Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

DISSERTATION

PRESENTED TO THE
UNIVERSITY OF DUBLIN, TRINITY COLLEGE
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MAGISTER IN ARTE INGENIARIA

Supporting the Curation of a Structured Collection from the Unstructured Open Corpus

---

Author:
Conor O'SHEA

Supervisor:
Prof. Owen CONLAN

May 19, 2016

Declaration

I, Conor O'Shea, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Conor O'Shea

May 19, 2016

Acknowledgements

There are a number of people to whom I owe a great deal of appreciation for their contribution to this work.

My supervisor Owen Conlan, who's enthusiasm, expertise and guidance made a daunting task much more manageable.

My parents and sisters, who have supported and encouraged me in my education for 25 years.

Philip, who provided occasional proof reading and frequent kind words. And my classmates for their welcoming of me onto the team despite arriving late to the party.

Thank you all.

Abstract

This paper presents the design, implementation, and evaluation of a Web annotation system which allows users to save and tag a reference to any piece of text on the Web while maintaining the resource in its original context. Semantic uplift technology provided by the DBpedia Spotlight Web service allows the system to present the user with suggested hierarchically structured semantic tags to annotate these references. The intention is to save the user time and provide further structure to the final collection when compared with a flat user-generated tagging system.

The system was evaluated with five participants conducting a small amount of research using the system. Participants found the system easy to use and beneficial to the research process. A number of areas for potential future work on this system that may result in further benefits to the research process were identified.

Summary

Conducting research on the Web has many benefits but also some pitfalls. For example information is so widely available and paths through the information so numerous that key resources which directed the course of the research can be lost and the collection of resources gained at the end of a period of research is often unstructured and specific resources can be difficult to find again.

This paper presents the design, implementation and evaluation of a Web annotation system which allows users to save and tag a reference to any piece of text on the Web. The fact that it is a reference allows the resource to remain in its original context. Semantic uplift tool DBpedia Spotlight allows the system to present the user with suggested hierarchically structured semantic tags for resources, to both save the user time and to provide further structure to the final collection.

The system is built as a browser extension using the Mozilla Firefox Add-On platform. It provides users with three major features:

1. Save and tag resources using suggested semantic tags or ordinary custom tags
2. View the list of saved resources

3. Visualise tag frequency in the saved resources

The suggested tags are generated by making a request to the DBpedia Spotlight Web service with the selected text as input.

Five users evaluated the system by using it while conducting research for a short period of time. The users then filled out a System Usability Scale questionnaire, a questionnaire to assess their computer proficiency, a qualitative questionnaire which focused on the fulfilment of system requirements, and a short interview which focused on potential improvements to the system amongst other things. A number of possible further developments to the system were identified including further data visualisation and support for cloud storage of annotations.

Evaluation participants of both extensive and limited computer proficiency found the system easy to use and beneficial to the research process. They suggested a number of potential use cases that had not been previously identified and indicated that some further work on visualisation of collection data to make it more clear would be beneficial.

Contents

List of Figures

Chapter 1

Introduction

The title of this research *"Supporting the curation of a structured collection from the unstructured open corpus."* warrants some unpacking. This chapter attempts to do this by discussing the background to this project, its contribution and its motivation. It will also outline the scope of the project and the structure and content of this dissertation.

1.1 Background

The question that this research attempts to answer is:

"How can a user researching a specific topic in an unstructured environment be supported in creating a coherent and structured collection of resources while researching?"

Two of the core concepts in this research are the open corpus and the structured collection.

A corpus is a collected body of texts on which some analysis is conducted. For example, Adaptive Hypermedia Systems (AHS) require a corpus which they then present to the user based on a model of the user's needs and a model of the structure of the corpus. Some analysis must therefore be conducted on the corpus to generate this model such that effective personalisations can be made. Naturally this analysis is distinctly easier if the structure of the corpus is known. This is the root of the distinction between open and closed corpus. Brusilovsky and Henze defined closed corpus as being (Brusilovsky and Henze, 2007a):

"...where documents and relationships between the documents are known to the system at design time."

And open corpus as:

"...a set of documents that is not known at design time and, moreover, can constantly change and expand"

Some of the limitations of solutions that are designed for a closed corpus are that obviously, resources are limited to those within the corpus, but also that the solution is not reusable in other corpora. This project aims to support structuring of collections of open corpus content from the Web in order to aid the research process.

Collections provide interesting information in their own right, beyond that which can be gained from the individual items. In galleries and museums for example, collections provide context to the items included and it is often vital that collections remain together so as not to lose that extra meaning. By collecting the resources that a user has found interesting, the system strives to help the user gain insights about the trajectory of the research,

how resources are connected, and also how focus changed over the course of the research. By adding semantic structure to this collection with the help of semantic tagging, there is further scope for analysis of this kind.

1.2 Contribution

The contribution of this work is the creation of a Web annotation application which allows users to collect and semantically tag text resources from anywhere on a web-page, and leave them within their original context. The semantic tagging allows for reduced overhead to the user and also results in a more structured collection than a flat tagging system.

1.3 Scope

The system provides users with a low-overhead method of bookmarking and semantically annotating sections of text from the open Web. A fully featured and realised application which is ready for release is beyond the scope of this project. This project focuses on bookmarking, semi-automatic annotation and visualisation features. The user is able to bookmark any highlightable text on the Web and tags may then be added to the bookmark either manually or using automatically generated tags from DBpedia Spotlight. Finally there will be a simple visualisation of the structure of the tags to provide the user with feedback as the the structure of the collection.

1.4 Motivation

By and large, research on the open Web is an unstructured process in an unstructured environment. The researcher is not always sure of where to look for information to begin with or indeed what information they should be looking for. As the research process continues, often the focus of the research can change entirely as the researcher gains further understanding of the field. However the process by which these changes came about is generally lost. More generally, during the research process the collection of resources that have provided information key to the research being conducted is often unstructured and some important resources may even be lost or forgotten about.

The curation of a structured collection during the course of the research process has the potential to prevent some of this key information from being lost. A structured collection would allow for better rediscoverability of key resources, and grouping of resources into sub-collections.

This structure also has the potential to indicate the direction of the research. By reflecting the structure of the collection back to the user, they may see patterns emerging which could spark new paths of research. Also analysis of paths taken by other users could open the possibility of suggestion of resources, which could further reduce the unstructured nature of the research process.

A curated collection of resources on a topic can be helpful in gaining an understanding of the topic. Particularly if the resources have been curated by someone knowledgeable of the topic. However, if the resources are removed from their original context, this can preclude the potential for further

exploration of the topic and full understanding of the resource. By simply highlighting the resource, this system allows it to be clearly identified but to remain in its original context.

1.5 Outline

The State of the Art

This section will detail the influences that this research took from the state of the art. These influences came mainly from the areas of Web annotation and the Semantic Web.

Design and Implementation

This section discusses the technologies used in the design of the system and its implementation details. The main technologies used are the Mozilla Firefox Add-On framework based on JavaScript and DBpedia Spotlight Web service which annotates input text with relevant DBpedia resources.

Evaluation

The evaluation section details the evaluation method for the system. Five participants took part in the evaluation. After using the application to conduct research for a short while they completed three short questionnaires including the System Usability Scale, and a short interview.

Future Work

A number of areas for potential future work on this system were identified, many of them during the evaluation. This section classifies them as either short-term work or long-term work and identifies a possible path to their completion where one is apparent.

Conclusion

The conclusion section draws some conclusions from the evaluation and the research as a whole.

Chapter 2

The State of the Art

The application proposed in this project draws on elements from a number of areas of research. This section will discuss the state of the art in the areas of research from which this project draws its inspiration. It will also discuss some limitations that this project intends to address.

The annotation of Web resources has been a feature that many tools have provided by varying methods for about two decades. In more recent years, as the popularity of the Semantic Web and Linked Data grows, some tools have begun to incorporate the annotation of documents with Linked Data resources. Using natural language processing and other semantic uplift techniques some tools can now automatically generate metadata for resources and enrich the structure of collections. As adaptive hypermedia systems look more towards the open Web for content instead of a purpose built closed corpus, the structure that can be gained from this automatically generated semantic metadata becomes extremely useful. This project draws inspiration from Web annotation technologies and also from Semantic Web technologies that aid the automatic annotation of resources.

2.1 Web Annotation

In the physical world, the annotating of resources is an important and intuitive part of the research process. Annotations provide the annotator and others who have access to them with insight into the subjective interpretation of the resource by the annotator at the time. Indeed Pierre de Fermat, famous for his teasing annotations, proposed arguably the most sought after mathematical conjecture in history, Fermat's Last Theorem in an annotation in the margin of a copy of Arithmetica. Finally proven in 1995 by Andrew Wiles (Wiles, 1995), perhaps had Fermat not been subject to the limitations of physical annotations, 358 years of mathematical effort may have been saved!

Web Annotations are an attempt to port that ability for a reader to express thoughts about a resource to documents and resources on the Web (W3C, web). The annotation of documents and pages on the Web has long been considered to be a valuable practice; providing for further interaction beyond the restrictive early WWW model of "active information publishers" and "passive information consumers". The data obtained from the annotation process can also be used to enrich the information on the document or page with semantic labelling for example. This information can then be used for filtering and searching information in the document (Vasudevan and Palmer, 1999).

