and long term focus and a collection of relevant version control update notes for updated implementation, see Appendix Figures 22 through 40. The result of supervisor feedback coupled with the proposals for focus formed deliverables for at least the next meeting. This would be reviewed by myself mid-week to reflect on progress and adapt to what I believed was the best cost-effective focus given time available. The delivery of this information initially continued as short written reports, this become more verbal towards the end of the project as progress developed into demonstrations of implementation.

Moreover, in terms of maintaining effective control over implementation with adapting requirements, focus was increased from the use up to the interim of the using the version control system Git (GitHub Inc 2012) for managing snapshots of the code base. The features of Git (GitHub Inc 2012) that were utilised were: comments on saved states/versions to track progress; version roll-backs for mistaken file overrides and multiple independent 'branches' for stable tested code and unstable code for rapid development and prototyping. Lastly, in regards to time management in relevance to the initially set time plan, likewise to the interim report the allocated time for potential work overflow did remained relevant.

4.2.2 What Aspects were Adapted

Conclusions from the interim report showed incompatibility of the time-plan documented in the initial plan with the actual time-management. This was seen as primarily due to adapting requirements and changes in deliverable focus. Therefore, it was perceived that this would be likely to continue, given the further increased distances in focus from the planned focus at the time it was written. The approach was taken to not strictly enforce the completion of the original milestones at certain dates as this would be outdated; but still follow the structure of the original time-plan. For example, the time-management was focused so that completion of a milestone was still completed at a milestone completion date but not necessarily at the same date as previously planned.

4.2.3 Reflection

In regards to project planning and control, the items that either remained persistent or changed were largely due to the reflections taken in the interim report. The approach was to not change anything which wasn't having a negative impact on the project and focus on the project implementation itself. It could be debated that other methods of project planning and control could have been explored in an attempt to produce an optimal strategy. On review I believe that the approach taken to continue the methods from the interim

report was suitable as it was not harming progress. The intention of the project was to experiment with autonomous information discovery and not project management strategies. That being said, if a method did start to have a negative impact on the project then the strategy would have been adapted on an as-needed basis. Moreover, the adapting changes in the deliverable focused took time away from exploring these strategies even if there was desire to explore them. A positive of the changing deliverables was that the current strategy was tested for flexibility where any problems would have been highlighted and no significant on-going problems was found.

The only minor negative in the project management strategy was the use of methods where the features would have brought significantly more benefit for use within teams, particularly version control commenting. Towards the end of the project, the motivation to use it increasingly dwindled due to the time taken to write detailed comments on every change outweighed the benefit, as I would be the sole benefactor for the scope of the project. On reflection the use of this tool in my opinion suited the initial plan aims and objectives more than the project itself. It was useful in some aspects to the project, in particular with ensuring consistency in documenting progress throughout for comparison against the time-plan milestones; which I believe would have otherwise lapsed due to diverted attention on solving technical challenges. Also, if work were to hypothetically be resumed in the future, then the status of any implementation would be known quickly, without having to analyse the code base extensively.

On reflection of the influence of the Agile software development manifesto (Highsmith and Fowler 2001) on the project management strategy, I believe the choice over alternatives such as the waterfall model (Benington 1956) is a justifiable choice. Not only did the principles logically fit the experimental evolving deliverable nature of the project, but they also matched my personal methodology of working better through prototyping to explore solutions. However, from reflections on the deliverable focus, the lack of foresight in design did cause some increased resources in development; which could be argued as a point towards the use of the waterfall model. Weighing the significance of these points up I believe that the influence of Agile (Highsmith and Fowler 2001) enabled the project deliverables to reach fuller potential by enabling the adaptation and expansion of requirements and therefore deliverables. These reflections have suggested that a hybrid approach might have been a better approach. By taking a waterfall model approach (Benington 1956) to the design elements at each significant change in deliverable focus, this would have the potential to reduce the amount of time spent on tasks such as code re-factoring. The exact specifics of this in relevance to the time plan would have to measured such as the time spent on complete design against prototyping and re-factoring. It cannot be concluded whether a hybrid approach would have been better for the project overall; if future work were to be conducted this could be explored.

5 Future Work

This sections aims to demonstrate deeper understanding of the project problem by suggesting additional work that would supplement the contributions from the project. As with discussion and evaluation of the project, information relevant to the core tagging algorithm and the test environment have been separated to iterate the interdependency of the tagging algorithm from how it was experimented within this scope. Improvements for both sections involve ideas for attempting to optimise performance, improve the infrastructure and resolve known limitations.

