Chapter 1: Introduction

1.1 Overview

Over the past few decades, the course of heritage conservation in Australia and around the world has been changing. As information technology becomes increasingly prevalent in all walks of life, it has also become an essential part of the long-term preservation and communication of the wealth of cultural, historic and Indigenous materials collected and maintained by heritage organisations such as trusts, libraries, archives and museums (Lynch, 2002; Lusenet, 2007; Mudgea, Ashleyb & Schroer, 2007).

The use of information technologies in heritage organisations provides many benefits, such as convenient management of information regarding heritage collections using record keeping software, fast and effective communication between researchers through email, and communicating news and events to stakeholders and the general population via public websites and social media.

Information technology also allows for new kinds of heritage materials to be collected. For example, it allows heritage artefacts to be digitised through digital photography and scanning, and for natively digital information such as electronic communications and multimedia to be collected and retained for posterity.

Whether heritage collections are "born digital" or are transcribed into digital form, they possess significant value. The United Nations Educational, Scientific and Cultural Organisation (UNESCO) has identified these digital heritage materials as having "lasting value and significance" and assert that they "constitute a heritage that should be protected and preserved for current and future generations… a global issue relevant to all countries and communities" (UNESCO, 2003).

In Australia, the National Trusts of Australia, representing each state in the Australian Council of National Trusts, are the peak bodies for heritage conservation. Their organisational vision is “a nation celebrating and conserving its cultural, Indigenous and natural heritage for present and future generations”, achieved through “advocacy, research and promotion” (National Trust, n.d.). We pursue this research focus by identifying, applying and developing computer science techniques that contribute to the next generation of tools available for heritage research and conservation, with a focus on local communities as both the experts and subjects of heritage projects.

1.1.1 Community Memory

Over time, a community experiences many notable events. These include celebrations and disasters, changing economic or political conditions, the actions of people of influence, and reactions to circumstances of a national or global scale. The communities in question can be small, such as a group of family and friends at a shared point in their lives, or of much larger scale, such as a town, city or nation.

The way in which these events are recorded or remembered affects community memory. Community memory (Schuler, 1998; Kubicek & Wagner, 2002) is the ability of a group to recall the details of its shared history. While we are all continuously experiencing a period of historic interest as members of various communities, naturally, many details of the community’s shared experience will be forgotten and never recorded, as it is difficult to predict what will someday be considered significant.

Events that are successfully identified as significant have been recorded for thousands of years using various mediums that will outlast the individuals that recorded or participated in them. Some mediums have been available for millennia, such as oral or written communication. New mediums continually emerge as technology advances and becomes widespread, such as video recording and photography, with video and photographic collections being maintained for decades by individuals, families and archival organisations around the world (Lynch, 2005).

Our capacity to record the present moment has never been greater, giving particular consideration to the wide prevalence of technologies like smartphones, digital cameras and the Internet. Indeed, in some cases the rate at which information is captured has exceeded our ability to manually catalogue and document it! (Lynch, 2005). Fortunately, digital media opens the possibility of using computational techniques that allow materials to be transformed, analysed and explored to an extent that isn't possible with traditional analogue media. This raises the question of whether it may in fact be desirable to digitise specific heritage materials to gain these benefits, particularly if it can augment the process of building community memory.

1.1.2 Digitisation of Heritage Material

Much of the new material being created, which might one day be recognised for having historic or cultural heritage value, is natively digital. For older heritage materials, selected collections of traditional media including photographs and slides are being digitally transcribed through digital photography and scanning techniques (Lynch, 2002; Mudgea, Ashleyb & Schroer, 2007; Carmel, Zwerdling & Yogev, 2012). Even challenging source material such as physical artefacts can have a virtual representation created through 3D photography and modelling software, allowing them to be digitised as well (Paquet, El-Hakim, Beraldin & Peters, 2001).

The materials created by this process have led to the emergence of the new research area of Digital Heritage. Digital heritage is the collection and preservation of cultural, historic, environmental and Indigenous heritage materials using digital media. This form of preservation provides a representation of the original material that is easy to computationally analyse and share, and augments the value of the original heritage materials by increasing their accessibility to researchers and the general public.

These digital representations can be stored locally on hard drives, flash drives or optical media, with physical transportation of the storage medium being the means of sharing the heritage materials with others, particularly for large quantities of data. Alternatively, digital representations can be made available on a network using shared drives or cloud storage. This allows a high-capacity network connection such as fibre optic broadband to be used as the means of conveniently transferring files to interested parties. This significantly reduces the need to transport the original materials from archives to different physical locations, with a commensurate reduction in the cost and risk of moving them.

By allowing heritage materials to be recorded, communicated and analysed by an audience far beyond what is feasibly possible with physical artefacts, digital heritage creates new opportunities for developing and utilising specialised information systems that can be employed in research or personal interest projects. Increasing the audience that these collections can be shared with also builds interest in these collections, and promotes heritage research projects that are affiliated with the source material (Lynch, 2005; Mudgea, Ashleyb & Schroer, 2007).

By providing historians and the general public with an unprecedented level of access to heritage collections, new information may be captured that is otherwise difficult to obtain. Historians can provide domain-specific knowledge for their area of expertise, and the general public can contribute their knowledge and perceptions through the process of crowdsourcing, where information is acquired by enlisting the assistance of a large group of interested participants. This type of information can form the basis of specialised software created to support digital heritage.

1.2 Digital Heritage Software

Digital heritage collections create demand for emerging computing technologies that can augment heritage research through the development of information systems that offer unprecedented capabilities and enhancements that are not possible with traditional analogue media.

For instance, wide-scale collaboration such as crowdsourcing becomes feasible when a virtually unlimited number of copies of a heritage resource can be created and shared electronically, contrasted with needing to share a single physical instance of a resource. Additionally, if records about the collection of heritage materials are kept in a digital format, a simple text search through those records can find relevant items in a fraction of the time that it would take to search through equivalent paper-based records. These are simple examples and are already widely used, but they demonstrate the conveniences afforded by digitally-formatted materials.

Numerous digital heritage researchers have expressed the desirability of software-supported investigation of heritage collections over the past decade (Webb & Canberra National Library of Australia, 2003; Mudgea, Ashleyb & Schroer, 2007; Lawless, Agosti, Conlan & Clough, 2013; Ardissono, Kuflik & Petrelli, 2012; Oomen & Aroyo, 2011). These researchers have identified the desirability of a number of key aims, including:

- The long-term preservation of information within heritage collections;
- Improved retrieval in large heritage collections using computational techniques to identify relevant materials;
- Increasing the relevance and accessibility of heritage materials for modern audiences through ease of sharing and open access;
- Community-driven approaches that encourage users to participate in the community memory building lifecycle.

1.2.1 Existing Research

A number of digital heritage research projects have sought to address these aims. Major works in this area include a massive cultural heritage digitisation project, several digital heritage services aiming to make large-scale cultural heritage data accessible to both viewers and experts, two cultural heritage conference series that discuss the future direction of digital heritage, and a benchmarking evaluation challenge that compares digital heritage systems.

Europeana

Europeana is an access point to over two million digital heritage materials, including books, paintings, music and archival records that have been provided by over 200 cultural heritage organisations in Europe. Open access has been provided to these records via a simple web interface which allows the materials and any available metadata to be accessed by experts, researchers, students and the general public (Isaac & Haslhofer, 2013).

This open web access allows novel research approaches to be applied to the collection in an effort to improve the linkage between items in the collection and the information available about them on the wider web, and also to use these external web resources as a means of supplementing the metadata currently available for the collection (Haslhofer, Momeni, Gay & Simon, 2010).

CULTURA

CULTURA (Hampson *et al.*, 2012a, Hampson *et al.*, 2012b) is an ambitious group of research projects that focus on natural language processing, personalised information retrieval and information presentation within Europeana. It provides content-aware adaptivity that responds to key individuals, events and dates identified among collections and uses the relationships detected between them as a way to present personalised, dynamic storylines between the entities (Refer to Figure 1.1 for their website).

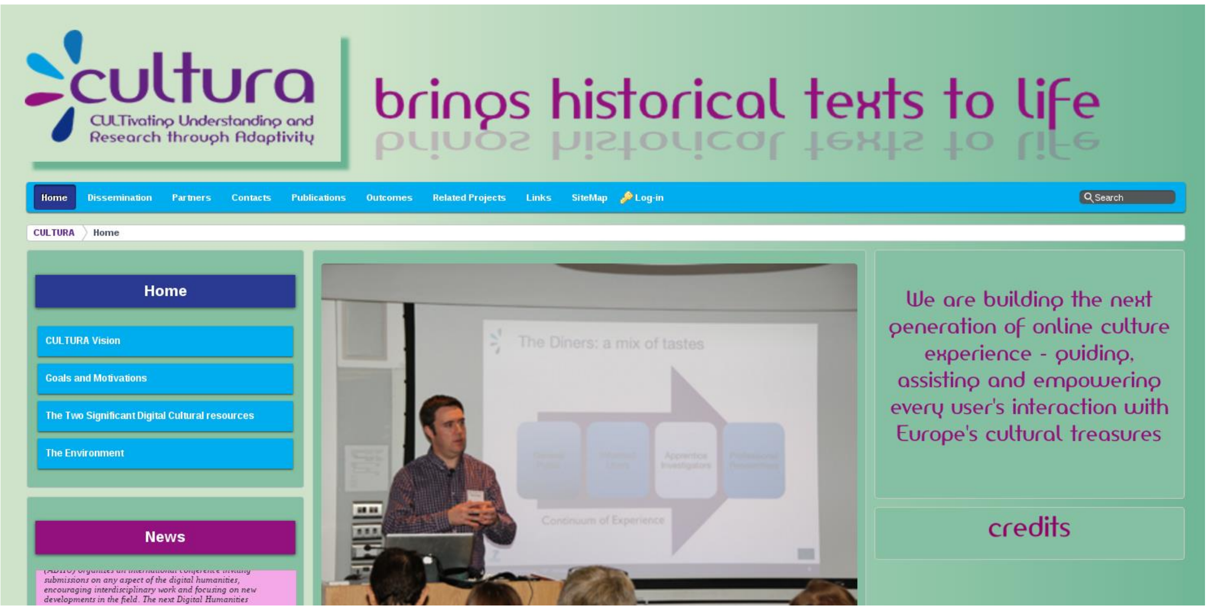


Figure 1.1 - Cultura Website (CULTURA, n.d.)