2.1.1 Social Bookmarking

For the last two decades, Web annotation has mostly taken the form of "Social Bookmarking". Social Bookmarking services allow users to share book-

marks of Web documents with an online community. They also allow users to add metadata to these bookmarks such as tags, comments and ratings (Noll and Meinel, 2007). User generated tags such as these not only aid the user in relocating items previously tagged, but also allow for the potential of a broader classification system for the tagged resources based on the tags chosen by users and how often individual tags are used to describe the same resource. This tagging process, pioneered by Flickr and Del.icio.us amongst others, gave rise to the coining of the term "Folksonomy" by Thomas Vander Wal in 2004 (Vander Wal, 2007). In this article, he defines a folksonomy as "the result of personal free tagging of information and objects (anything with a URL) for ones own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information."

This folksonomy style classification allows for improved rediscoverability of resources for individuals and within a community. But when these folksonomies are public they can also provide data which is useful for enhancing Web search (Heymann et al., 2008) (Yanbe et al., 2007) and may be integrated into the Semantic Web to help to populate ontologies for example (Specia and Motta, 2007).

Del.icio.us

Del.icio.us was launched in 2003 and provided users with a platform to save bookmarks to Web pages along with relevant tags and a note. These bookmarks were then aggregated together with other users bookmarks. This allowed for posts with similar tags to be grouped together and if enough users tagged a specific URL, a loose classification of that page could be inferred

(Buddulf, 2004).

Del.icio.us provided users with three main methods of content discovery. The URL `del.icio.us/<username>` would present all the recent posts by a given user. `del.icio.us/tag/<tag>` would present all the most recent URLs posted with that tag, and `del.icio.us/url/<URL-code>` would present all the tags that users have given to that URL. Some of these patterns could also be combined. For example `del.icio.us/username/tag1+tag2` would present all the posts by that user with the tags "tag1" and "tag2".

In 2011 Del.icio.us released a feature called "stacks". Stacks allowed users to group together links on a stack page along with a title and a description for the stack page. This feature was removed however in 2012.

Diigo

Diigo was launched in 2006 and provides a very similar set of features to Del.icio.us. Users may tag and share bookmarks amongst groups or maintain them privately. Diigo provides two main features further to those provided by Del.icio.us: users may highlight and annotate any section of a webpage and there are Chrome, iOS, Android and Windows Phone applications making Diigo more flexible for use on different devices (Diigo, web).

A.nnotate

A.nnotate was launched in 2008 and provides a similar service to Diigo as regards annotating sections of documents. However A.nnotate is focused

towards uploaded PDF, word processing and image documents. Users may make annotations and comments about any section of the document and share this with other users. Diigo also added PDF support for documents already hosted on the Web.

Limitations

There are two main problems with this format of annotation for the research process that this project hopes to address. Firstly, thinking up tags to add to a resource can be a time consuming process and can therefore interfere with the fluency of the research process. Secondly, the tags that are generated in this way are inherently unstructured and limited in meaning, particularly meaning which is machine-understandable. This project hopes to address these limitations by incorporating some Semantic Web technologies.

2.2 DBpedia and The Semantic Web

The Semantic Web is a concept coined by Tim Berners-Lee to refer to Web content that is "meaningful to computers" (Berners-Lee et al., 2001). The W3C provide many of the standards which are core to the functionality of the Semantic Web, in particular the Resource Description Framework which provides the standard for how resources on the Semantic Web should be described. According to the W3C "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries." This sharing and linking of data that is also meaningful to machines allows for machines to make inferences and

create new meaningful data.

2.2.1 Linked Data

Linked Data refers to a method of publishing structured data on the Web and linking it to other data sets to add context and increase discoverability. Tim Berners-Lee outlined a set of principles to which Linked Data should adhere. They are considered best practice for contributing data to the Semantic Web (Bizer et al., 2009a).
Linked Data Principles (Berners-Lee, 2006):

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. so that they can discover more things

DBpedia is an example of a large and open Linked Data set.

2.2.2 DBpedia

DBpedia is a community project to extract data from Wikipedia and to publish it on the Web as Linked Open Data (Auer et al., 2007) (Bizer et al., 2009b). It provides:

- An extraction framework to extract structured data from Wikipedia articles and express it in RDF.

- The RDF triples published accessibly.
- Interlinking of the data set with other large open datasets, and
- Interfaces and modules to allow the dataset to be accessed via Web services etc.

The extraction framework functions by parsing each Wikipedia page and sending the resulting Abstract Syntax Tree to various extractors to extract RDF triples from it. Mapping communities aid the work of the extraction algorithms by manually mapping properties from Wikipedia's infoboxes to properties in the DBpedia ontology.

DBpedia resources are then linked to many other large open datasets such as YAGO and Geo Names. DBpedia also functions as a key central hub for datasets to connect to in order to link them to the Web of open Linked Data. As of 2015 there were approximately 39 million links made to DBpedia from various datasets (Lehmann et al., 2015). Figure 2.1[1] shows a small subsection of the largest and most closely linked datasets to DBpedia.

2.3 Semantic Annotation

While the Semantic Web is becoming more and more a part of the way information is stored and presented on the Web - many search engines now use the schemas defined by the schema.org project to display infoboxes and star ratings alongside search results for example - the Web today still favours multimedia documents as a method of disseminating information (Mrabet

[1] Image from http://lod-cloud.net/ - Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak

Figure 2.1: DBpedia links to other datasets

et al., 2015). Semantic annotation aims to make text on the Web more understandable to machines. This has many advantages such as allowing the machine to personalise the granularity of the content shown to the user or display data differently depending on context for example.

In order to semantically annotate a text resource, entities must generally be identified, disambiguated and then annotated (Nagarajan, 2006). Identification of entities is accomplished using an assortment of techniques depending on the domain. This often includes some natural language processing and in some cases (such as DBpedia Spotlight), ontology-driven extraction, which matches terms in the text with class instances in a populated ontology.

The disambiguation step is important when there are multiple entities that a term might refer to. For example, it may be unclear whether a text containing the word "apple" is referring to the object that is a member of the class "company" or the class "fruit". In this instance disambiguation may be accomplished by the referring to textual context. For example, the surrounding

text may refer to "oranges" and "pears" which are sibling entities in the reference data set to one of the candidates. This would be a strong indication that the text is referring to that candidate.

2.3.1 DBpedia Spotlight

DBpedia Spotlight is a tool which automatically annotates text with DBpedia resource URIs in order to link documents to the Web of Data via the DBpedia hub (Mendes et al., 2011). Spotlight has three stages of annotation: the spotting stage, candidate selection stage and disambiguation stage. The algorithm is also subject to significant amounts of configuration. This configuration allows for limitation of annotations to instances of a single class, multiple classes, or even the result of an arbitrary SPARQL query. For example annotations on a piece of text may be restricted to "politicians born between 1960 and 1970". This example query with explanation can be found in Appendix A. Annotations may also be configured based on their prominence ("how many times a resource is mentioned in Wikipedia"), topical relevance ("how close a paragraph is to a DBpedia resources context"), or contextual ambiguity ("Is there more than one candidate resource with similarly high topical relevance for this surface form in its current context?").

In order to spot phrases in the candidate text, Spotlight compiles a lexicon consisting of labels that are attached to each DBpedia resource, alongside redirects, disambiguation information, and other data extracted from DBpedia resources. "Surface forms" (i.e. phrases from the text that potentially match one or more DBpedia resources) are identified in the text using string matching from this lexicon, and then candidate DBpedia resources are selected based on the identified surface forms.

Inverse Document Frequency (IDF) is a measure used in Information Retrieval to estimate the specificity of a word or phrase, with the goal of giving more importance to more specific terms (Sparck Jones, 1972). IDF measures how often a term appears in resources. Spotlight introduces Inverse Candidate Frequency (ICF). When disambiguating terms, a term may be very specific and therefore have a high IDF, but also be very popular in a domain and as such be present in many or all of the candidate resources. ICF addresses this problem by calculating based only on the candidates for disambiguation so that terms that are specific to a small number of the candidates receive the highest value. This value is combined with the Term Frequency (how often the term appears in the resource) to find a measure of how important the term is to that resource.

2.4 Metadata and Structure

Exploring unstructured hypermedia such as the Web can be an inefficient experience. Vast quantities of data are easily found, but drawing coherent conclusions from this unstructured information can be difficult as resources begin to meld into one another and which piece of information lies in which resource becomes very difficult to keep track of. This experience of disorientation is often referred to as being "Lost in Hyperspace" (Theng, 1997).

One popular approach to taming hypermedia has been Adaptive Hypermedia Systems, which use resource metadata to structure the way in which resources are presented to the user. Adaptive Hypermedia Systems provide the user with personalised content using a model of the user's goals, preferences and knowledge and also a model of the content they are deliver-

ing (Brusilovsky, 2001b). Adaptive Hypermedia is particularly applicable to education given the varying interests, goals and prior knowledge of different students. Education was one of the first areas where adaptive hypermedia was applied (Brusilovsky, 2001a).