5.1 For the Tagging Solution

The successful prototype testing results documented in the interim report prompted the decision to shift focus towards testing the applicability of the core tagging algorithm in information discovery scenarios. The mechanisms of the algorithm were suitable to do so but are potentially suboptimal.

A possible opportunity to optimise the tagging precision and recall is to analyse the human tagging data from Mechanical Turk for patterns in the part of speech tags of the tags they selected. If possible, to also perform more tests for a larger set of results to analyse. The outcome of this would suggest whether an optimised set of part of speech chunking patterns could be achieved. For example, if no contradictions exist with any of the patterns (e.g. a verb followed by a noun is always a tag) then this would suggest that hypothetically with enough results, all pattern possibilities could be captured for 100% precision in tags produced computationally. However, if contradictions were found, for example a pattern is only a tag for some words and not others, then this would indicate that 100% precision would not be achievable. It should be noted that results from the prototype results, evidence showed that 100% recall would not be possible due to variations in words contained in a tag before and after a mutually agreed word; e.g. given a string with words represented by W X Y Z, one person may tag 'X Y' as a keyword, another 'W X Y' or 'X Y Z'. Therefore it should be noted that an 'optimised' algorithm is a matter of perspective, for example 100% precision may be considered optimised as a subset of the tags would match all human derived tags; but this could equally be achieved by creating a set of all combinations of sequential words without the algorithm at all. Some might consider optimised as being as close to 100% precision and recall as possible, for example, to minimise the potential for redundant processing of tags when iterating; leading to subsequent higher chance for worse case runtime complexity and storage space complexity. Therefore, analysing the patterns in this way could produce an 'optimised' set of patterns by indicating what tags would increase precision without potentially reducing the recall too much; the thresholds for which also being up to interpretation.

In addition to attempting to optimise the patterns for producing tags, experimentations could be performed with using different text corpora as the reference for classifying words with part of speech tags. The corpus used in the implementation of the algorithm in this project is the Brown corpus, which is widely accredited but is based on texts published in 1961 (Francis and Kucera 1979). It could therefore be explored as to whether using a corpus based on more recent texts would produce better part of speech classifications for words or not. For example, newer corpora might provide more accurate classifications of new words rather than falling back to a default (e.g. Noun) or words where the use of those words may have changed since 1961.

Additionally, the natural language processing methods implemented into the tagging algorithm was chosen from assumed suitability. Experiments could be performed that assess the impact of using other methods in addition to, or as a replacement of the existing methods. For example, is there any beneficial effect of using the process of chinking over or with chunking; which refers to the opposite operation of chunking by splitting sequential words up by some pattern rather than joining them together.

Another proposed task for future work is to find solutions to the currently known limitations to the tagging algorithm. Most notably, the incapability in resolving spelling errors and problems relating to irregular word capitalisation.

5.2 For the Experimentations

The small result-set used in the experiments is arguably one of the most significant limitations of this project, therefore performing more of the same task-based experiments to improve validity in the resulting trends would be advisable for future work. Additionally resources could be spent on ensuring conformity to Twitter's display guidelines (Twitter, Inc 2012) and suitable hosting to then perform moderated amounts of testing using crowd-sourcing services; which would provide benefits such as controllable demographics and potentially speedier responses. Some additional tasks for future work could include implementing more external data sources for each content medium to negate the possibility of bias results further and implement other processes which make use of the tags produced from the algorithm, such as location or event recognition. Beyond the context of this project, investigations could be carried out around the suggestion of sets of tags being a summarisation of text rather than capturing what the text is about for categorisation. More specifically, could the tagging algorithm be better, worse or equally suited in other environments aside from information discovery.

5.2.1 For Experiment Environment

If the current experiment environment were to be continued in future work, improvements could be made to the infrastructure for more efficient use, particularly if implemented for use outside of controlled experiments like the scope of this project. Firstly, investigate into the possibility of being able to interact with any external REST API (Fielding 2000) through generic URL wrapping rather than a library. A library can be instantly become deprecated if the location an endpoint or sending input parameters, however, the method for reaching an endpoint (through URLs) is standardised. It would be beneficial therefore if a method for specifying a location and parameters as specifiable variables existed, so that any changes could be made quickly for a service and be decoupled from any other code. This would theoretically be possible for any accessible REST API (Fielding 2000) based upon experiences of this project. Considerations would have to be made into specific authorisation methods of each API as this would not be predictable (e.g. via OAuth (Parecki) or api key).