Source: http://www.cultura-strep.eu/

PATHS

PATHS (Agirre *et al*., 2013) is a novel method for exploring digital heritage collections. PATHS is targeted towards digital library audiences, where its primary goal is the personalised dissemination of heritage collections through a novel navigation strategy that guides viewers along thematic pathways in a collection. These pathways can be created by experts or other users, and assist with exploring the Europeana heritage collection (Refer to Figure 1.2 for their website).

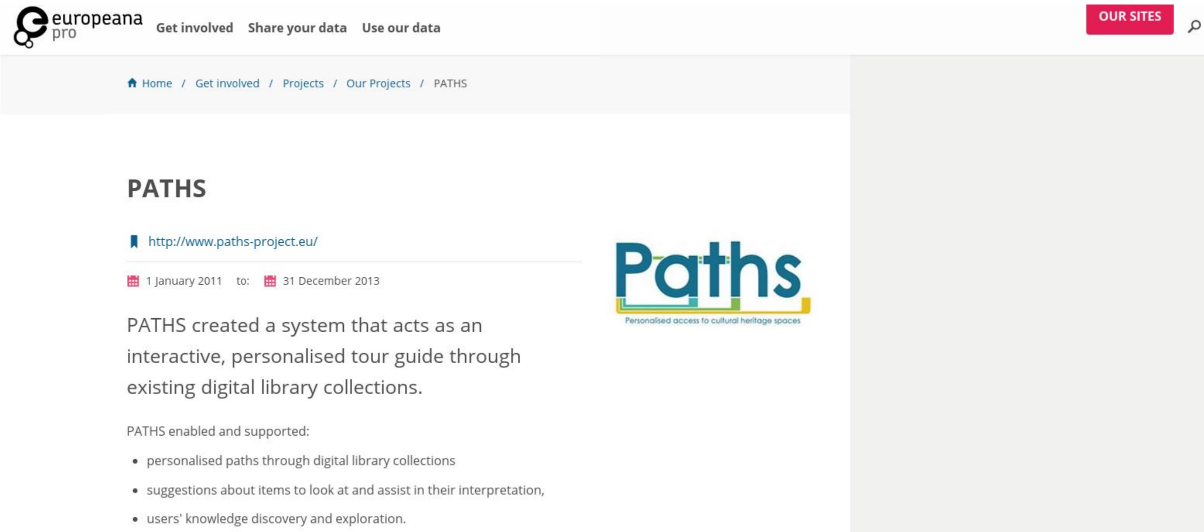


Figure 1.2 - Paths Europeana Website (Europeana, n.d.)

Source: http://pro.europeana.eu/project/paths

Sculpteur

Digital libraries benefit from a diverse collection of multimedia resources, including novel data formats such as 3D models that virtually represent physical artefacts. Sculpteur (Goodall *et al*., 2004) is an example of a utility that users can use to search for 3D heritage models and navigate between them. Modelling heritage artefacts (see Figure 1.3) goes part way to solving the dilemma of how a fragile, finite physical resource can be shared safely with a wide audience, and is becoming increasingly popular as scanning and networking capabilities improve around the world. Specific parts of the 3D model could then be annotated to specify its cultural or historical significance, allowing models to be grouped with those they are related to (Yu & Hunter, 2013).

Figure 1.3 - Example 3D Models in Sculpteur (SCULPTEUR, 2005)

Source: http://www.sculpteurweb.org/html/3Dmodels.html

eCHASE

eCHASE (Sinclair *et al*., 2005) highlights the commercial benefits to effective heritage information retrieval, and seeks to identify how large and valuable heritage collections can be monetised through collaboration with commercial partners as a means of achieving financial self-sustainability in those collections. It allows commercial entities in fields such as education, e-commerce and tourism to search collections for images, film or audio resources that can be licenced and used in products such as advertising, DVDs, books and interactive software.

PATCH

PATCH is a workshop series that gathers a diverse group of researchers and experts together to discuss questions relating to the open access and personalisation of digital heritage collections (Ardissono *et al.*, 2015)**.** It focuses on issues of mobile, context-aware and personalised dissemination of heritage resources, the capture of new knowledge from collective intelligence, and extending the contexts of use of digital heritage materials outside of the traditional contexts like museums and galleries.

ENRICH 2013

The Exploration, Navigation and Retrieval of Information in Cultural Heritage workshop (ENRICH 2013) was a forum where researchers discussed the application of information retrieval to digital heritage, identifying opportunities for collaboration and the steps that needed to be taken to advance the agenda of research in this area **(**Lawless, Agosti, Clough & Conlan, 2013). Critical points identified included the need for context-aware retrieval, which responds to entities and relationships in collections, personalised IR, which responds to the intent of the user, and community-aware retrieval, which incorporates a community's interest and experience in the retrieval process.

CHiC

Cultural Heritage in CLEF (CHiC) was a lab run in 2013 to address issues of multilingual information retrieval in cultural heritage, including tasks for multilingual information retrieval and enrichment in 13 different languages (Petras *et al.*, 2013). This lab brought the friendly competition between research teams in evaluations that the CLEF events foster to the application area of digital heritage, allowing multiple approaches to be compared against one another in a series of uniformly administered and well-designed evaluation exercises.

1.2.2 Research Opportunities

Existing research places emphasis on personalised information retrieval and accessibility of heritage collections from a digital library perspective (Borgman, 1999; Historic Environment Scotland, n.d.). This focuses on the opportunities that become available in the later parts of the information lifecycle of heritage collections, when sufficient information is present to describe a heritage collection, and the main challenge has become making this information available and useful for users.

This research project aims to advance the state of the field by addressing critical needs that are not adequately addressed in previous work, specifically with regard to information collection and analysis in the early phases of the information lifecycle of heritage collections. This project seeks to capture and cultivate the information that eventually becomes the core data for other digital heritage systems, particularly when this information is scarce and difficult to obtain, such as when the information needs of viewers are not well defined, or when initial information about the materials in a collection is largely absent.

When searching or exploring these collections, the information needs of the viewers are likely to be diverse. While on one occasion, a viewer might be searching for a particular person, on a separate occasion they might be looking for an event, object or location. This makes it difficult to predict what information the user will require, and means that collecting a range of information is more desirable than focusing on one specific aspect of the heritage collections.

With the large volume of data in the collections and the general absence of semantically-rich annotations, there is scarce information that can be used for traditional methods of computationally searching these collections for relevant entries. For example, if these collections are not thoroughly documented, common methods of retrieving information that rely on searching through text will not be applicable, yet a search technique for exploring these collections, which can contain hundreds or thousands of items, is highly desirable.

The challenge, then, is to determine a method by which these diverse and semantically complex collections of heritage data can effectively be explored and searched, and that encourages the viewer to contribute new information and participate in community memory building, making it easier for the next viewer to access the information contained in the collection.

The main contributions in this project are:

- Identifying and analysing the challenges encountered by multimedia information technology when specifically applied to digital heritage.
- Providing strategies and techniques to account for these challenges at critical points throughout the digital heritage information lifecycle.
- Designing an application and algorithm that allows the rapid collection and exploration of heritage annotations, fostering information collection through community memory building.
- Providing a facility whereby annotations can be exported, which provides the necessary priming data for other information retrieval technologies used in digital heritage research.

These contributions are targeted at a key point in the evolution of digital heritage, seeking to occupy the point between when heritage collections are being digitised, and when sufficient information about these collections has been captured to be able to apply automated computational techniques such as machine learning to analyse heritage materials on a vast scale.

1.3 Following Chapters

When considering the approach that needs to be taken, we need to address two key questions. How can we scale human analysis so that it can effectively process a large body of information without compromising the quality or consistency of indexing? Also, how can we index this information in such a manner that a computer is able to organise it based on the semantic meaning uncovered by the users? These questions are explored in the remainder of the thesis, which is organised as follows:

In Chapter 2, we present findings from an extensive literature review on the topics of information retrieval and multi-agent computing systems. This provides background information on the current state of research and the trends that have emerged in recent years. Having identified the application domain and research aims in the introduction, the literature review focuses on the computer science techniques that provide the means to make these aims possible.

In Chapter 3, we outline the software architecture used in the thesis and the factors which influenced design considerations. This presents a complete information system using technologies from information retrieval systems, multi-agent computing, multimedia information retrieval and digital heritage that allows knowledge from participants to be collected and cultivated to annotate heritage collections.

In Chapter 4, we explore the similarity measure algorithm used in the project. This is the core component of the heritage system developed in this research project and the method by which the heritage system supports its users in their annotation work.

In Chapter 5, we evaluate the use of the system under controlled conditions and consider the factors which can impact on a user's interactions with the system. This allows us to determine whether the suggestion mechanism described in the previous chapter has been adequately accepted and utilised by the users.

In Chapter 6, we assess a case study that the project was applied to. This provides a real-world example of the uses of the heritage system, and examines the external factors that limit or encourage its application.

In Chapter 7, we conclude with finishing remarks and prospective future work. We summarise the results of the research project and identify key areas where it could be further improved for real-world application and use.

Chapter 2: Literature Review

2.1 Introduction

The technique developed as part of this research project needs to address two competing needs. On one hand, we need to collect and cultivate semantically-rich information from experts and volunteers as part of a community memory building process. On the other hand, we need this information to be accessible to computer algorithms so that the annotation process can be supported, reducing the effort required by participants to produce high quality, consistent data. Compromising on either aspect runs the risk of compromising the system as a whole.

These contrasting requirements parallel the challenge encountered by search technologies. Search engines must have some means of determining which pieces of information should be returned in response to a user's query. Traditionally, this is achieved through either a mostly automated, computational approach that analyses the information for keywords, or through a mostly manual approach where a social network of users provides information cues in the form of tagging, popularity scores, relevance feedback, or some other social indicator (Lew, Sebe, Djeraba & Jain, 2006).

In the automated methods, the search engine is able to process vast quantities of information by using web crawlers to automatically visit websites and index information based on keywords. With the sizeable amount of computing power available to these systems, indexing a colossal amount of information is an achievable feat. However, the information that is processed is not truly understood by the web crawler, and the information's significance cannot be captured by this process.

Contrasting with this are manual approaches that rely primarily on users to perform this indexing. Websites are visited by users who are able to interpret and understand the significance of what is being viewed and then build web directories or other indexing structures to catalogue this semantic information. This yields high quality results, but the amount of information that can be indexed and the consistency of the approach is limited by the users.