Many of these techniques however suffer from lack of re-usability and interoperability and were often restricted to a closed corpus of resources. This has been referred to as the open corpus problem:

> *"Is the applicability of the adaptive hypermedia techniques restricted by nature to closed corpus of educational resources or is it possible to develop open corpus adaptive hypermedia that will successfully work in such contexts as the Web and educational repositories?"* (Brusilovsky and Henze, 2007b)

CULTURA is an adaptive hypermedia system for "navigating digitised cultural heritage archives." (Bailey et al., 2012) It uses entity and relationship extraction techniques alongside social network analysis on the entities and relationships to add structure to the collection. This allows it to provide users with recommended resources. It also allows users to bookmark and annotate the resources. User trials found that professional researchers who knew the collection well were less likely to use the recommendations but found the bookmarking and annotating more useful.

CULTURA therefore forms the basis of inspiration to create a system which would allow researchers to create a structured collection of key resources while researching. Semantic Web technologies provide tools that allow for the collection to be structured without imposing a significant overhead on the user.

Chapter 3

Design and Implementation

3.1 Design

3.1.1 Technologies

Firefox Add-On SDK

The Firefox Add-On SDK allows developers to produce installable applications that add to the functionality of the Firefox browser. There are three major types of add-on: Extensions, Themes and Plugins. Extensions modify or add new features to the browser. This could be a button on the taskbar to change the colour of highlighted text or a warning icon that tells you every time you are asked for a password on an unencrypted page. Themes modify the appearance of the Firefox browser, changing the background image and colours of buttons and taskbars. Plugins render Web contents that the browser does not support natively. Adobe Flash Player would be an example of this.

Since early 2015[1], Add-Ons are developed using the JPM framework. JPM is based on Node.js and is installed using the Node.js package manager. Add-Ons are programmed using JavaScript with HTML and CSS for visual components. The Add-On is then executed using the Firefox JavaScript runtime environment.

The Add-On developed in this project is an extension as it adds new features to the browser. The framework provides developers with a number of APIs which provide simple access to some of the features of the browser. Some of the APIs used in this project were `Button`, `Tabs`, `SimpleStorage`, `Request`, `Selection`, `Panel` and `Sidebar`.

`Button` allows for the addition of new buttons on the taskbar. Figure 3.1 shows the buttons on the taskbar with the three new buttons appearing on the right-hand side.

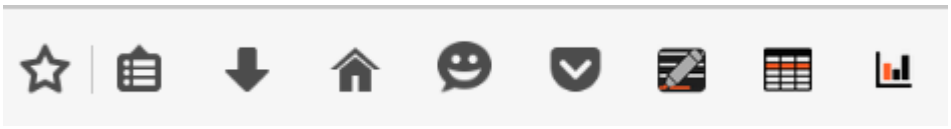

Figure 3.1: Taskbar Buttons

`Tabs` provides access to the tabs which are currently open in the browser and also allows for new tabs to be launched. In this project the Tabs API was used to obtain the URL for the highlighted text and also to launch new tabs when references were selected.

`SimpleStorage` is one of three main APIs that provide data persistence across multiple sessions of Firefox. It is more heavy-weight than preferences, which stores simple user preferences and more light-weight than the SQLite implementation. As the data that needed to be stored for this ap-

[1] https://blog.mozilla.org/addons/2015/02/26/jpm-replaces-cfx-for-firefox-38/

plication lent itself easily to a simple array structure, `SimpleStorage` was more than adequate.

The `Request` API allows the extension to make simple network requests. This was used to make a HTTP request to the DBpedia spotlight service.

`Selection` provides access to the selected pieces of the DOM of the current tab. This was used to extract the selected text for processing and saving.

`Panel` and `Sidebar` both provide windows for creating user interfaces for users to interact with the extension. The functionality of these interfaces is expressed using HTML, CSS and JavaScript. Naturally `Sidebar` provides a sidebar style window, where a `Panel` can be configured to appear anywhere on the screen and be of any size.

Firefox or Chrome

Firefox was chosen as a platform for this project over Chrome largely because of the extra freedoms that it affords developers. Chrome extensions are “sandboxed” and unable to access files created by other extensions or programs. This freedom was useful for prototyping to allow for a number of possible implementation paths, such as having a separate python application to provide data visualisation for example. However, there is no reason that the current extension could not be implemented in Chrome now and if it were to be released there would ideally be an implementation for both browsers.

D3.js

D3.js (Data-Driven Documents) is a JavaScript library focused on manipulating HTML documents and Scalable Vector Graphics (SVG) to represent data[2]. It provides a number of abstractions and tools to facilitate this goal.

Selections allow collections of DOM objects to be selected and modified together. Objects can be selected using containment and attribute values, as well as the more standard tag name, class and ID. Modifications that can be made include adding event listeners, setting attributes and styles, and changing inner HTML.

Dynamic Properties allow styles and attributes to be expressed as functions rather than simply as constants. This allows the colour of an object to be changed based on dynamic data for example.

D3.js allows nodes that are being updated to be treated differently to those being added (entering) and those being removed (exiting). This gives greater control over transitions when data is updated.

D3.js manipulates well established Web standards, HTML, CSS and SVG, and does not introduce any proprietary or new visual representations. This is helpful in a number of ways, not least that the browser's element inspector remains a helpful debugging tool as the nodes are all natively supported.

Animated transitions are provided by interpolating styles and attributes over time. D3.js transitions also work alongside CSS3 transitions.

[2] https://d3js.org/

DBpedia Spotlight

DBpedia Spotlight services can be accessed by using the Web service or by installing it and running a new instance of the server. It provides a number of configuration possibilities in order that it can be tailored to a broad range of situations.

Spotlight allows the user to select which sets of entities it should choose its annotations from. This can allow it to be more accurate when working in a specific domain, or to restrict its annotations to people for example and completely disregard all other things. Entities may be whitelisted or blacklisted based either on their class or an arbitrary SPARQL[3] query. Restricting by SPARQL query allows quite powerful and specific restrictions such as "politicians born between 1960 and 1970"[4].

The *Support* parameter provides a measure of resource prominence. In some situations, it may be advantageous to restrict obscure entities from annotations. Spotlight does this by counting the number of inlinks to the Wikipedia page on which each resource is based. Those with a small number of inlinks are assumed to be obscure. The support parameter sets the minimum number of inlinks a resource should have to be used for an annotation.

Topical Pertinence can be restricted based on the similarity score returned from the disambiguation step. This similarity score is high for terms which appear in paragraphs where there are many other terms that relate to the selected DBpedia resource.

[3] SPARQL is a query language designed for querying RDF triple-stores like DBpedia.
[4] See Appendix A

It is possible however that more than one resource has a high topical pertinence for a given term. The *Contextual Ambiguity* provides a measure of the difference in similarity scores for the first and second ranked resources. This gives an idea of how close and therefore ambiguous the decision was.

Spotlight also provides a combination score of *Disambiguation Confidence*, which takes into account the Support, Topical Pertinence, and Contextual Ambiguity and returns a number between 0 and 1 which represents its confidence that the annotation is correct. The parameter is designed such that setting a confidence level of 0.7 should eliminate 70% of incorrect annotations. Spotlight also provides a feature where it can return all the candidate resources for each term alongside their confidence and other metric scores and let the application decide what to do with the information.

3.1.2 Goals and Objectives

The main objective in the development of this application is to create a lightweight extension which allows the user to curate a tagged collection, while researching for example, without adding a large overhead to the user. In order to accomplish this, there are three main goals that must be realised.

1. Users must be able to highlight a piece of text, annotate it with semantically structured tags, and save the reference and annotations without disrupting or fundamentally altering the research process.
2. Users must be able to look back over the references and annotations that were collected, and return to them if desired.
3. Users must be provided with visual feedback as to the structure of the annotations created during the research process.

3.1.3 Application Architecture

As previously described the application is built using the Firefox Add-On platform and integrates directly into the browser. As in Figure 3.2 the whole application is contained within the Firefox extension except for the request to the DBpedia spotlight service. The extension consists of three main components: annotation creation UI, annotation visualisation UI, and annotation storage. Each of these is managed by the main controller, which also manages the request made to the DBpedia Spotlight service.