Additionally, features could be implemented to facilitate more efficient use of the modularised endpoints. For example, if a user wanted to: tokenise, client, part of speech tag a piece of text, but not perform chunking of tags, this would require three API calls and is inefficient in terms of speed and network traffic for both the API and the consumer. To improve the usability of performing these operations a method of chaining calls together could be devised so that only a single API call would be required for multiple endpoints with sequential output to input.

Lastly, actions could be performed to improve the suitability for larger scale deployment outside similar scope of this project. This includes but not necessarily limited to, expanding use of features of the TastyPie framework (Lindsley et al 2012) (adopted as the backbone of the API platform) such as: 3rd party application authentication and management; user quota throttling of API endpoints if necessary; output caching to manage server computation load more efficiently and authorisation mechanisms such as session or OAuth between the consumer and the platform. Additionally, a more conclusive testing strategy would be beneficial for stability, particularly if used outside the scope of this project due to reduced focus as a result of time constraints.

6 Conclusion

In conclusion, the main contributions of the project can be summarised as a demonstration of the difficulties associated with autonomous information discovery using crowd-sourced data. Overall the aims and objectives of the project went further than initially intended to investigate into providing more novel implementations of the lower mechanisms (such as method of tagging) and also attempting autonomous deployment of information discovery processes. The completeness of the scope covering low level mechanisms through to scalable platforms through to consumer applications, provides a broader outlook of suitability; which would have been lost with more depth towards a narrower scope. Given the time available for the project, broadness or depth would have to be compromised regardless of the design choices made. This is the primary contribution of the project for use in future works. In that the project has shown aspects to approach and design that work well and those that work not so well for autonomous capabilities of an entire information discovery environment. The contributions and deliverables of the project have the ability to be used in further research both as independent components or remaining together, for at least the tasks suggested as future work.

However, the project has also demonstrated the capability of emulating human cognition towards tagging to an extent that provides higher precision on average than a popular alternative, Alchemy (Orchestr8, LLC 2012). The project has also provided a basis of evidence for further exploration including: shortfalls in encoding and decoding context of tags and discrepancies in the effectiveness of tags with different information discovery processes. Additionally, conceptual issues have been highlighted regarding unpredictability of REST architecture structure for autonomous data retrieval. Additionally, indications have been that show human derived tags produce more of a summarisation of the text rather than categorising it. Which could be better suited in environments where the tags are used with data that is similarly structured. The project planning and control strategy remained suitable for the project across the evolution of the deliverables; with only minor issues relating to some tools and methods being too heavyweight for the context they were used. Lastly the content of the report has at least followed what was set out in the initial plan.

7 Reflection on Learning

The purpose of this section is to review my learning experiences from undertaking the project including what I expected and what I didn't expect in regards to personal development, as well as any advantages and disadvantages as a result. Several personal learning aims were set in the initial plan which intended to provide at least a suitable foundation of knowledge to take forward; including exercising my current skill-set and preparing requisites for future success. The first of which was:

"Exercise knowledge gained over the past two years of the degree programme such as: building upon the foundations of mainstream programming languages (such as Python) to gain experience of a project scale larger than previous academic projects; practice development paradigms (such as Object-Oriented) and project management techniques."

It is my opinion that this aim was met with the implementations of both the tagging algorithm and the experiment environment environment. Throughout the project the approach was taken that if I possessed existing knowledge applicable to solve a problem that would be used to exercise use in an unseen environment. When choice of technologies were negligible, the process was to chose relevant but previously unexplored technologies where the learning curve was deemed suitable. In order to boost the broadness of my skill-set and practising applying learning techniques quickly and where necessary. Overall the effectiveness of this aims execution is subjective, however, I believe that the intention and execution were suitable. With neither existing or new skills having a majority focus and learning new skills provided challenges without hindering progress. A possible reason for this is my previous academic projects have largely surrounded technologies related to web services and thus computational thinking in this area was transferable.

A second personal learning aim set out was:

"Gain understanding of processes used in industry and academia which I have little to no experience of; such as version control systems (e.g Git) and project life cycles (e.g. Agile). Use the opportunity to explore emerging technologies such as Rich Internet Applications) by means of frameworks (such as Node.js or Backbone.js) and standards (such as HTML5)."

The project overall conforms to this learning aim, however on reflection the suitability of the technologies varied. The intention in extending my skill-set to include experimenting with methodologies not practised in previous academic projects was positive overall. In some cases however it was more trying to make the methodologies fit the project for this personal aim, than the methodologies fitting the project for benefit of the project. This was especially apparent for the version control system Git (GitHub Inc 2012), where the extensive features would be better suited for team-based environments or long-term code management. The subset of features used in this project, namely version backups, could have been facilitated with simpler mechanisms such as auto cloud storage backups or manual compression of the project state to a file. Therefore the use of Git use in this project did not provide a means of gaining an understanding to a depth which would be directly scalable.