A combined approach makes it possible to balance the strengths of both manual and automatic techniques by employing users to process the semantic information that is difficult to analyse computationally, and automating organisational tasks in order to handle large quantities of information (Li, Snoek & Worring, 2009; Hare *et al*., 2006; Wu, Yang, Yu & Hua, 2009; Zhao & Grosky, 2002). It is essentially following the adage of using the most effective tool for each job. While this provides a partial solution, the limitations of this approach are the combined unresolved issues experienced by the individual approaches. The performance and consistency of users will still be a limiting factor to processing information, and the computer's ability to "understand" information will limit how effectively it can be organised.

We need to examine a number of fields in order to see how these problems can be reconciled in our technique, which by necessity will follow a combined approach. One focus of research is targeted towards understanding how groups of users can interact in a massively parallel operation, which is required to meet the information processing requirements of the application. The areas of Multiagent Computing and Human Computation are explored for this reason. The other focus of research is targeted towards developing a set of computational methods that can be used to meet the data organisation and analysis requirements of the application. The areas of Information Systems and Content-Based Information Retrieval are examined to find these techniques.

2.2 Information Systems

The quantity of information accessible by the public has massively increased in recent decades. With the widespread availability of computers and multimedia creation devices such as digital cameras and smartphones, as well as the pervasiveness of vast networks such as the Internet by which to communicate, our lives have changed to incorporate information as a central concern in our work and recreation.

In order to utilise the wealth of information available to us, we require effective information systems that allow us to contribute to and explore this information. Continual research and development in Information Systems focuses on the core issues of how to analyse new kinds of information, how to index the information for search and exploration, and how to retrieve and display information in an intuitive and meaningful way for users (Lops, De Gemmis & Semeraro, 2011).

This section first addresses the common components of information systems before investigating four kinds of widely-used systems. As the technique developed in this research project is in itself a type of information system, our investigation of common concerns and what has proven to be effective in the past is likely to be advantageous.

2.2.1 Analysing Information

Information can take many forms, including text, images, audio, video and binary data. Each form of information has its own unique challenges that must be overcome before information systems can effectively interpret it, which is essential for filtering information to fit the information needs of a user.

Traditional methods of accessing information draw from the textbook metaphor, where the user finds an appropriate information source and then reads it in a linear manner. In addition to finding the information that was originally sought after, the user is likely to encounter a large amount of unsought information, which may not be a useful return on the time invested in reading it. Using the linear approach on a large, information-rich source is likely to lead to “information overload” (Hiltz & Turoff, 1985), where the capacity of the user to process information to find what they are seeking has been exceeded and has started to cause them mental distress.

An alternative approach to linear accessing of information is to seek specific pieces of information only when they are contextually required. This leverages computer systems as a substitute, transactive memory. So long as the information is available and readily accessible, it can be "known" to the user whenever they need it. Social commentary has referred to this as the "Google Effect", where people are less reliant on their own faculty to recall information and instead remember where to find it when needed (Sparrow, Liu & Wegner, 2011). Improved methods of categorising and searching information are needed for this to be performed effectively.

Information discovery in transactive memories can be assisted by software by promoting exploration through contextual links using the semantic content of information. Providing the option for users to navigate to related content enables the user to confine their reading process to the information that is most likely to be relevant to them. For example, rather than reading an article in its entirety, users may use in-page navigation to move immediately to the subsection of the article that interests them specifically, allowing them to retrieve the information they are seeking more effectively.

2.2.2 Information Indexing

The essential aim of information systems is to provide a means by which a user's search query can be related back to relevant pieces of information, referred to as *documents*, which can then be returned to the user. This can be achieved through the process of indexing, organisation and retrieval using metadata (Singhal, 2001).

Metadata is information about information. Simple examples could be the photographer, time and location of an image stored in an information system. This kind of metadata has a very close relationship to the information being stored. Pieces of information can also have relationships with other information, such as a number of images making up a photo album. If an information system can capture these relationships, then the rich, implicit information in these relationships forms a facet by which it can be explored.

Indexing information is the process by which potential search terms are associated with a document as metadata. In many cases, this is achieved through using keywords, as the user will usually formulate their search query in text. In an ideal situation, every keyword relevant to a document would be ranked by relevance and attached to the document, but in reality, the indexing of documents is often incomplete and relevance can be ambiguous. For example, searching for “travel suitcases” or “travel luggage” may yield different results if documents aren’t labelled with both terms. Additionally, is the presence of the term “travel” actually necessary in results?

Information organisation allows relationships between documents to be exploited to provide improvements over the results that would be returned by the documents alone. There are numerous methods by which documents can be processed and related to one another based on similarity, such as similarity measures in feature extraction, or proximity in semantic networks, with similar documents being able to share missing indexing information (Zhao, Wu & Ngo, 2010). For instance, missing or superfluous keywords can be extrapolated by examining similar videos (Zhao, Wu & Ngo, 2010).

Organisation also allows documents deemed to be more important to have a higher ranking than less important documents, which enables the information system to promote documents most likely to be relevant to a user. Consider the PageRank algorithm used in Google Search, which is able to effectively rank web pages by importance through assigning a level of authority to pages based on the number and authority of inbound and outgoing links in that page (Page, Brin, Motwani & Winograd, 1999).

2.2.3 Information Retrieval

Information retrieval is the process by which a user's query is translated into indexing terms which can be presented to the information system to find matching documents. This can involve keyword extraction, keyword expansion to include synonymous search terms (Chirita, Firan & Nejdl, 2007), and the collection of additional search data such as a user's recent interactions with the information system (Borges & Levene, 2000; Garg & Weber, 2008) for relevance feedback. The documents retrieved by the system are then displayed in a ranked, paginated list for the user to interact with.

The goal of information retrieval systems is to develop an effective architecture for collecting, processing, storing, analysing and retrieving high-level semantically meaningful metadata about documents of various types that adapts over time as additional information is added and as users interact with the system. These systems seek to effectively utilise the advantages offered by metadata, promoting the idea that a user should be able to leverage both the content and semantics of data to find, discover and share information.

"Finding" is a directed search where the user's query needs to be analysed and matched to the metadata of digital objects they are seeking to locate. "Discovering" is an undirected search where a user browses the semantic links between digital objects, such as the members of a class of information or the relationship between classes. These concepts are focused on being able to precisely access the user's desired information with the greatest possible ease.

"Sharing" is the process by which additional value is added to a digital object from the interaction of users. On the web, this has almost become synonymous with the deliberate action of re-posting material on a social network, but it can also include processes such as collecting anonymous user metrics as a measure of the interestingness of the material, indicating that it should become more visible to discovery or search for other users, such as views on a popular video (Szabo & Huberman, 2010).

Numerous prominent approaches are used for information retrieval, including Search Engines, Web Directories and Knowledge Management Systems, as well as Informal Information Systems which allow information to be discovered and shared by indirect methods.

2.2.4 Search Engines

Information resides in many locations and takes many forms on vast networks such as the Internet. Viewed from an end user's perspective, this makes it extremely difficult to find and access relevant information to assist with the activity they are performing. In order to feasibly use this information, the user relies on an intermediate piece of software to interpret their requests and return links to the information they seek. The most common kind of software used for this is a search engine (Brin & Page, 2012).

Search engines are perhaps the most visible application of information systems, and utilise computational agents called web crawlers to explore new websites and index them based on their content and the links which point to the website. Due to ranking algorithms favouring established documents which are well-linked to by other documents, a search for a topic is more likely to retrieve historical documents than recent documents (Doerr & Iorizzo, 2008). This helps to establish an element of stability in search rankings, although ideally, the recency of a document should not affect an overall measure of its quality and relevance unless recency is a factor of the search.

By almost exclusively using automated methods, search engines are able to scale very well to vast quantities of information and large numbers of search queries, though the indexing they perform is semantically shallow when compared to what people can conceptualise and communicate.

2.2.5 Web Directories

While search engines provide an effective means to deal with vast quantities of information through the application of computational methods, manual methods can still yield good results in information systems. One only needs to consider Wikipedia to see that crowdsourced manual work can yield a very rich and comprehensive resource (Kittur, Chi, Pendleton, Suh & Mytkowicz, 2007).

One such manual method is the creation of web directories by a group of users. In a web directory, a large number of websites are categorised and organised by users based on their understanding of the website and the relationship the website has with other web resources. In traditional web directories, this is done through a formal ontology of websites within fixed categories or classes (Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004). Websites are visited by users who are able to interpret and understand the significance of what is being viewed and then build web directories to catalogue this information. This yields high quality results, but the amount of information that can be indexed and the consistency of the approach is severely limited by the scalability and variability of the manual organisation techniques employed.

One of the problems with the web directory method of organising information is that it relies on categorising information, employing the library cataloguing metaphor that assigns each piece of content to a single subject area, when what is more desirable is a system which associates information based on its semantic relevance (Doerr & Iorizzo, 2008). Categories don’t provide sufficient granularity to answer specific questions, which is an objective of information systems. Categories can also hide semantically similar documents when these documents fall under separate categories. The cataloguing process additionally requires a high amount of cognitive effort to decide on the most appropriate category when several are possibilities, or potentially even more cognitive effort when multiple categories are permitted and the list of categories is very large (Doerr & Iorizzo, 2008).

A technique commonly associated with being part of the Social Web is social tagging. Tagging can create a folksonomy (a portmanteau of folk and taxonomy) of tags which relate back to content by establishing a link between the tag and the web content (Lu, Hu & Park, 2011). This is an informal means of categorising information through crowdsourcing that produces a result that bears a strong similarity to web directories. New techniques can also consider the identity of the user doing the tagging, creating a tripartite system where the link between user, tag and resource can be explored as additional metadata. This allows the tendencies of individual users to be captured and used as a factor in determining the relevance and quality of the tag (Lu, Hu & Park, 2011).

2.2.6 Knowledge Management Systems

Tagging is primarily an unstructured process, with information being voluntarily contributed by users, but structured methods for capturing knowledge also exist. Knowledge management is a field which attempts to capture the intellectual assets of an organisation in the form of translating implicit knowledge to explicit knowledge and facilitating explicit knowledge sharing among members of the organisation. The motivation for this is to provide a means of building a pool of knowledge from which employees can learn in order to improve efficiency and effectiveness, and that would also prevent this knowledge from being lost through turnover of staff (Tuzhilin, 2011; Alavi & Leidner, 2001).