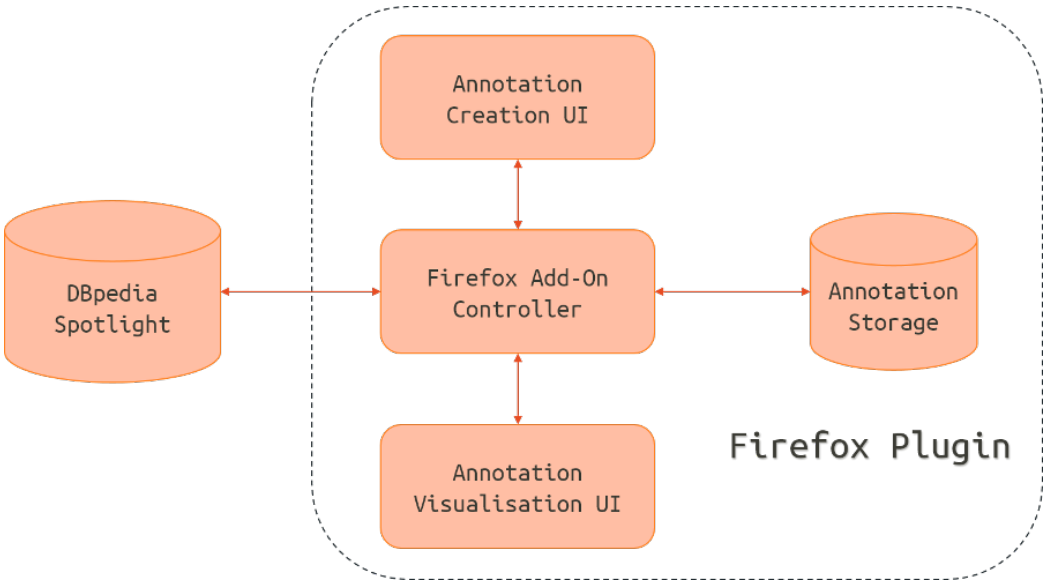

Figure 3.2: System Architecture Diagram

Add-On Scripts and Content Scripts

The architecture of the Firefox platform is such that there are two distinct types of JavaScript scripts which form the basis of an extension's structure. Content scripts run in the context of a tab or UI window and therefore have

access to the DOM of that context. Content scripts are therefore used to change the appearance and functionality of DOM objects in tabs and UI windows. They can be loaded into already existing documents on the Web to provide added functionality, or loaded into UI windows that are part of the application.

Add-On Scripts are not run in the context of any DOM and therefore have access to no DOM elements, however they do have access to Firefox SDK APIs which are not available to content scripts. Add-On scripts therefore implement the main Add-On logic and provide flow control for the application. Add-On Scripts also provide services to the content scripts such as the passing of information between content scripts.

Event Driven Architecture

The Firefox platform is also based on an event-driven architecture. Events occur when an object changes in state. Examples of this could be a form being submitted, a network request completing, or a new browser window being opened. These are all built in events in the framework, but events may also be defined by adding event listeners to objects. These events can then be used to trigger communication with the Add-On script and request a service for example.

3.2 Implementation

3.2.1 Highlighting

The first step in using the application is selecting a piece of text and clicking the "Save Resource" button. This launches the sidebar and loads a content script into the current tab to highlight the selected text as shown in Listing 3.1.

```
function saveResource(state) {
  sidebar.show();
  tabs.activeTab.attach({
    contentScriptFile: [self.data.url('js/highlighter.js')]
    });
}
```

Listing 3.1: On clicking the Save Resource UI button.

The content script searches for the text in the document and highlights it by changing its background to yellow as shown in Figure 3.3.

3.2.2 Generation of Tags

When the Add-On script receives the selected text, it makes a request to DBpedia rest Web service. As seen in Listing 3.2, the body of the request contains three parameters: text, confidence and support. The text is the input text that is to be annotated. Confidence and support are the parameters described in section 3.1.1 that indicate a minimum confidence level for annotations and a minimum number of Wikipedia inlinks respectively. This request could be adjusted to restrict the DBpedia resources referenced by

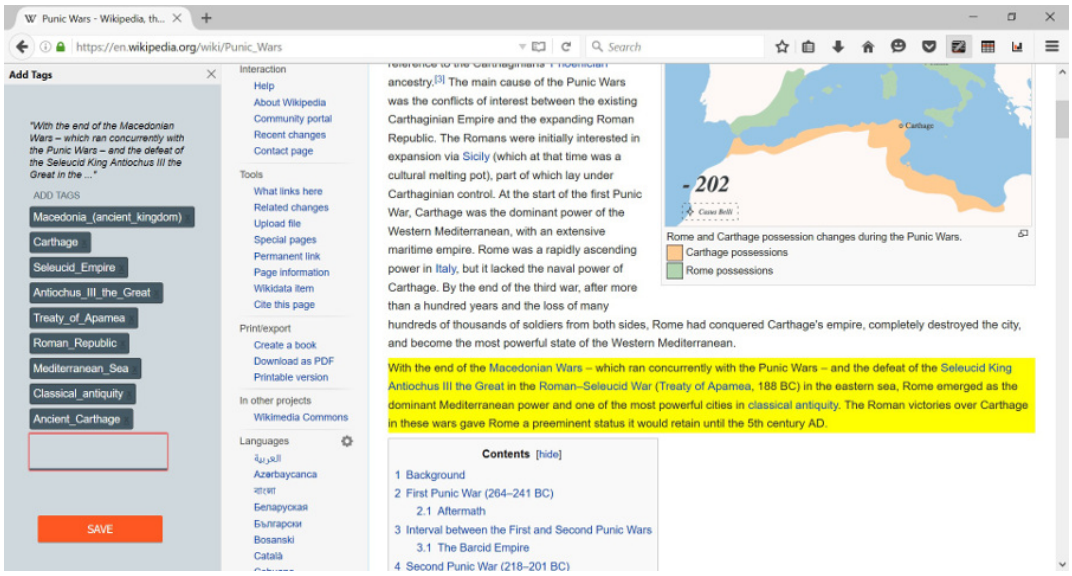

Figure 3.3: Text highlighting and sidebar

including a "types" or "sparql" parameter. It could also be adjusted to return all candidates for the annotation of each term alongside a confidence level for each one by making the request to the `/candidates` endpoint rather than `/annotate`. Currently these settings are not made available to the user to adjust as they are hard-coded into the application. Since the application is designed to work in all domains as much as is possible, no restrictions are placed on the type of resources that are used for annotation. The confidence and support values were chosen such that only the most unconfident and obscure annotations were omitted, leaving the user to decide for more contentious instances.

```
var request = require("sdk/request").Request;
request({
  url: "http://spotlight.sztaki.hu:2222/rest/annotate",
  content: {text : input,
    confidence: 0.5,
    support : 10},
  onComplete: function (response) {
    // Parse response to obtain annotations.
  }
}).post();
```

Listing 3.2: Request made to DBpedia Spotlight Web service.

3.2.3 Saving of Annotations

Once the request returns a response, an event is triggered which launches the onComplete function in the request configuration as in Listing 3.2. The response consists of the input text with annotations provided by added HTML links to the relevant DBpedia resources. This response is then parsed. The DBpedia resources are extracted and any duplicates are removed. All DBpedia resource URIs follow the structure `http://dbpedia.org/resource/<resourceID>`. The first section of these URIs is removed, leaving just the unique resource ID, which is added to an array and passed to the sidebar to be presented to the user.

When the sidebar receives the array of DBpedia resources, it pre-fills the tag manager with the resources and loads the tag manager onto the sidebar as can be seen in Figure 3.3. The tag manager being used is a jQuery plugin by Max Favilli[5]. Tags may then be added or removed by the user before clicking

[5]http://welldonethings.com/tags/manager - Licenced under the Mozilla Public Licence

the save button to save the resource reference and tags. These tags are passed back to the Add-On script where they are saved alongside the text, a timestamp, and the URL of the resource.

These annotations are stored using the simple-storage API provided by the Firefox SDK as seen in Listing 3.3. They are stored using a 2-dimensional array[6] with the primary index referring to the sequence of annotations and the secondary index referring to each piece of data in the annotation (e.g. index 0 refers to the timestamp and index 3 refers to the tags.) Firefox provides a quota of 5MB of simple-storage to each Add-On. Each annotation takes up approximately 1KB of storage space, so there should be space for approximately 5000 annotations. If the storage quota is reached, the user is notified that some annotations must be deleted before continuing.

```
var simpleStorage = require('sdk/simple-storage');
sidebar.port.on("annotation", function(data) {
  annotation = [simpleStorage.storage.count, Date.now(), tabs.
      activeTab.url, data[0], data[1]];
  simpleStorage.storage.annotations.push(annotation);
  sidebar.hide();
});
simpleStorage.on("OverQuota", function () {
  window.alert("The storage quota has been reached. Please delete
      some annotations before continuing.")
});
```

Listing 3.3: On receiving the annotations from the sidebar, the Add-On script stores them in simple-storage

[6] In JavaScript multi-dimensional arrays are implemented by simply putting arrays inside each other.

3.2.4 Viewing Annotations

Users may view the annotations that they have created using the View Annotations panel as shown in Figure 3.4. Here the user may delete any of the annotations, or click on them to open a new tab at the URL referenced by the annotation. The Add-On script will the load the highlighting script again to highlight the referenced text and automatically scroll to focus on it.