In terms of the project life-cycle management strategy, full adoption of the Agile manifesto (Highsmith and Fowler 2001) was not possible as the principles behind it are largely focused on teams. Nevertheless, the experimental nature of the project and evolving project deliverables did enable influence from the Agile manifesto (Highsmith and Fowler 2001) in the project management. I believe that this partial application, supplemented with previous theoretical understanding has provided me basis of experience which could be applied to a team Agile environment. In terms of exploring emerging technologies, the project provided

an opportunity for me to extend my skill-set with experimenting with emerging technologies without a irreversible negative effect on project progress. As with the use of Git however, I believe that in cases such as the framework behind the platform applications, the intention to try and fit these technologies to the project was not effective. On the other hand, where these technologies did not fit due to incompatibilities or lack of time, fall-backs were available in my existing skill-set and the opportunity provided a means to apply these to a new problem. Overall, the amount of emerging technologies adopted was smaller than desired, however, it is my opinion that the negatives were minimal and the strength of my skill-set was improved nonetheless.

The final personal learning aim set out was:

"Complete a project that has the objectives and complexity to be an attainable stepping stone into the area of research (Psychology in Computer Science), that I wish to taking forward into postgraduate study."

The final deliverables of the project changed considerably from the initial plan through to post-implementation. These changes in my opinion added to the contributions the project initially intended to provide and that this is also the case regarding personal preparation for further study. An arguable negative is the design choice of breadth in the project scope, rather than depth (i.e. lower mechanisms through to delivery infrastructure and to consumer application). By focusing on depth towards a single project deliverable, such focusing on optimising the tagging mechanism, it would have honed greater understanding of some areas of research (i.e. natural language processing). Equally, by taking the approach of honing on breadth of understanding, it has provided a wider set of understanding of lower-level algorithms through to application strategies. With my exact knowledge requirements for future developments not being fully known, I believe that approaching the problem with the aim for breadth in knowledge is justifiable.

In addition to aims set in the initial plan, my experiences by undertaking this project have highlighted my lack of experience at the start of the project regarding project management of this scale. My previous experiences provided a suitable foundation to undertake the project, I overestimated my capabilities in optimally managing task management and taking into consideration Hofstadters Law (Hofstadter 1999). This predominately only had a minor effect on workload scheduling and did not have a decremental effect on achieving the project deliverables on project completion.

References

Ashkenas, J. 2012. *Backbone.js* [Online]. Available at: http://backbonejs.org/ [Accessed: 21 April 2013].

Benington, H.D. 1956. Production of Large Computer Programs. In: Navy Mathematical Computing Advisory Panel. *Proceedings of the Symposium on Advanced Programming Methods for Digital Computers*. Washington, D.C., 28-29 June, 1956. Washington, D.C: Office of Naval Research, pp. 15-27.

DeSandro, D. 2013. *jQuery Masonry* [Online]. Available at: http://masonry.desandro.com/ [Accessed: 21 April 2013].

Django Software Foundation. 2012. *The Web framework for perfectionists with deadlines.* [Online]. Available at: https://www.djangoproject.com/ [Accessed: 21 April 2013].

DuVander, A. 2010. *New Job Requirement: Experience Building RESTful APIs* [Online]. Available at: http://blog.programmableweb.com/2010/06/09/new-job-requirement-experience-building-restful-apis/ [Accessed: 27 April 2013].

Fielding, T.R. 2000. *Architectural Styles and the Design of Network-based Software Architectures.* PhD Thesis, University of California.

Francis, W.N and Kucera, H. 1979. *Brown Corpus Manual* [Online]. Available at: http://icame.uib.no/brown/bcm.html [Accessed: 21 April 2013].

GitHub Inc. 2012. *github* [Online]. Available at: https://github.com/ [Accessed: 12 December 2012].

Gnass, F. 2013. *spin.js* [Online]. Available at: http://fgnass.github.io/spin.js/ [Accessed: 21 April 2013].

Google, Inc. 2013. *Freebase API* [Online]. Available at: https://developers.google.com/freebase/ [Accessed: 21 April 2013].

Google, Inc. 2013. *Frequently Asked Questions - Freebase API* [Online]. Available at: https://developers.google.com/freebase/faq [Accessed: 21 April 2013].