Initially a popular area of research, interest in the field of knowledge management has slowed due to the gap between the objectives of knowledge management and the reality of what was achieved by implementations of knowledge management systems. A criticism of knowledge management is that it relies on the assumption that implicit knowledge can be effectively codified into explicit knowledge, when in reality some implicit knowledge is very difficult to capture (Tuzhilin, 2011).

Knowledge management is also commonly technocentric, with knowledge being shared through an interface that lacks the subtle benefits of establishing a context in which skills are learned, identifying an individual who can be responsible for maintaining the information's correctness through timely updating, and identifying individuals as aides for additional learning. Employees can be hesitant to share knowledge when their specialist skills are their leverage for promotion and job security (Davison, Ou & Martinsons, 2013).

2.2.7 Informal Information Sharing

While explicit and formalised knowledge management has been well studied, less research has been conducted in informal knowledge sharing systems, existing in the modern era in the form of blogs, forums and wikis, and in the past as "conversations at the water cooler". Davidson, Ou & Martinsons (2013) show that informal knowledge sharing can be preferable in numerous contexts to formal knowledge management systems. In the informal systems, the problem's context and the learner's and teacher's identity are established prior to knowledge sharing, which is particularly desirable as certain people are identified as experts by this exchange, encouraging further knowledge sharing when the learner wishes to ask additional questions.

Systems focused on user-generated content like chatrooms and forums may essentially be considered informal knowledge sharing systems. Methods of traversing this content with the goal of knowledge acquisition could address several of the issues with traditional knowledge management. Specifically, people are encouraged to share across the organisational boundaries which would normally restrict where knowledge can be accessed or contributed, and the wider audience of these sites increases the number of experts who are able to formulate and express their knowledge in response to a learner's questions.

Knowledge management could be reimagined, utilising a software stack that incorporates content management as an underlying technology for capturing and sharing knowledge. This layered approach to knowledge management could incorporate informal knowledge sharing processes as a means of generating new insights and lines of enquiry in the field (Tuzhilin, 2011).

2.2.8 Discussion

The idea of transactive memory is particularly appealing for digital heritage applications. It allows experts and volunteers to share their own partial understanding of a particular subject in order to build a more comprehensive picture than they would otherwise individually hold. This also includes the potential for capturing conflicting or contentious accounts, which are fascinating to examine when a range of different perspectives exist on a matter. The use of contextual links is also desirable, as it would allow users to navigate between semantically-related subjects, such as from a particular event to the important persons involved in the event.

In order to create a system that supports these features, we examined several information retrieval systems to identify the most desirable characteristics of each of them to use for a specialised information retrieval system in digital heritage. Critically, the quality and richness of information captured by web directories, knowledge management systems and informal information sharing is ideal, but a more scalable approach such as those used in search engines is still needed to connect disparate pieces of data using some kind of content-based indexing, rather than relying on users to manually make those connections.

2.3 Content-Based Information Retrieval

Multimedia tagging has traditionally been a field dominated by two approaches. One approach is to use automated and semi-automated computational techniques which utilise content analysis to index information. The other approach is to use large numbers of users to manually categorise information (Lew, Sebe, Djeraba & Jain, 2006). Computational techniques are desirable because they allow vast quantities of information to be indexed, while human techniques are desirable due to the high semantic quality of indexing that they can produce. Approaches which use both computers and humans, described as being "multimodal", have been shown to be effective in generating indexes which score well on both measures (Li, Snoek & Worring, 2009; Hare *et al*., 2006; Wu, Yang, Yu & Hua, 2009; Zhao & Grosky, 2002).

Content analysis is advantageous for information retrieval purposes as it allows information to be retrieved based on its content rather than its annotations, which is useful since content is implicitly present in information while annotations may be sparse. In addition, content is implicitly a good description of the information in any situation where the user wishes to retrieve the information based on a recalled property of the content in directed search, browsing or surfing (Datta, Joshi, Li & Wang, 2008).

In a directed searching, users attempt to find a piece of information that the user knows exists. One of the best ways to accomplish this is to allow the user to search based on features of the content they remember, such as recalling a quotation used in a text document, or an image with certain visual properties, such as “Images of a Tiger”. Users can also browse a category of related content to find items which match what they are seeking, which are likely to be categorised together based on metadata or content, such as browsing from “Striped Animal” to “Tiger”. Users surfing without a distinct goal can do this by traversing links between pieces of information, which may be in the form of relationships detected in the content, such as to “Wildlife” to “Predators” to “Tiger”.

One of the first steps to enabling this kind of content-based search is to extract and analyse the key features of the subject that distinguish it from other related pieces of multimedia. This is the process investigated in this subsection, along with the challenges that come with associating features with the high-level concepts people use for search queries.

2.3.1 Feature Vectors

The goal of content analysis is to represent the important details of text, audio, images or video in a concise form which can then be used for comparison. This concise representation of content is called a *feature vector*, with feature vectors being produced using feature extraction algorithms. Similarity measures can be used to evaluate if two feature vectors are similar, which implies that the original material is also similar. This is important when establishing the relationships between content based on similarity, or when retrieving information using feature vectors derived from the user's query (Liu, Zhang, Lu & Ma, 2007).

Multiple feature vectors can be produced using different algorithms for the same content, with similarity measures which account for multiple vectors being generally more flexible and reliable than those that rely entirely on one (Bennett & Lanning, 2007). The primary disadvantage with having a large number of feature vectors is high dimensionality, where to process each additional feature vector requires exponentially more computing time. Dimensionality reduction techniques attempt to remove redundancy in the feature vectors, creating a concise number of high-significance vectors for comparison purposes (Furnas *et al*., 1988).

2.3.2 Feature Extraction Algorithms

A feature extraction algorithm is any algorithm that can process an input signal and produce a concise representation of the salient details in the signal, which forms a feature vector. Feature extraction algorithms target any aspect of data of the input signal and can also target metadata about the signal (Liu, Zhang, Lu & Ma, 2007). For example, an image feature extraction algorithm might target certain shapes in images, while a metadata extraction algorithm could target locations or photographers. For a feature extraction algorithm to be effective, it is necessary for the algorithm to accurately capture the essential details of the signal and also to capture these details in a practical amount of computational processing time.

A common method for indexing text is to produce an inverted list (Ferragina, González, Navarro & Venturini, 2009). In these lists, numerous keywords in a text are extracted and their position is recorded to allow for fast searching of that text. A comprehensive list would be larger than the original document, so selecting the right words as keywords to form the index is a challenge of this approach. Another challenge is that these indexes do not allow for substrings to be easily located. An alternative method is a full compressed text index which captures the entire document and losslessly compresses it, with any queries being converted to the compressed form and then used to search the compressed document. This mitigates the aforementioned problems with inverted lists, but is more difficult to construct from an implementation point of view (Ferragina, González, Navarro & Venturini, 2009).

These methods perform indexing within a text that allows for the location of a substring or keyword to be detected, but the locations of keywords in text is not necessarily required for a text feature vector, merely the number of occurrences within the text. If a document contains the same keywords as a user's search query, it's likely to be relevant regardless of whether the keywords appear towards the start or towards the end of the document. This means that relatively simple indexing techniques can be adapted to allow for two documents to be compared for similarity.

This can be achieved by building a histogram of the frequency of occurrence of keywords in a text and then comparing histograms to determine the similarity between two sections of text. This is called the bag-of-words approach (Harris, 1954). There are methods of enhancing the results obtained through this method, including filtering the keywords based on stop words to reduce the size of the feature vector (Wilbur & Sirotkin, 1992), normalising the histogram to allow a short passage to be compared to a longer passage (Singhal, Buckley & Mitra, 1996), using bi-gram two word groupings as keywords rather than single words (Collins, 1996), grouping synonymous keywords into synsets (Miller, 1995), and using converged word hashes in place of words for indexing (Forman & Kirshenbaum, 2008).

2.3.3 Similarity Measures

A similarity measure compares two feature vectors to find their similarity. A feature extraction algorithm performed on two similar input signals should produce feature vectors with a small distance between them, while two different input signals should produce feature vectors with a greater distance (Datta, Joshi, Li & Wang, 2008).

A common method of finding this value is to calculate the distance between feature vectors, such as through Euclidean distance, which applies Pythagoras's theorem to find the distance between the vectors. A common method for then categorising these feature vectors into a number of classes is through clustering and classification, such as by using the k-means algorithm, which groups feature vectors by iteratively evaluating the centroid location of the class and classifying all feature vectors which are close to the centroid (Datta, Joshi, Li & Wang, 2008).

This can be applied to the bag of words model by creating a vector where each entry in the vector is the frequency of which a specific word in a text dictionary occurs, such as "[1, 0, 2, 4]" or "[0, 1, 1, 3]" for a four-word dictionary. The Euclidean distance can then be calculated between them. In this example, the distance is 2. These feature vectors can be improved through weights being applied to the frequencies, such as the popular tf-idf weighting which compares the number of times a keyword appears in the text (thus, likely to be a more significant term) balanced against the number of times the word appears in corpus (thus, a more common term implying less significance) (Salton & Buckley, 1988).

This similarity measure focuses on lexical matching, but there is also the issue of semantic matching. For example, “I have a dog” and “I own an animal” are semantically similar but lexically dissimilar. One of the popular methods for determining the semantic similarity between texts is to use Latent Semantic Analysis (LSA), sometimes referred to as Latent Semantic Indexing (LSI) (Deerwester, Dumais, Landauer, Furnas & Harshman, 1990). LSA identifies patterns in the relationships between keywords and semantic concepts using the assumption that words in similar contexts will have similar meanings. A vector is produced for each word where the values of the vector are the number of times the word appears in each paragraph in a large body of text. The number of values are condensed using singular value decomposition, after which the word vectors can be compared with one another to determine if they are similar. If their cosine is close to 1, they are similar, and if their cosine is close to 0 they are dissimilar (Mihalcea, Corley & Strapparava, 2006).

2.3.4 The Semantic Gap

Computers are able to analyse data such as images, text, sound and video using feature extraction algorithms to produce low-level features as metadata about these objects. A person, on the other hand, creates high-level metadata that includes information about the semantics and meaning of the object. This high-level metadata is valuable because it is typically how a person would find, discover and share information.