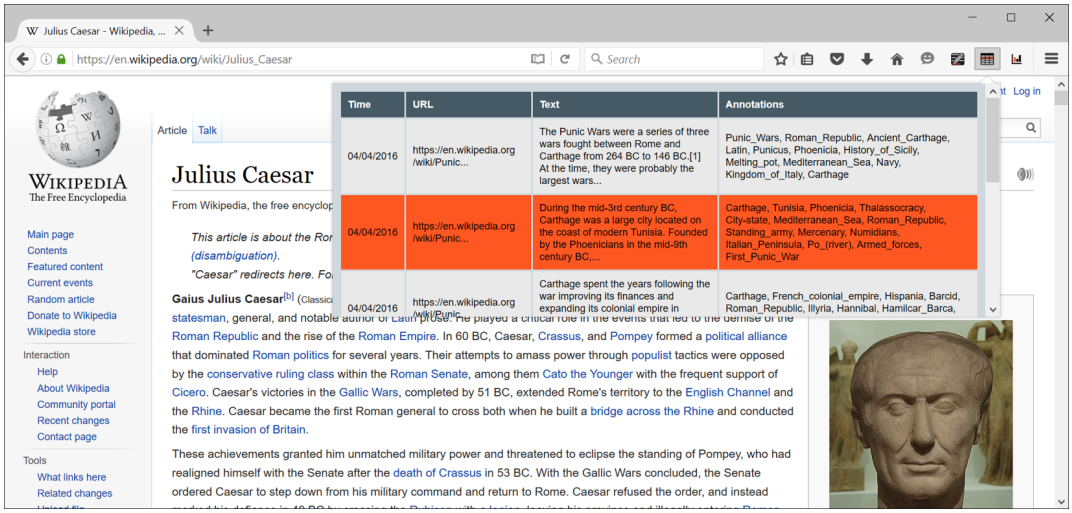

Figure 3.4: View created annotations

3.2.5 Tag Visualisation

The final function of the application allows the user to visualise how the distribution and frequency of tags changed over the course of the research. It can be launched by clicking the right-most action button on the taskbar as shown in Figure 3.5.

This is accomplished with a bar chart created using the D3.js JavaScript

library[7]. While waiting on the Add-On script to pass on the tag data, the graph content script draws the axes using the D3.js SVG axis function as shown in Listing 3.4.

```
var y = d3.scale.linear()
    .range([height, 0]);
var yAxis = d3.svg.axis()
    .scale(y)
    .orient("left")
    .ticks(10, "d");
```

Listing 3.4: D3.js axis creation

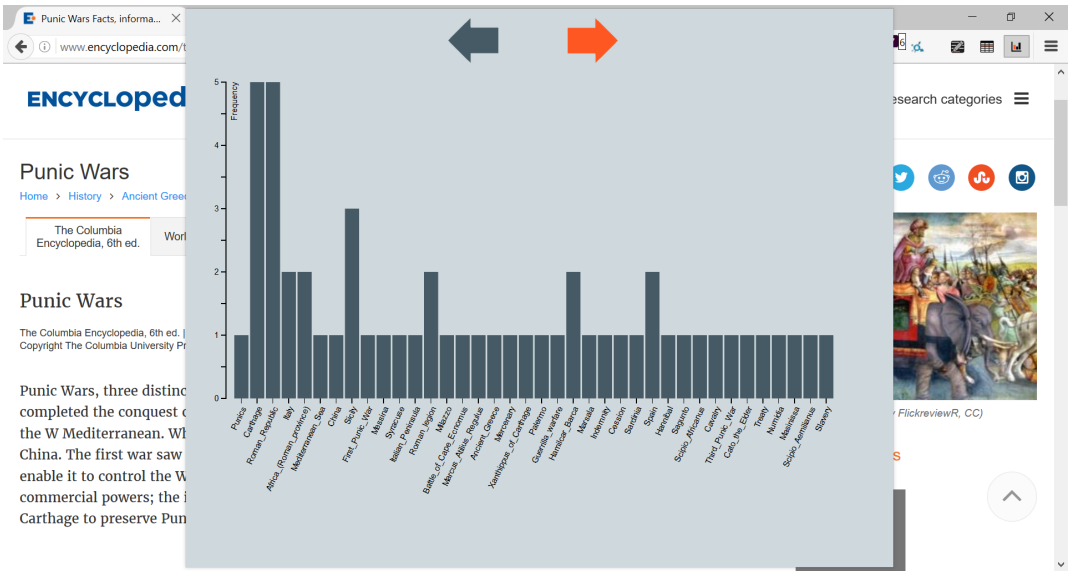

Figure 3.5: Tag frequency visualisation

When the annotations are sent from the Add-On script to the graph script, the graph script retrieves all the tags from the first 5 annotations and sorts them into an array of pairs of tag name and how often the tag has appeared. It then charts the tags on a bar chart as shown in figure 3.5. The arrows allow the user to shift the window of graphed annotations forward and backwards

[7] http://d3js.org/

to see how the frequency of particular tags changed over the course of the research. For example, when the right arrow is clicked, the script will compile and graph all the tags from the 2nd to the 6th annotation, and so on until the most recent annotation has been reached. The left arrow will bring the user back through the annotations until the first annotation is reached.

Chapter 4

Evaluation

In order to determine to what extent the objectives of this project were accomplished, an evaluation was carried out where test subjects used the application for a short while and answered questions about their experiences.

4.1 Evaluation Method

In order to realise the goals of this project as laid out in Section 3.1.2, the evaluation must demonstrate the following:

- The user interface is intuitive and easy to use.
- The automatically provided annotations by and large accurately reflect the content of the resource.
- Users can easily add tags to annotations.
- Users can review previous annotations and return to the resources referenced.

- Users are able to see the change in focus of the tags over time.
- Annotating resources did not significantly detrimentally interfere with the research process.
- Users found the tools provided to be beneficial to the research process in at least some circumstances.

For the evaluation, users were set two topics to research using the application and also given some time at the end to review the collection that they had made and the tags that they had generated. They then completed a brief questionnaire to assess their experience with computers and also a System Usability Scale questionnaire. Finally, they took part in a brief interview consisting of more broad questions about the system and completed a short questionnaire focused on the requirements for the system.

Before beginning to use the system, participants were given a very quick introduction to the main functionality of the system. Care was taken to ensure that the information imparted in this introduction could be easily recreated in the form of some tooltips that could appear the first time a new user uses the application.

The purpose of the participants researching the two topics was to allow them to become familiar with all the features of the application. They were told about the potential use cases that had been identified for the system and were given a very brief introduction to the function of the three action buttons in the application. The first research period was on a topic of their own choosing that they thought themselves to be reasonably expert in. The second topic that they researched was chosen for them and was something that they knew relatively little about. If the first topic was specific, the second

topic was broad and vise versa. This was in order to give the participants an opportunity to use the system in both a broad and a specific field.

The computer proficiency questionnaire was a quick way to get a rough estimate of how comfortable each of the participants were in using computers. No broadly accepted standard questionnaire could be found for this purpose and any similar questionnaires that could be found were generally too long and were designed to be the focal point of the research itself, such as measuring the computer proficiency in a country. In order to get a quick and easy measurement, participants were asked if they had completed any formal training in a Computer Science related subject and asked to give an indication of how much time they would spend using a computer. Finally, the participants were asked to respond to a subset of six of the Likert items from the computer confidence section of the questionnaire in (Fogarty et al., 2001). Care was taken to ensure that 3 of the Likert items were phrased positively (i.e. "Disagree" inferred less proficiency and "Agree" inferred more proficiency) and 3 were phrased negatively as this is considered best practice. The full questionnaire can be seen at Appendix C.

The System Usability Scale (SUS) is a Likert scale used for quickly and easily assessing a system's usability (Brooke et al., 1996). The SUS questionnaire was completed after participants had finished using the system and before the interview began as it is indicated that it is generally used "before any debriefing or discussion takes place." The SUS produces a single value between 0 and 100 which represents the overall usability of the system where a score above 68 is considered above average. The full SUS questionnaire used can be found in Appendix B.

The interview section of the evaluation allowed participants to give more

qualitative information about their experiences using the application. There were 5 main questions or talking points:

1. Which of the identified use cases do you think the application is more suited to? Why?
2. Did you find the application more useful in a known or an unknown domain? Why?
3. Are there any use cases that have not been identified that you think this application is well suited to?
4. Of the features that have been identified but not yet implemented, which do you think is the most important? Why?
5. Are there any other features that have not been identified that you would like to see implemented?

This interview style section allowed participants to discuss potential uses and features for the system that had not been thought of yet.

The final questionnaire that participants filled out consisted of 7 questions where each one addressed one of the requirements of the system directly. Answers were free text. The qualitative approach was taken again to allow participants to express thoughts that may not have been previously thought of. Since the sample was so small, there would have been very little statistical power in a quantitative approach. The questions were as follows:

1. Did you find the system intuitive and easy to use?
2. To what extent did the automatically provided tags reflect the content of the resources that were being annotated?
3. Were you able to easily add your own tags to annotations?

4. Did you find it easy to refer back to annotated resources?
5. Were you able to see the change in focus over time of the tags using the graphing feature?
6. To what extent did you feel that the annotation process interfered with the research process?
7. Do you think the tools provided by the system are beneficial to the research process in certain circumstances?

4.2 Evaluation Procedure

Participants were first given a brief introduction to the application and its features and also told the two main use cases that had been identified: a researcher looking to track the research process and an educator looking to build a collection of resources for students.