Google, Inc. 2013. *MQL Overview* [Online]. Available at: https://developers.google.com/freebase/v1/mql-overview [Accessed: 21 April 2013].

Google, Inc. 2013. *YouTube API v2.0* [Online]. Available at: https://developers.google.com/youtube/2.0/developers_guide_protocol [Accessed: 21 April 2013].

Highsmith, J. and Fowler, M. 2001. The Agile Manifesto. *Software Development Magazine* August 2001, pp. 29-30.

Hofstadter, D.R. 1999. *Godel, Escher, Bach: An Eternal Golden Braid.* New York: Basic Books.

InuYaksa. 2013. *jQuery NiceScroll plugin - scrolling for desktop, mobile and touch devices* [Online]. Available at: http://areaaperta.com/nicescroll/ [Accessed: 21 April 2013].

Lindsley, D. et al. 2012. *Welcome to Tastypie!* [Online]. Available at: http://django-tastypie.readthedocs.org/en/latest/ [Accessed: 21 April 2013].

NLTK Project. 2012. *NLTK 2.0 documentation* [Online]. Available at: http://nltk.org/ [Accessed: 21 April 2013].

Orchestr8, LLC. 2012. *AlchemyAPI Keyword / Terminology Extraction* [Online]. Available at: http://www.alchemyapi.com/api/keyword/ [Accessed: 21 April 2013].

Parecki. A. *An open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.* [Online]. Available at: http://oauth.net/ [Accessed: 25 December 2012].
Reenskaug, T. 1979. Thing-model-view-editor-an example from a planningsystem. *Xerox PARC technical note* May 1979.

Singletary, T. 2013. *Planning for API v1s Retirement — Twitter Developers* [Online]. Available at: https://dev.twitter.com/blog/planning-for-api-v1-retirement [Accessed: 21 April 2013].

Tangelder, J. 2013. *Hammer.js - A javascript library for multi-touch gestures* [Online]. Available at: http://eightmedia.github.io/hammer.js/ [Accessed: 21 April 2013]

The jQuery Foundation. 2013. *jQuery* [Online]. Available at: http://jquery.com/ [Accessed: 21 April 2013].

The jQuery Foundation. 2013. *jQuery UI* [Online]. Available at: http://jquery.com/ [Accessed: 21 April 2013].

Twitter, Inc. 2012. *Developer Display Requirements* [Online]. Available at: https://dev.twitter.com/terms/display-requirements [Accessed: 21 April 2013].

Twitter, Inc. 2013. *REST API Rate Limiting in v1.1* [Online]. Available at: https://dev.twitter.com/docs/rate-limiting/1.1 [Accessed: 27 April 2013]

Twitter, Inc. 2013. *The Twitter REST API* [Online]. Available at: https://dev.twitter.com/docs/api [Accessed: 21 April 2013].

Twitter, Inc. 2013. *Using hashtags on Twitter* [Online]. Available at: https://support.twitter.com/articles/49309-using-hashtags-on-twitter [Accessed: 21 April 2013].

Uysal, I. and Croft, B.W. 2011. User Oriented Tweet Ranking: A Filtering Approach to Microblogs. In: Berendt B. et al. eds. *CIKM'11: Proceedings of the 20th ACM international conference on Information and knowledge management.* Glasgow, Scotland, UK, 24-28 October, 2011. New York: ACM, pp. 2261-2264.

Verdone, M. et al. 2013. *Python Twitter Tools* (command-line client and IRC bot) [Online]. Available at: http://eightmedia.github.io/hammer.js/ [Accessed: 21 April 2013].

W3C. 2004. *RDF/XML Syntax Specification (Revised)* [Online]. Available at: http://www.w3.org/TR/REC-rdf-syntax/ [Accessed: 21 April 2013].

W3C. 2013. *HTML 5.1 Nightly* [Online]. Available at: http://www.w3.org/html/wg/drafts/html/master/

[Accessed: 21 April 2013].

W3C. 2013. *SPARQL 1.1 Overview* [Online]. Available at: http://www.w3.org/TR/sparql11-overview/ [Accessed: 21 April 2013].

Yahoo! 2013. *REST Response Format* [Online]. Available at: http://www.flickr.com/services/api/response.rest.html [Accessed: 21 April 2013].

Yantsiou, L. 2008. *Pipes News Aggregator* [Online]. Available at: http://pipes.yahoo.com/pipes/pipe.info?_id=dc2ed1fb04aba532f3d063e734c47c50 [Accessed: 23 April 2013].