The greatest issue in feature extraction research is relating the feature vectors that have been extracted with the semantics present in the input signal. The process of translating low-level features into high-level, semantically meaningful metadata is a difficult one and is referred to as bridging the semantic gap (Hare *et al*., 2006; Zhao & Grosky, 2002). The semantic gap has been the subject of research in fields such as object recognition, facial recognition and machine vision for decades (Celma & Serra, 2008; Wang, Zhang & Zhang, 2008). By considering information systems from a usage-driven perspective, rather than a data-driven perspective, we can develop systems that more closely emulate the expectations of users.

Bridging the semantic gap would allow computers to understand multimedia as well as humans can. This would be an extremely important development in a wide variety of projects such as robotic vision (Martens, Lambert & Van de Walle, 2010), artificial intelligence natural language processing, natural interfaces and information retrieval. This understanding, in addition to the rapid rate at which computers can process and store information, would lead to a period of rapid technological development.

One of two approaches has predominantly been followed in relation to bridging the semantic gap. Either operations are performed exclusively on the semantic level in order to avoid needing to bridge the gap, such as comparing a user's search with metadata stored on a document, or low-level features are painstakingly linked with high-level semantics through techniques such as supervised training to recognise certain objects (Bloehdorn *et al.*, 2005).

While bridging the gap has only been achieved in a limited sense, the popularity of web information retrieval services such as Google and Yahoo! show that there are still strong commercial applications for this early stage of technology and a great possibility of further commercial opportunities coming from breakthroughs in this area (Datta, Joshi, Li & Wang, 2008).

There are multiple levels at which the semantic gap needs to be addressed to equal the capabilities of a human (Liu, Zhang, Lu & Ma, 2007; Hare, Lewis, Enser & Sandom, 2006). Firstly, it requires the ability to detect the composition of the input signal. Secondly, it requires the ability to determine the significance of the composition. Thirdly, it requires the ability to infer implicit context from the explicit details in the image. An example of this is to have an image of a political protest about the Vietnam War. First we need to determine that there are a number of people in the image, then we need to determine that together, the people form a political rally, and thirdly we need to use our general knowledge and the visual cues in the image to infer that it is related to the Vietnam War. While this is a relatively simple task for a human, it is much more challenging for a computer.

Each level is progressively more difficult, so while systems exist that attempt to address each of these levels, the results of systems that attempt to function at the highest level are much weaker than those operating at the lowest level. Because of this, systems that are operated manually by human users are still employed, though the performance of these systems can often be improved with the application of a semi-automatic system that puts human users in a supervising or supported role where the software attempts to improve the efficiency of the operator through a number of enhancement and automation techniques (Beale, 2007).

An area of artificial intelligence most closely related to text-based feature extraction is the field of Natural Language Processing (NLP). The goal of NLP is to be able to extract knowledge from natural text automatically. This was a popular field in the 70's and 80's, but research in the field slowed as it became apparent this would be a long-term research goal. In the last decade, a few notable examples of NLP were proposed, including a simple technique called TextRunner (Banko, Cafarella, Soderland, Broadhead & Etzioni, 2007), which detects nouns in text and uses the text between them to relate the two nouns. For example, "Bob is friends with Sally" would link the two nouns Bob and Sally with the relationship "is friends with". Another approach by Paşca, Lin, Bigham, Lifchits & Jain (2006) uses a set of seed facts and patterns for knowledge acquisition that allows highly-rated knowledge extracted from text to be added to the set of seeds for a successive pass.

While these kind of approaches do not propose methods by which the software can gain a deeper *understanding* of the text, it does allow a useful knowledge base to be developed which other approaches can utilise, such as through relational clustering, which can produce semantic networks where strongly related names or relationships are grouped together (Kok & Domingos, 2008).

Semantic Networks are a way of representing pieces of information in a structure that highlights the relationships between them (Sowa, 2006). Each piece of information is represented as a node of a graph, and the edges in the graph are the relationships. Network analysis techniques can be applied to semantic networks to allow multiple pieces of information to be compared based on the sets of their relationships. An intuitive example would be stating that if two entities have a similar set of relationships with other entities in the network, then the two entities are likely to be similar in nature.

One of the best known semantic networks is WordNet, which collects over 150,000 words in the English language into semantically-related groups called synsets (Miller, 1995). WordNet has been used as a component in numerous information retrieval applications such as sentiment analysis (Baccianella, Esuli & Sebastiani, 2010) and detecting the relatedness of concepts (Pedersen, Patwardhan & Michelizzi, 2004), due to the usefulness of the information it has captured about synonymous and related terms in the English language.

2.3.5 The Semantic Web

One of the largest and most exciting applications of semantic network technologies is the Semantic Web (Berners-Lee, Hendler & Lassila, 2001). While computers are very capable of following the electronic links between documents and resources on the Internet, they have only a very limited understanding of what those documents and resources represent.

When a user requests that certain information be retrieved, a search engine uses the information it has collected to return what it evaluates as being the best result. These search engines are reliant on users to then examine the results and select those that are relevant from among those that are not useful. For the purposes of information retrieval, the system is also then reliant on the user analysing the content of the results to determine the answer they sought.

While this is reasonably effective for finding information on a specific search topic, asking a complex query such as “where was Queen Elizabeth II’s grandfather born?” will yield much weaker results as the search engine has no means of understanding the query and evaluating the answer, or even determining whether websites satisfy the request unless it contains that exact text. The user will then need to perform additional work to find the specific information they seek.

The Semantic Web attempts to address this issue by including semantic information in websites that categorises the material that appears on a page. For instance, indicating one link in a web page is to a supporting reference, while another link may be to an example. This allows a semantic search engine to use computational techniques to interpret websites and return better results to a user query. Semantic technologies are also being employed in a limited sense in consumer technologies, such as the Google Knowledge Graph, showing an increased acceptance of the technology from the perspectives of both developers and users (Singhal, 2012).

There are a number of challenges facing the Semantic web, including the vastness, vagueness and inconsistency of the information available on the Internet that must be incorporated into the semantic web. For the semantic web to become a reality, techniques that are comparable to a high-level understanding of content, must be developed. Some of the promising areas for achieving this are in the fields of artificial intelligence, including fuzzy logic, evolutionary computing and neural networks (Chen, Wu & Cudré-Mauroux, 2012).

2.3.6 The Social-Semantic Web

Just as the addition of social technologies to the existing web led to the use of the term Social Web or Web 2.0, the (Social-)Semantic Web or Web 3.0 is a new term coined for the speculative fusion of social web technologies with semantic networks. It re-envisions the semantic web as being built with complementary social technologies that use contributions of humans as a useful source of information that has been made available in a machine-interpretable format, enabling seemingly intelligent and intuitive electronic support systems (Berners-Lee, Hendler & Lassila, 2001).

Social networks and technologies produce a vast amount of user-generated content and provides an interface between users and the information by which it can be accessed. The application of semantic technologies on this user-generated content could potentially yield a system for creating, describing and retrieving this wealth of information. However, such a system faces a number of challenges arising from the often informal nature of the information, such as from off-topic noise and identifying entities that pose a problem when attempting to create formal semantic ontologies (Sheth, & Nagarajan, 2009).

One area in which social-semantic technologies are being developed is in semantic wikis. A wiki provides an environment in which users can interact via pages developed with relatively simple markup in an online text editor, supported by features such as automatic version tracking and search. A semantic wiki aims to capture the relationship between these pages through additional metadata to be able to automatically generate overview pages and categories, which currently requires intensive manual work to keep updated (Schaffert, Bry, Baumeister & Kiesel, 2008). Semantic wikis could offer additional information to the techniques currently being employed on wikis that generate semantic networks of concept relatedness, such as the work conducted by Wojtinnek, Pulman & Völker (2012).

Another area in which social-semantic technology is being developed is in social-semantic bookmarking. Social bookmarking allows many users to collaborate on annotating links to webpages with a number of tags that provide a means of accessing these links organised within categories. The relationship between individual tags is typically ignored, leading to a flat structure that has a poor representation of the semantics of the tags. An approach by Wei & Ram (2012) proposes to use network analysis techniques on the tag space to categorise tags into a hierarchy based on their intended purposes, which can provide additional semantic information about the links they reference.

It is also possible to apply semantic techniques to other social technologies such as collaborative filtering, which uses the properties of an object to find other objects which the user might also be interested in examining (Breese, Heckerman & Kadie, 1998). Determining the relationship between these properties could allow these systems to intelligently guide the user through a product flow to arrive at an optimal product recommendation. For example, collaborative filtering may be good at logically extrapolating a set of preferences the user is likely to express from the smaller set of preferences they have already expressed, but semantic techniques could be used to make new preference suggestions which users have not historically expressed.

2.3.7 Discussion

Considering the difficulty of bridging the semantic gap, an approach which relies on human users to interpret the heritage materials they are viewing, then making computational judgements on whether two items are similar based on the user's perceptions is a feasible way of developing a system which generates additional value to the sum of the contributions of its users. If the user's perceptions and knowledge are recorded as text, text feature extraction algorithms can be used as the interchange format between human users and computational algorithms.

This approach operates using two semantic levels. Occupying the lower data space are the heritage collections, stored and represented as digital images. Experts and volunteers can examine these images and record their observations as annotations, which forms the higher semantic space. Content-based information retrieval algorithms operate directly on this semantic space (rather than on the data space, as is commonly done) to identify when patterns or descriptions of annotations imply that the images they link to are related.

The software can use these detected similarities to intelligently support the user, such as by helping them with the task of contributing additional annotations, or assisting with finding information relevant to the user's interests. Identifying the scope of support opportunities and selecting the means by which this support can be provided forms the next part of our literature review.

2.4 Multiagent Computing

An agent can be thought of as an independent entity that is able to perceive their environment, perform some logic on what is perceived, and interact with their environment based on this logic. The capabilities of an agent can range from trivial to very complex, and the definition can be broadly applied, including to software programs and human users, individuals or a group. Agents can be used to assist with modelling such as in simulations, or as part of complex systems that exhibit interesting and desirable properties (van der Hoek & Wooldridge, 2008).

Agents can communicate through their environment or through some form of communication protocol. When two or more agents are present and interact together in a meaningful way, they become part of a multiagent system. These systems can span across a countless number of agents and incorporate a variety of agent types, for example a large group of software agents assisting a few human agents to perform a cooperative task.