Each of the participants were asked to choose a topic that they considered themselves to be relatively expert in and spend 15 minutes researching the topic. They were told when they had 2 minutes left so that they could look back over the collection and over the graphs that the system generated. When finished the 15 minutes, they were given a topic that they were not particularly expert in and they repeated the research process for another 15 minutes, again given warning of the final 2 minutes. It was also ensured that out of the two topics that the participants researched, one was specific and the other broad.

Having completed using the application, the participants filled out the SUS and Computer Proficiency questionnaires. Next they answered the ques-

tions in the interview format, and finally they filled out a brief qualitative questionnaire designed to directly address each of the requirements for the system.

4.3 Results

4.3.1 System Usability Scale

There were only 5 participants who took part in this evaluation, therefore the results of the SUS could not be considered statistically robust and the sample would certainly not be considered representative of the general population. Three of the participants had engaged in formal training in the general area of Computer Science and two had not. The value returned from the scale for these 5 participants was 75.4. In the SUS scale 68 is considered the average mark.

4.3.2 Interview

Which of the identified use cases do you think the application is more suited to? Why?

For reference, the identified use cases presented to the participants are as follows:

- A researcher looking to track the research process
- An educator looking to build a corpus for students

All five participants mentioned teachers in their responses to this question. Three of the participants felt that the system would be useful for the purposes of lesson planning, that a broad topic could be researched initially and resources relating to sub-topics could be filtered out easily by tag when looking to plan a lesson on the given sup-topic. This falls somewhat between both use cases, but mostly on the side of research. One participant mentioned that in their experience, there were a lot of online resources for teachers that could be hard to keep track of and that going through them and categorising them once would be helpful. One participant also felt that it would be helpful to keep track of references when researching for a research paper.

One participant felt that the system was best suited to collecting a set of resources to hand over to students and mentioned that the fact that the resources were left in context would provide a lot of freedom to students to explore around the topic.

Did you find the application more useful in a known or an unknown domain? Why?

Responses to this question were more varied, with three participants feeling the system was more suited to an unknown domain and two to a known domain. One participant who felt that the system was more suited to research in an unknown domain felt that it was more difficult to generate ones own tags in an unknown domain and that it was nice to be able to have the tags generated and allow patterns to emerge. Another felt that it was easier to select short concise segments in the unknown domain.

One participant who felt the system was more suited to the known domain simply felt that it was easier to know what was important to select and what wasn't in the known domain.

Are there any use cases that have not been identified that you think this application is well suited to?

The participants identified a number of possible use cases for the application. One participant felt it could be useful in fashion market research where there can be many conflicting statements about the same market segment. They felt it would be helpful to be able to compile these all in one place. In a similar vein, another participant felt that the application would also be helpful in the corporate world where lengthy reports often need to be summarised.

Two participants felt that the application could be useful in education. One proposed that it could be helpful in teaching the research process by giving users feedback on which resources led them in new directions for example. The other participant felt it could be used as an educational tool that students could learn by finding resources and that the application would help in making connections between resources.

Of the features that have been identified but not yet implemented, which do you think is the most important? Why?

The features presented to the participants as having been identified are as follows. They are divided up according to the use cases that require them: For researchers

- Further data visualisation such as the connections between tags and the closeness of tags.
- Implementation on many devices and annotations stored on a server such that work could be continued on a mobile device etc.

For educational curriculum development

- Integration into an educational adaptive hypermedia system to support student navigation through the resources.
- Support for further metadata types such as the type of resource (e.g. summary, question, solution etc.) and a note field.

For both

- Filtering the collection by tag and by search.
- Support for further types of media such as images, audio and video.
- Further work to ensure that when resources are updated, the reference is not lost or made invalid.

Two of the participants felt that the most important feature to implement was support for other types of media. In particular, the topics that these participants chose were quite visual in nature, so the lack of support for images and video was quite restrictive.

Two participants felt that filtering by tag and by search was most important. In particular, for the use case of a teacher using the system to collect resources on a broad topic, this feature would make it much more easy to filter for resources connected to a sub-topic for a single lesson for example.

One participant felt that further data visualisation was most important, to

allow for patterns in the research to be seen more easily.

Are there any other features that have not been identified that you would like to see implemented?

A potential feature that participants mentioned was the ability to edit annotations by removing and adding tags. One participant also proposed that the system could recommend possible resources to the user.

4.3.3 Requirement Focused Questionnaire

Did you find the system intuitive and easy to use?

All participants indicated that they found the system easy to use after the initial explanation. One participant suggested a right-click to save function.

To what extent did the automatically provided tags reflect the content of the resources that were being annotated?

Three of the five participants felt the generated tags reflected the resource content well, the other two found that it varied somewhat depending on the content of the resource. One participant commented *"This varied, it seemed biased towards place names and names of persons like 'New York', 'Karl Marx' rather than descriptive nouns like 'democracy', 'feminism', 'equality' etc. This apparent bias may work positively in certain research domains."*

Were you able to easily add your own tags to annotations?

All participants felt that they were able to use this feature easily. Two participants mentioned however that sometimes they would miss a tag by clicking the save button before they added the tag to the list.

Did you find it easy to refer back to annotated resources?

No participant found any trouble in using the system to bring them back to previously annotated resources. However, since the participants were only researching for 15 minutes at a time, the number of resources they collected was relatively small (in the order of 15 - 20). It is likely that locating the correct resource to return to would become more difficult as the collection gets larger.

Were you able to see the change in focus over time of the tags using the graphing feature?

Of the 4 participants who tried this feature, one did not find the change in focus clear, and another found that as long as they were diligent in removing irrelevant tags, it became very clear. The final two had no difficulties making out the change in focus of tags.

To what extent did you feel that the annotation process interfered with the research process?

Three participants felt that the system did not interfere with the research process. One participant felt that the process became more considered and structured, but that this may result in missing some "outside-of-the-box" ideas. The final participant felt that the adding of custom tags forced greater thought about text selection and fewer unnecessary sources.

Do you think the tools provided by the system are beneficial to the research process in certain circumstances?

All participants felt that the tools provided by the system were beneficial in certain circumstances. University essays, dissertations and lesson planning were all mentioned as circumstances where the system would be useful. One participant thought that the system would work better linked to a specific library of books or articles as the tagging could provide *"more intelligent searches as later users can search within the library for text intelligently tagged by humans, rather than simple searches for words and related words or sentences within a body of text."* Another participant felt that the system would be most useful in domains that were reasonably well known to the user so that they could remove unnecessary tags effectively.

Chapter 5

Future Work

This section describes a number of ways in which this work could be furthered in the future. It is split into two main sections. The first will cover future work that is thought to be straight forward with a clear implementation path. The second will cover larger pieces of work which may have many possible implementation paths. Naturally, this is not a black and white distinction in all cases and some instances such as search functionality may be very straight forward in their simplest form and also arbitrarily complicated as demands on functionality increase. For the sake of clarity a category has been chosen for each piece of work based on estimated required complexity for adequate functionality in this system.

5.1 Short-term Work

In this case it seems that the most simple work is also the most pressing in need for the system. The work in this section is by and large the work that is

likely to bring the most return on investment.

User-adjustable DBpedia Spotlight Settings

As described in Section 3.1.1, DBpedia Spotlight provides a number of configuration settings which allow for restrictions to be made on the DBpedia resources that Spotlight will use to annotate with, and also to set a minimum confidence level for annotations. Currently these configuration settings are hard coded into the application, however it may be beneficial to users to be able to access these settings. It is important to this application that the main user flow is free from distractions to reduce cognitive load. For this reason these settings could be adjusted on a settings page, perhaps when the user is beginning a new piece of research. One participant found that they were getting a large number of irrelevant tags when researching their chosen topic. This type of problem could potentially be mitigated by allowing the user to select types of DBpedia resources that are not relevant.

Both the Confidence and Support values could be set using a simple sliding bar with the range of Confidence being 0-1 and Support 0-1000. Whitelisting and blacklisting DBpedia types by type name and SPARQL query would be slightly more cumbersome. Types could be shown to the user in a hierarchy with check boxes for the user to select[1]. The SPARQL input box need only be a text input box, however ideally it would include SPARQL syntax highlighting and some syntax checking.

[1]This is implemented in the DBpedia Spotlight demo found at http://dbpedia-spotlight.github.io/demo/

Search

Currently the application does not enable users to search through the collection of annotations. For larger periods of research this would become a necessity. It would also be extremely helpful for filtering annotations by which tags they contain as mentioned by a number of participants in the evaluation. A simple text search to search the stored text and a tag search to allow annotations to be filtered by tag would be relatively trivial to implement. Ideally the tag search would support Boolean operators to allow users to express the union and complement of tag sets rather than just the intersection.

This is also an area where the added structure of using DBpedia resources as tags becomes helpful. All the DBpedia classes and sub-classes could be included in the application as pre-defined tags. These tags could then be added to a search to refer to all sub-types of the class. For example "SportsEvent" is a DBpedia class. Adding the tag SportsEvent to the search could return all annotations with a tag that is a sub-type of SportsEvent.

User Interface Adjustments

There are some small user interface adjustments identified by evaluation participants that could make the application more intuitive and reduce overhead in using it. Firstly, after highlighting text, a natural next step would be to right-click. Adding the "Save Resource" function to the right-click menu would help with the natural flow of using the application for some users.

The second is on the sidebar when adding tags, some participants found that they would type a custom tag and then click save, but because the tag hadn’t been added to the tag list, the tag would be lost. This could hopefully

be remedied by adding a "save tag" button beside the input to make the function more clear.

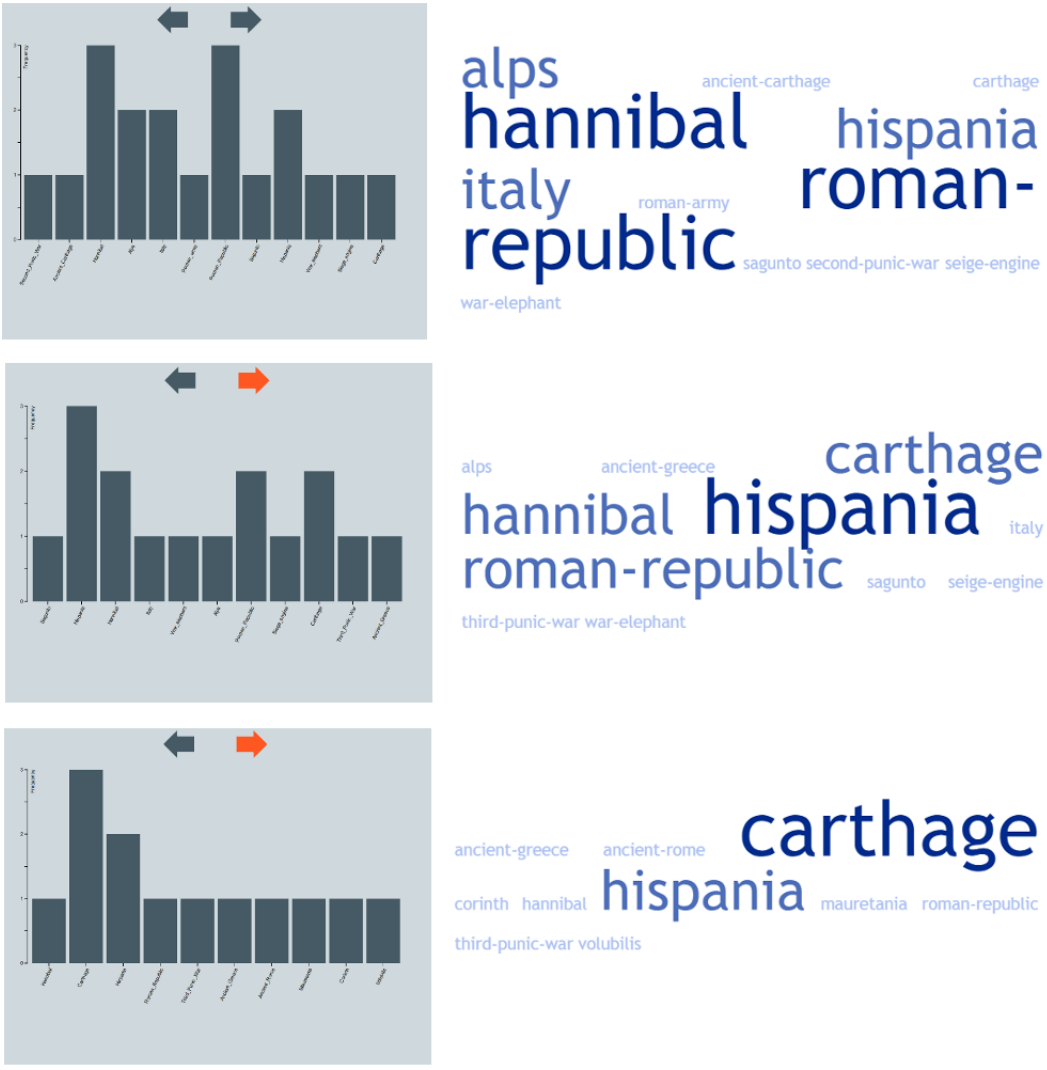

Figure 5.1: Tag visualisation graph vs cloud

Further Data Visualisation

Some participants found the visualisation somewhat unclear and found it difficult to make out how tags had changed over time. This is something

that warrants a significant amount of development to accurately and clearly display all the helpful information that can be garnered from the structure of the semantic tags, however as a first step, a tag cloud instead of a bar chart may convey the information more clearly.

Figure 5.1 shows on the left hand side the graph of the tag frequency as the user moves through the annotations and on the right hand side, the same data expressed in a tag-cloud form. The tag cloud style shows more clearly which tags are most prominent and therefore may provide more clarity on how the tags of most importance changed over the course of the research period.

Further to the visualisation of the frequency of different tags, it would be helpful to have a visualisation of which tags were similar to each other. DBpedia Spotlight Rel8 adds another endpoint (`/related`) to the Spotlight service which when passed a DBpedia resource as a parameter, returns a list of all related DBpedia resources. This could be graphed using a hierarchical edge bundling algorithm as seen in Figure 5.2[2]. Tags could be bundled by their DBpedia parent classes.

Finally, rather than the annotations being bundled arbitrarily simply based on their place in the sequence of annotations, some analysis could be done on the closeness of the tags (based on information from DBpedia) to identify when the research has changed significantly in its focus. Tags from annotations that are both close in time and in topic could be grouped together, potentially making the visualisation of the flow of the research process more clear.

[2]Image from http://bl.ocks.org/mbostock/1044242

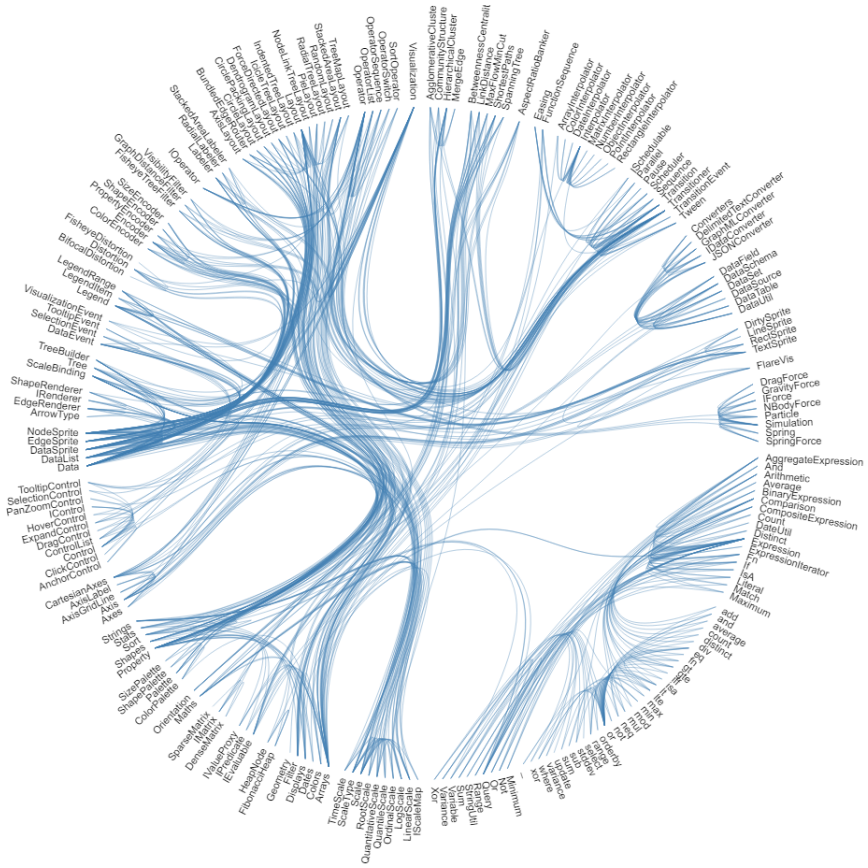

Figure 5.2: Hierarchical Edge Bundling

Support For Further Metadata

Support for different types of metadata such as adding a note to an was mentioned a number of times as a possible feature during the evaluation. If the focus of the application was to provide teachers with a tool so compile a collection of resources for students, it would be helpful for them to be able to classify what kind of resource it is. For example a resource may be classified as a summary, question, solution, example, primary source, secondary source etc.

5.2 Long-term Work

Ensuring References Remain Valid

Currently if the text referenced by an annotation changes, the application may not be able to find that text again. This is unideal, particularly if the change in the text is small and insignificant to the overall meaning. As a first step, the document could be searched sentence by sentence to match to sentences in the saved text by word frequency. When a partial or full match is found, the surrounding sentences could be compared to the other sentences in the saved text, and so long as the match was close enough, the segment could be re-saved. Failing that, the user could be asked if they wish to update the reference or to delete it. With further research, this could become more reliable, however user involvement would always be important to ensure the text still fulfils its purpose for the user despite being changed.