Multiagent systems are a flexible means of working with distributed computations. As the agents are independent and self-organising, these systems are also somewhat able to recover from faults resulting from the removal of some or many of these agents. Agents may differ in what they know and how adeptly they can perform certain tasks, so one of the principal aspects of designing a multiagent system is considering what agents should fulfil which roles, and how they should communicate to accomplish the overarching goals of the system (Odell, Parunak, Brueckner & Sauter, 2003).

2.4.1 Agent Roles

Some problems which are trivial for humans to solve are difficult to solve computationally. For example, consider attempting to create a system that can evaluate concepts like beauty or creativity. Likewise, some computational tasks are labour-intensive for a human to solve, but can be performed very effectively by a computer. This leads to the idea of having software and human agents occupy complementary roles that ensure effective operation of a system and minimise the complexity of the problem being addressed by allowing communication between the agents to collaboratively solve complex tasks (Shahaf & Amir, 2007).

While the term “role” is vague, in multiagent systems it is likely to connotate a set of responsibilities and a set of privileges which enable the agent to perform these responsibilities. In many cases this will additionally imply a communication protocol which will allow other agents to interact with the agent filling the role if the role is to provide a service. Roles are not necessarily fixed, and an agent may move between roles if the agent and system support this behaviour, such as a human user moving their attention from one task to another. Roles may also exist for a limited duration of time, or until the role is no longer necessary (Zhu & Zhou, 2008).

While software can be created to fill many different kinds of roles and every copy of a software agent will perform that role with equal effectiveness, due to variance in individuals, human agents may be predisposed towards filling only a limited number of roles which appeal to their personalities, and may have different levels of proficiency at filling these roles. Nevertheless, some simple roles are able to be filled by almost any human agent through the natural faculties that are possessed by people, such as perception, comprehension and observation.

2.4.2 Contributions of Human Agents

People can contribute to human computation passively or actively. Systems can analyse both types of contributions concurrently as a means of acquiring additional information from users (Borges & Levene, 2000; Breese, Heckerman & Kadie, 1998).

A user can contribute passively simply through usage of a system. The usage patterns can then be mined for information (Borges & Levene, 2000; Garg & Weber, 2008). This includes metrics such as the time spent using a particular service or examining a specific item of content, selections such as the items the user most commonly searches for or navigates to, and interactions such as expanding posts, watching videos or liking content. These types of contributions are generally of particular value to popularity or ranking systems.

A user can make an active contribution if they are explicitly assigned a task or select a task to perform. The result of this task makes a vital contribution to the system, hence why this is sometimes referred to as *human in the loop* processing (Schirner, Erdogmus, Chowdhury & Padir, 2013). This includes producing or expanding content, such as listing observations and related knowledge about an object, organising and improving content, such as marking objects as being semantically similar, or acting as a decision maker.

Decision making is particularly important as it allows a system to pose a question to a user or group of users for non-trivial problems rather than requiring a complex decision making algorithm to be developed. These decisions can also be used as training material for an artificial intelligence which allows the system to adapt what it has observed from humans to produce its own responses to frequent problems. It also provides the capacity for supervision when the system is about to make a decision which would require approval from the users, such as deleting or merging material.

2.4.3 Contributions of Software Agents

Software agents provide a means of recording and analysing information that has been supplied by other sources. They are ideal for storing, processing and transferring large amounts of information, as well as delivering well-presented and timely information and services to users wherever users can access the application (Nwana, 1996).

Tasks which are repetitive, time-consuming and tedious for human users are appropriate areas to apply software agents. This frees human agents for creative and informed work. However, software agents can also be used to guide users to high-value work areas through simple communication methods shown in the user interface. For example, software agents can analyse who a user is friends with in a social network and determine who else is likely to have met the user. The interface can then show a notification which asks the user to respond to whether or not they know the suggested people.

A combined approach makes it possible to balance the strengths of both manual and automated techniques by employing users to process the semantic information that is difficult to process computationally, and use software to automate other tasks in order to handle large quantities of information.

The main bottleneck of this approach, similar to a manual approach, is the human processing power that can be brought to whatever application is being worked on. Multi-agent systems and social computing are a means of directly addressing this challenge, with the proposed solution being to have as many screens with as many users behind them as is appropriate for the size of the project, and having a portion of the computing power at each user's workstation being directed towards supporting and enhancing the contributions of the individual users to achieve maximum effectiveness.

The secondary bottleneck of the approach, similar to the automated approach, is the ability of the system to detect the approximate significance and semantics of information to assist users. This can be addressed by designing the underlying data structure as a semantic network, where the relationships between pieces of information are one of the central concerns of the data structure, instead of being ignored as superfluous data. Software analysis can be performed on the relationships between the nodes of the network to provide an indication of whether objects may contain additional semantics beyond what has already been detected. Users can then be consulted to confirm these results if they are accurate.

2.4.4 Hierarchies of Agents

In addition to considering whether a role is needed and should be filled by a human or software agent, we must also consider how the roles are structured with relation to one another. A simple organisational structure is the idea of tasks being carried out in parallel by peers. Another simple structure is to consider the idea of an agent being "one level above", such as a manager, or "one level below" in the hierarchy, such as a worker. A formal method for defining roles and structuring them with reuse as an objective is explored by Tran & Low (2008). In this method, roles are identified and appropriate agent types are selected to fill these roles. Next, the internal logic of agents is designed, as is their communication protocol. Finally, agents are given a means of interacting with their environment.

Manager agents may be responsible for a single worker agent, or they may be responsible for many workers. The number of worker agents should be determined by either the number of roles that need to be filled for the manager to perform their work appropriately or an unrestricted number when worker peers can perform a task in parallel. An example system would be a human manager with several software workers who assist with repetitive tasks, information retrieval and communication. Alternatively, we could consider a number of human workers performing tasks being coordinated by a software manager who assigns a new task to each worker when they complete their previous task.

2.4.5 Multiagent Architectures

Distributed systems are systems where users or software applications do not need to be co-located with one another in order to function, and operate over networks such as the Internet. Multiagent systems are an effective way to create and manage a distributed system, and a number of different methods exist by which agents in a distributed system can be organised in an architecture of clients, servers or peers over a network (Babaoglu, Meling & Montresor, 2002).

In the simplest non-distributed case, we have a single hardware system which encompasses all agents. For example, this could be several software agents existing within a single computer with a single human operator. As all communication and interaction is performed locally, much of the organisational complexity of other architectures can be avoided. The number of agents in this kind of system is strictly limited, however, as only one human agent or group of human agents can access the computer interface, and the capabilities of the hardware of the computer limit the number of software agents that can exist.

In a peer-to-peer architecture, any number of computers can be linked together, supporting as many human and software agents as required. In these systems, it is very easy to organise groups of agents into peers, where each of the peers is a group of software and human agents which exist on a local computer, but it is difficult to organise the peers into a distinct hierarchy across the network of computers.

In a client-server architecture, like a peer-to-peer architecture, it is easy to accommodate a large number of human and software agents across many computers. Unlike the peer-to-peer system, it is much easier to establish a hierarchy if the server is used as the central authority which dictates which clients fill which roles. There is a limit, however, on the number of human and software agents that can exist on the server, much like in a single computer system.

Finally, a hybrid system, while the most complex organisation of agents across distributed architecture, allows for the greatest amount of flexibility. A simple hybrid system could be composed of one group of peers acting as a server and one or more groups of peers acting as clients. This is an especially relevant architecture when considering the possibilities offered by cloud computing, where physical machines can be abstracted into multiple virtual machines with no persistent local storage and a shared remote database, whose configuration and management is supplied by a Platform as a Service provider (Chang, Abu-Amara & Sanford, 2010). This significantly reduces the complexity the software developer needs to address, which would typically be the main drawback of a hybrid system.

2.4.6 Discussion

When considering how agents should be structured to support a digital heritage application, we need to observe several important characteristics about the experts and volunteers who are using the system.

Experts and volunteers will provide diverse perspectives and knowledge on a given subject matter, and their personal interests are likely to be the guiding force that directs them to the materials they will contribute to. This strongly supports the adoption of a parallel structure, where human agents are free to access whatever information they wish, rather than being assigned a fixed area to work on by a software agent acting as a manager.

However, passive software support that indicates where valuable work can be contributed is also useful as a prompt for contributions that can be made. Passive support for linking work conducted by human agents is also a desirable feature, as human agents may find it difficult to search through the breadth of information contributed to the system by other users, particularly as the amount of information that has been captured grows larger. For example, identifying related heritage resources and allowing users to easily navigate between them casts the work conducted by users as part of a greater whole, rather than work conducted in isolation.

A system which observes these concerns therefore places the users as equal and freely-operating peers, who independently utilise and contribute to the information contained within the system, while software agents acting as workers and aides support continuous indirect communication between human users by prompting targeted contributions that promote the creation of a well-connected network of information, with a wide range of benefits to the outcomes of the system.

2.5 Human Computing

The Social Web, sometimes referred to as Web 2.0, is used to describe an ongoing paradigm shift in the way the World Wide Web is being used. Our usage is moving from static websites published by individuals and organisations towards dynamic websites that feature user-generated content, online services, collaboration and social interaction (O'reilly, 2007). This change is gradual and influenced by popular websites which showcase these new technologies and inspire us to reconsider how we interact with the Internet. One only needs to look at examples such as Google, Facebook, YouTube and Wikipedia to see the potential of what this new paradigm of web usage offers.

Our changing behaviour is matched by changing roles of how we produce and consume information. Fischer (2009) argues that we have become “prosumers” of information, both producers and consumers, and that traditional models of learning that presents schooling as knowledge acquisition and work as knowledge application are outdated, and that in today’s society we are constantly learning and applying our knowledge. This process results in a culture of participation built upon interaction.

The social web offers a wealth of information to researchers in the form of social interaction. From simple social indicators such as “liking” a piece of content through to extended informative contributions made by people from a wide range of backgrounds, the Social Web has many opportunities for information collection and processing which can provide a broad range of opportunities.

2.5.1 Social Computing

Social computing is a term used to describe the intersection of social interaction and computing. On a simple level, it addresses how computers can support social behaviour between individuals through technology, such as using VoIP, email, instant messaging, blogs and social networks. On a more complex level, it addresses how social behaviour can be analysed so that groups of people can carry out useful computations through combining their knowledge and expertise.