Cross Device Compatibility

Allowing users to research and make annotations from any device and also to have their collections synced across devices would add significant benefit for some users. This would require moving the storage of the local system and onto a server. A simple SQL database would serve the functionality easily, however care would of course have to be taken to maintain user data privacy.

For mobile devices a dedicated application would probably be the best approach. The application could be called from the Internet browser of choice using the "share" function implemented on both iOS and Android, and the

application could then present the user with the automatically generated tags and an option to add custom tags. The app could also present the collection and visualisations to the user.

On the desktop, supporting the Chrome browser would be relatively straight forward as both frameworks are built on JavaScript, HTML and CSS.

Multiple Users

Multiple users working on the same collection is a natural follow on from supporting online storage of annotations. Once permissions infrastructure has been set up allowing users to add other users to projects etc. there is no significant structural impediment to allowing multiple users work on the same project. However, there are some features which would need to be added. It would need to be recorded which user made each annotation. It would also be helpful if users were to be notified when multiple users attempt to annotate the same resource.

Automatic Recommendation

This application lends itself to providing recommendations to the user for tags to add and also for resources that might be helpful. Tags might be recommended to the user on the basis that they had added them to similar resources in the past (with the similarity derived from the tags that the resource generates). Users could also be recommended to retrospectively add tags to old annotations which are similar. Another way in which recommended tags could be identified is by analysing the tags that have been generated and the DBpedia resources that are related to them. If there are

resources that are closely related to a lot of the tags that have been generated, they could be relevant to the resource.

Resource recommendation was mentioned by one of the evaluation participants as a potentially helpful feature. It could be accomplished on the basis of the resources which other users have annotated after annotating resources with similar tags to the ones used by the user. Custom tags could also be suggested in this way, based on the custom tags of other users.

Adding Structure to Custom Tags

Currently, tags added by users themselves are not linked to any DBpedia resource and so lack the extra structure and meaning that that adds to the tag. This could be improved for at least some custom tags by adding a DBpedia resource search to the tag input box which would make suggestions of DBpedia resources as the user typed. This could be accomplished by making Ajax requests directly to DBpedia, however it would require some development to ensure that results were responsive and accurate, and that not too many results were returned.

Supporting Images, Audio, and Video

Supporting of images, audio, and video was something that a number of evaluation participants mentioned as being potentially very helpful. As a first step, it would not be a huge job to save the source file location of an image, audio file or video file as the reference to find that part of the page again. However, this does have some limitations as source would only be expressed in that way for audio and video in HTML5. There also arises

a difficulty in generating tags that would effectively describe the piece of media. Analysing captions and alternate text would be an effective first step, however they are not always present and therefore surrounding text may have to be relied upon. Being able to reference a specific temporal section in an audio or video resource would also be a helpful feature.

Support for PDF

Since the vast majority of academic papers are in PDF form online, adding support for PDF documents would be a significant feature for this application. Adding this support would likely not be trivial, however it has been done quite successfully by many other services such as A.nnotate and Diigo, described in Section 2.1.1.

Chapter 6

Conclusion

The aim of this project was to develop a system which would allow users to create a structured collection of key resources while researching on the Web, without adding prohibitive overhead to the research process. Another key component of the project was that users could target a specific piece of text anywhere within a web-page rather than just the web-page itself. The system developed to accomplish this was a Mozilla Firefox Extension which worked with DBpedia Spotlight to automatically generate tags relevant to the piece of text that the user wished to save a reference to.

The project built on a number of influences from the state of the art. Web annotation has been around for a long time and many different approaches have been made. These annotation tools formed the basis of the requirements for the system. The additions made by this project build on recent developments in natural language processing and the Semantic Web in the form of DBpedia and DBpedia Spotlight in particular.

Evaluation participants who were completely new to the system were able

to use it with minimal introduction and found that overall, the system did not negatively impact the research process significantly. By the use of DBpedia resources as tags, the collection generated was structured with hierarchical, semantic tags, allowing for significant potential further analysis. Feedback of that structure to the user however was not as clear as it could be and further development of that feedback would be a key part of this work in the future.

The system has a number of areas which warrant future work. The main area of weakness as identified in the evaluation was the visualisation of tag structure. There is much left to explore as regards the benefits that using structured semantic tags over regular unstructured tags provides to visualisation of structure and other aspects of the application. Other areas that were mentioned as particularly important by evaluation participants were support for further types of media and search functionality for the collection view.

Bibliography

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). *Dbpedia: A nucleus for a web of open data*. Springer.

Bailey, E., Lawless, S., OConnor, A., Sweetnam, S., Conlan, O., Hampson, C., and Wade, V. (2012). Cultura: Supporting enhanced exploration of cultural archives through personalisation. In *the Proceedings of the 2nd International Conference on Humanities, Society and Culture, ICHSC*.

Berners-Lee, T. (2006). Linked data. *online posting, Feb*.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, 284(5):28–37.

Bizer, C., Heath, T., and Berners-Lee, T. (2009a). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009b). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.

Brusilovsky, P. (2001a). Adaptive educational hypermedia. *International PEG Conference*, 10:8–12.

Brusilovsky, P. (2001b). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11:87–110.

Brusilovsky, P. and Henze, N. (2007a). Open corpus adaptive educational hypermedia. In *The adaptive web*, pages 671–696. Springer.

Brusilovsky, P. and Henze, N. (2007b). Open corpus adaptive educational hypermedia. In *In Brusilovsky P, Kobsa A, Nejdl W (eds), The Adaptive Web: Methods and Strategies of Web Personalization, LNCS 4321*, pages 671–696. Springer.

Buddulf, M. (2004). Introducing del.icio.us. *XML.com*.

Diigo (web). Diigo homepage. www.diigo.com. Accessed: 2016-05-15.

Fogarty, G., Cretchley, P., Harman, C., Ellerton, N., and Konki, N. (2001). Validation of a questionnaire to measure mathematics confidence, computer confidence, and attitudes towards the use of technology for learning mathematics. *Mathematics Education Research Journal*, 13(2):154–160.

Heymann, P., Koutrika, G., and Garcia-Molina, H. (2008). Can social bookmarking improve web search? In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 195–206. ACM.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia

spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM.

Mrabet, Y., Gardent, C., Foulonneau, M., Simperl, E., and Ras, E. (2015). Towards knowledge-driven annotation. In *AAAI*, pages 2425–2431.

Nagarajan, M. (2006). Semantic annotations in web services. In *Semantic Web Services, Processes and Applications*, pages 35–61. Springer.

Noll, M. G. and Meinel, C. (2007). *Web search personalization via social bookmarking and tagging*. Springer.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Specia, L. and Motta, E. (2007). Integrating folksonomies with the semantic web. In *The semantic web: research and applications*, pages 624–639. Springer.

Theng, Y. L. (1997). *Addressing the" lost in hyperspace" problem in hypertext.* PhD thesis, Middlesex University.

Vander Wal, T. (2007). Folksonomy. *online posting, Feb*, 7.

Vasudevan, V. and Palmer, M. (1999). On web annotations: Promises and pitfalls of current web infrastructure. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 9–pp. IEEE.

W3C (web). Web annotation working group. https://www.w3.org/annotation. Accessed: 2016-05-02.

Wiles, A. (1995). Modular elliptic curves and fermat's last theorem. *Annals of mathematics*, 141(3):443–551.

Yanbe, Y., Jatowt, A., Nakamura, S., and Tanaka, K. (2007). Can social bookmarking enhance search in the web? In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 107–116. ACM.

Appendix A

SPARQL

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?politician
WHERE {
    ?politician a dbo:OfficeHolder .
    ?person dbo:birthDate ?birth .
    FILTER (?birth > "1960-01-01"^^xsd:date && ?birth < "
       1970-01-01"^^xsd:date) .
}
```

Listing A.1: SPARQL query that returns all politicians born between 1960 and 1970

Appendix B

System Usability Scale Questionnaire

Please read the following sentences and tick the box which best represents your level of agreement with the sentence.

1. I think that I would like to use this system frequently.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

2. I found the system unnecessarily complex.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

3. I thought the system was easy to use.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

4. I think that I would need the support of a technical person to be able to use this system.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

5. I found the various functions in this system were well integrated.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

6. I thought there was too much inconsistency in this system.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

7. I would imagine that most people would learn to use this system very quickly.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

8. I found the system very cumbersome to use.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

9. I felt very confident using the system.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

10. I needed to learn a lot of things before I could get going with this system.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

Appendix C

Computer Proficiency Questionnaire

Have you completed any formal education in Computer Science or a similar subject?

☐ Yes ☐ No

If yes, what was it?

_____

How much time would you spend using a computer in the average week?

○ None at all

○ 1-2 hours per week

○ 3-4 hours per week

○ 1-2 hours per day

○ 3-4 hours per day

○ Greater than 4 hours per day

Please read the following sentences and tick the box which best represents your level of agreement with the sentence.

1. When I have difficulties using a computer I know I can handle them.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

2. I have never felt myself able to learn how to use computers.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

3. I am what people might call a computer person.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

4. I enjoy trying new things on a computer.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

5. It takes me much longer to understand how to use computers than the average person.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐

6. I find using computers confusing.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
☐	☐	☐	☐	☐