Technology that facilitates communication allows users to perform structured and unstructured collaborative tasks as though they were communicating in person. Its advantages are that it allows users to better manage their communication (responding when they are available, rather than interrupting their current task), and can introduce flexibility to communication (when the individuals are separated by physical distance or communicating at different times) (Johansen, 1988).

Collaborative technologies have been developed for work and learning: Computer-Supported Collaborative Work (CSCW) and Computer-Supported Collaborative Learning (CSCL) are the two fields associated with these developments. The goal of these systems is enabling effective communication through digital channels. Technology can provide additional benefits over physical communication, such as features which enhance workflows or the collaborative learning process, such as collaborative mind mapping or brainstorming software that allows users to contribute highly visual information, or a collaborative design space where multiple participants can sketch ideas collaboratively (Sangiorgi, Beuvens & Vanderdonckt, 2012).

Users are also able to collaborate on tasks that specifically require the contribution of a group of people to be successful. Examples include collective sources of information such as Wikipedia, collective sources of personal news and social interaction such as Facebook, or collective sources of links to online content such as Reddit. In these situations, the contributions of users with diverse skills and knowledge act to produce a result that exceeds the quality of what could be achieved individually, encouraging the development of systems that specifically seek to enable social computing applications.

2.5.2 Social Computing Systems

Social computing systems can be implemented with a variety of simple technologies such as social networks, wikis, blogs and content tagging, or more complex technologies such as social bookmarking and recommender systems (Breese, Heckerman & Kadie, 1998).

Systems which enable users to contribute information and opinions to users with related interests is a basic form of social computing where communication is facilitated but not aggregated or analysed by computational methods. This communication can be achieved through blogs, wikis or through a social network. Users are responsible for establishing relationships with other users, selecting information sources and the type of information they share. Users who post better or more interesting information receive more attention compared to other users, essentially creating a relevance filtering system (Leavitt & Clark, 2014).

A more complex form of social computing in terms of analysis performed are social bookmarking services and recommendation systems (Breese, Heckerman & Kadie, 1998). In social bookmarking systems, the user recommends certain resources via bookmarks, and these bookmarks are aggregated across the userbase, with the most popular bookmarks having the highest ranking. Recommender systems analyse what products are liked or purchased by a user and aggregates these statistics across the userbase to determine when interest in one item is likely to correlate to interest in a related item, particularly when the user for which the recommendation is being made has similar preferences to an established group of other users, such as providing movie recommendations on Netflix (Bennett & Lanning, 2007).

Social computing has an enormous range of applications. Examples include areas such as online communities and entertainment, business, as well as the public sector.

Online communities benefit from social computing through the rich economic, social and political systems it can create. Supply and demand created by and manipulated by users can create a virtual economy, while interaction with other users both on a social and political level is more emotionally satisfying than interacting with scripted events and non-player characters in games (Wang, Carley, Zeng & Mao, 2007). Social interactions can also be analysed to help software or robotic agents to perform more realistic and effective communication for use in the entertainment industry, such as for non-playable characters in video games. The responses people provide to interactions with software can help train the software to give better responses themselves, such as in chatbots (Wang, Carley, Zeng & Mao, 2007).

Social computing can be employed in business through recommender systems, which collect user opinions and aggregate them to rank items for promotion, as well as filtering systems which suggest similar products to what a user has shopped for in the past. It can also be used as an indicator for supply and demand for certain types of products (Wang, Carley, Zeng & Mao, 2007). A form of social computation used for information organisation is content tagging, where short text descriptions are associated with content such as images, video, sound or links. These tags allow the user to browse content which shares the same or similar tags as a means of performing directed exploration of the content. Users can also provide alternatives to keyword tags when a different type of content tagging is more appropriate, such as narratives or descriptions (Marshall, 2009).

In the public sector, social computing can analyse social trends for applications in political discourse, counter-terrorism, healthcare and public transport. Trends can be extrapolated to produce forecasting systems which help to gauge the costs and benefits of a decision (Wang, Carley, Zeng & Mao, 2007). An additional area in which social computing is developing is the field of social and community intelligence. With the ubiquity of GPS-enabled mobile devices and stationary computing devices in urban areas, it is possible to take social computing into physical reality with applications in public transport optimisation, traffic monitoring, and disease outbreak control (Zhang, Guo & Yu, 2011). In this way, the manner in which the population interacts with their environment can be mined for information to support decision making and optimisation.

2.5.3 Human-Based Computation

The reason why human-based computation is so desirable in social computing systems is that people are typically able to form a much stronger understanding of the material they are processing than computers. This includes better understanding of the significance of the material and of the semantic relationships the material has with other pieces of material (Zhao & Grosky, 2002). In the case of information systems, the result of human processing will typically be much more relevant to the requirements and interests of people than the results of computational processing would be. Consider a well-written discussion article and a table of summarised data, compared to a much larger table of raw data. Both are meaningful, but the former is more accessible to viewers than the latter.

Another advantage is that human computation allows for the parallel production of content. In cases where the user is engaged in an activity with scarce additional information in the form of annotations or content, it is possible to request the user to produce this creative material in addition to their processing tasks.

The primary drawback of human computation is that users will typically perform tasks significantly more slowly than a computer. While a computer can perform thousands of simple tasks per minute, a human user may only be able to complete a handful of tasks. This can be partially addressed by effectively managing the human agent's time, maximising the benefit obtained by the system by directing users to tasks that the computer would find too complex or ambiguous. There is also the option of engaging multiple human agents through social computing to increase the work that can be performed per minute.

In certain situations, due perhaps to a lack of knowledge or experience, or perhaps for other reasons, people may be uncertain about the decisions they make, sometimes even making mistakes when performing the task assigned to them for human computation. By treating the result of every human computation as though it was complete and correct, unexpected complications can arise in the operation of a system (Lughofer *et al.*, 2009). There are numerous means of mitigating this issue. These include human-based correction mechanisms such as aggregating answers over a group of people (much like the "Ask the Audience" option in game shows), by having decisions checked by other people, or computational correction mechanisms such as developing flexible systems that tolerate uncertainty and incorrect decisions through techniques such as fuzzy logic (Zadeh, 1965). The field of Multiagent Computing is particularly invested in these techniques.

In certain cases, uncertainty can be explicitly or implicitly reported as an additional form of information about a decision making process. High uncertainty can signify a difficult question or one with multiple conflicting perspectives. Systems utilising this can be more robust in how they handle human computation through developing a better understanding of the question and responding accordingly (Cedilnik & Rheingans, 2000).

Finding common areas where human computation is effective allows it to be readily employed when those situations are identified. Quinn & Bederson (2011) identify seven effective application areas for human computation: collection, search, parallel tasks, iterative improvement, active learning, statistical processing and genetic algorithms.

Collection, search and parallel tasks use a group of users to perform activities in parallel, where individual efforts cumulatively achieve a significant result. In collection, this involves building and improving a knowledge base where users are the authors and editors of the knowledge stored within it. In search, this involves selecting appropriate results from a database when the search criteria are too complex to be expressed to a search engine, such as finding visually appealing artwork or selecting potential candidates for a job. In parallel tasks, potentially unrelated tasks are performed as needed to ensure continued operation of the system. This could include a control-room scenario where people deal with issues flagged by the system.

Iterative improvement is structured in a chain or hierarchy of users to improve upon the answer provided by the previous users. In a chain, the user is only given one item of input and their contribution comes from performing additional work on the input. In a hierarchy, the users may be presented with several items of input and are required to select and filter the best input to pass along, not necessarily performing additional work on the selected item.

Active learning, statistical processing and genetic algorithms are structured as aggregations of the user group’s answers. Active learning finds the most informative example for an unlabelled training set and aggregates responses made by users who have been asked to provide an annotation for it. Statistical processing asks the same question to a group of users with their average or most popular decision being accepted as the “correct” decision. Genetic algorithms can use an aggregated evaluation of an entity’s fitness in situations where human users have a more accurate perception of the fitness than a computer, such as when an entity is measured by how visually or auditorily pleasing it is to human observers.

Essentially, many processes can be outsourced to human agents and it is in fact desirable to do so when humans outperform other types of agent. Not all human agents will be motivated for the same reasons or want to perform the same processes though; specifically selecting contribution types for the human agents, or allowing for self-selection, can lead to improved user productivity and satisfaction (Kosorukoff, 2001).

2.5.4 Human-Centric Computation

The user is at the centre of the information seeking task, but the systems used to assist with solving the task aren’t always structured this way. The major shortcomings in these systems are that supporting the user is not a primary focus, that the user's individual requirements and context are not considered, and that the processing the user performs during decision making is not retained and used to enhance the system to improve the experience of other users who are solving the same or similar questions (Shtykh & Jin, 2011).

Human Centric Computing (HCC) takes an alternative approach to human-computer interaction compared with other computing paradigms. Instead of emphasising a computer program as the means by which a particular task must be completed, and having users act as operators of the program, responsible for translating a business or organisational objective into tasks that can be performed by it, HCC envisions the business or organisational objective as an objective being performed by users, and seeks to design software that supports or augments a user's capacity to achieve it (Waibel *et al.*, 2010).

By focusing on the user's context and intentions, HCC guides software developers into building better interfaces to programs and ensures that these programs serve useful and unobtrusive roles in the user's work. This is achieved through interface design where the program attempts to assist or guide a user with the task that the user has selected, rather than passively waiting for the user to correctly interact with the software.

HCC is an important complement to the concept of Human Computation, as human processing time is significantly more valuable than computational processing time due to its relative scarcity and high effectiveness when dealing with complex decisions or concepts. Therefore, it is desirable for a system implementing a human-in-the-loop architecture to proactively support a user by maximising the impact of each of the user's contributions and minimising the amount of time spent in navigational and trivial tasks.

While existing text and graphical interfaces are capable of supporting a user, a far-reaching goal of Human Centric Computing is to move towards more natural interfaces. These include interfaces that are capable of interpreting natural speech, gestures, facial expression, the work context of the user and the user's environment. Developing systems which can understand human behaviour is a very challenging ambition, but the reduced cognitive load to interact with these ubiquitous computing systems would allow for virtually effortless communication and the potential to employ any person as an agent for human computing (Pantic, Pentland, Nijholt & Huang, 2007).

2.5.5 Human-Computer Joint Exploration

There are a number of areas in which human-centric computing practices can enhance the process of human computation. Joint exploration is an approach where computational methods are used to assist users in navigating and annotating content. The objective is to use a suggestion mechanism to direct a user's attention to the most useful areas for tagging and reduce the cognitive load on the user by automating or assisting with repetitive tasks. This can be achieved in a number of ways (Wang, Ni, Hua & Chua, 2012).

One method to improve the quality of tagged information and increase the number of tags associated with a particular object is assistive tagging. In this model, tags with a high likelihood of being applied to an object are suggested to the user and the user takes responsibility for curating these suggestions, adding relevant tags and rejecting inappropriate suggestions. As more tags are added to an object, the accuracy of the suggestion mechanism generally improves to the point where there are no likely suggestions left to offer, resulting in an effectively and comprehensively tagged object.

Another method to improve the number of tags applied to objects is to offer users a batch of objects which can all be tagged simultaneously, or allow a user to select a batch prior to tagging. This allows general tags to be applied to many objects, granting a large improvement in productivity.

Tag refinement aims to take user-generated tags and improve their quality, either merging tags which have a high rate of coincidence, inferring that one implies the other, or splitting compound tags into several atomic tags to give finer control to tagging mechanisms.

Lastly, active tagging highlights certain objects for tagging for which tags would be particularly informative. These can include objects which are weakly connected to the main body of knowledge and therefore require additional tags to become more strongly linked, such as newly-added objects, or objects which have formed small networks which could be more strongly connected to the larger network of information by adding bridging concepts.

2.5.6 Discussion

Social computation is a means of addressing the limited human computing resources available to small-scale heritage research projects. It allows like-minded individuals around the world to collaborate on specific collections and share their understanding, rather than working in isolation. This research project seeks to capitalise on the recent trends towards the Social Web and allow analysis of user-generated content to provide a novel means of organising and exploring heritage collections.

The project also seeks to adopt human-centric computing practices by unobtrusively providing suggestions to the user, and automatically updating information displayed through the interface to continually reflect the best suggestions that the software is able to provide. The methods identified in human-computer joint exploration are a strong inspiration for adopting these user-friendly practices.

2.6 Summary

When we began the literature review process, we sought techniques that would allow two seemingly competing needs to be reconciled, allowing participants to contribute semantically-rich information relating to digital heritage collections while also structuring this information in a way that makes it interpretable by a computer algorithm, which reciprocally supports and augments the efforts of participants. By combining research from Information Systems, Content-Based Information Retrieval, Multiagent Computing and Human Computation, this novel goal can be achieved.

Through social computing, large amounts of information can be analysed by breaking the heritage annotation task into parallel subtasks which can be performed by individual human agents. This allows what would otherwise be an overly complex task to perform computationally to be carried out through a multiagent system over the web. This gives us a practicable means by which we can perform human computation on a broad scale and meaningfully interpret heritage materials to produce high-quality annotations.

Content-based information retrieval provides two methods by which software agents may “understand” the data they are processing; by capturing the relationships between items and analysing this metadata, we can make some generalisations about the similarity or dissimilarity of those items based on their annotation sets. Text feature extraction applied to the descriptions of the annotations allows us to form longitudinal links between similar annotations created by different users, which encourages the formation of a well-connected network. By applying computational analysis directly on the semantic space, we avoid the issue of bridging the semantic gap. The result of this is what will hopefully be a highly effective tool for information retrieval in digital heritage applications.

Chapter 3: System Architecture

3.1 Introduction

Earlier in the thesis, we identified that existing heritage information systems tended to focus on presenting existing heritage information to viewers, who are regarded as passive users of the system. This misses out on the opportunities gained by allowing users to participate as active contributors, where users effectively act as both the producers and consumers of heritage information and become a valuable asset to growing and enriching heritage projects.

Our research conducted during the literature review has provided a diverse set of tools to work with when looking to design a proactive, user contribution-driven system. Traditional information retrieval systems can be combined with ideas taken from content-based information retrieval and multiagent systems, including human and social computation, to create a new kind of digital heritage system that emphasises the value that can be captured through the user's participation.

In this chapter, we explore the architecture of the software which was specially built for this research project and that incorporates these ideas. Titled "Semantic Annotation by Group Exploration" (SAGE), this application is comprised of multiple software agents interacting with human agents via a web interface in order to capture, explore and refine information about a digital collection.

SAGE was developed as a series of prototypes, with feedback on each prototype providing the basis for selecting the modifications to be introduced in the successive prototype. In total, three major versions were released prior to the final version of SAGE that was used in testing, with many incremental releases between each version to provide bugfixes and minor enhancements over the course of eighteen months of development.

This approach allowed techniques adapted from existing literature to be steadily implemented in SAGE as the need for different capabilities were uncovered, and was critical to successfully identifying and addressing the unique challenges that were presented when applying these techniques to this thesis' research domain.

SAGE is made available for use by digital heritage researchers and other multimedia information retrieval research projects, for which SAGE can be readily extended to include new algorithms and evaluations for comparisons between approaches. SAGE is highly suited to being applied as a utility for producing and exporting a set of ground-truth annotations for a heritage image dataset, recorded in the simple and accessible CSV format for data interchange, which can be then used as priming material for automated or semi-automated annotation techniques used in future works.

We begin designing this architecture by analysing the types of data that are required to satisfy the information needs of the users of this system. This consolidates the application areas that this software can be used for, and also outlines its expected usage patterns.

Next, we look at the types of agents that must be created to represent the data that SAGE captures and manipulates. We also consider the manner in which these agents should behave when presented with certain external interactions, allowing them to function semi-autonomously and contribute value back to the multiagent ecosystem.

We then investigate how this information can be made accessible to users through a user interface. This is the medium through which the software and human agents will communicate and interact, and is responsible for facilitating all necessary user tasks.

Following this, we consider the ancillary systems used in the application, which provide important services to the software agents and user interface, such as persistent storage and security.

Finally, we address how the system can be extended using different state-of-the-art similarity measures, and how they can be compared against one another using a suite of evaluation tools that help to identify the most suitable similarity measure for a given application area or heritage data format.

3.2 Related Work

Key related works in this section are projects that have implemented and tested novel designs for capturing the relationships between the documents in a generic image collection. This provides the ideal data structure for our work in digital heritage collections, as these relationships provide a crucial point of leverage for enabling the use of software agents to support the work conducted by the users of SAGE.

3.2.1 Digital Objects

Digital objects are a piece of information taking any digital form. This includes multimedia files like text, images, audio files or video files, or specialised formats of data, like databases, spreadsheets, presentations, word processing documents, or data files.

Digital objects are an abstraction of the information value of data that disregards the format that it is expressed in. This allows the information to be conceptualised without consideration of the technical requirements of how it is encoded and recorded, and simply reflects on the fact that the object has information value that can be utilised as part of complex operations.

3.2.2 Object Ontologies

Ontologies provide a formal vocabulary for categorising and describing the relationships between digital objects. Objects are referred to as entities, and the nature of the relationship between entities is referred to as a class. An example could be that the entities London and England are related by the class CapitalOf. Annotations that conform to these ontologies can be added to digital objects and interpreted by machines. Unlike a folksonomy where only the existence of a relationship between entities can be analysed, ontologies allow classes to function as the bridge that actually connects semantic networks to knowledge representation (Kiryakov, Popov, Terziev, Manov & Ognyanoff, 2004).

The W3C recommends several formats for representing semantic information, particularly in the Semantic Web. Adhering to standards allows for different semantic web technologies to operate on the same data. This has the benefit of allowing complementary software to operate on the same data, or competing software to be directly compared to determine the best algorithm to use in the system. These formats are RDF and OWL (Chen, Wu & Cudré-Mauroux, 2012).

Under the RDF (Resource Description Framework), data is represented as a triplicate of the format <subject> <predicate> <object>, with predicates being supplied by the vocabulary definitions in RDFS (Resource Description Framework Schema). RDF statements can be serialised in formats such as XML (W3C, 2001a). An example is shown in Figure 3.1 below.

```xml
<?xml version="1.0"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:a="http://description.org/schema/">
    <rdf:Description>
      <rdf:subject resource="http://www.w3.org/Home/Lassila" />
      <rdf:predicate resource="http://description.org/schema/Creator" />
      <rdf:object>Ora Lassila</rdf:object>
      <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement" />
      <a:attributedTo>Ralph Swick</a:attributedTo>
    </rdf:Description>
  </rdf:RDF>
```

Figure 3.1 - RDF example (W3C, 2001b)

RDF is used in many semantic networks available on the Internet. Its main alternative, OWL (Web Ontology Languages), allows for more detailed semantic statements than RDF. OWL uses a vocabulary of primitive statements to describe a full ontology of the relationship between two objects, such as the relations between classes, cardinality, equality, richer typing of properties, characteristics of properties and enumerated classes (W3C, 2012).

3.2.3 Semantic Tagging

Despite the benefits of using ontologies, describing the significance of objects and the relationships between them in a formal vocabulary is a time-consuming process that can only be performed by those that are familiar with the format. A less controlled, unstructured format reduces the need for consistency and thoroughness, effectively reducing the barrier of entry to new annotators and reducing the effort required to contribute each annotation.

Users can manually annotate objects using short text-based descriptions called tags (Wu, Yang, Yu & Hua, 2009). These are popular in social networking and image sharing websites, with the use of hashtags being popular among these. A hashtag is a tag taking a particular form where a hash symbol is prepended to the tag, with separate hashtags being delimited by spaces.

Tags relate to the content as well as the context of the object. For instance, a picture of a waterfall may have the tag "#waterfall", but might also have tags such as "#familyholiday2013" or "#naturewalk" which provide information that isn't explicitly present in the photo. The list of tags associated with an object can be viewed, providing additional information about the object, or a specific tag can be used to browse objects that have been annotated with the same tag.

Contrasted with ontologies, allowing the free tagging of documents creates what is referred to as a folksonomy. One of the main criticisms of folksonomies is based on the perception that by not constraining the vocabulary, the variance in tags will continue to increase with many tags actually referring to the same concept as would have been identified by an ontology. There is evidence to show that users who are able to see the semantic tags created by other users are likely to mimic them, and that users with similar information needs will create semantically similar tags. This suggests that as a folksonomy is developed, it converges towards its own vocabulary (Fu, Kannampallil, Kang & He, 2010).

Gupta, Li, Yin & Han (2010) present a comprehensive survey on the types of social tagging techniques, as well as their applications for producing recommendations, tag visualisations, and the challenges stemming from the use of tags. This research emphasises the use of folksonomies as a way of supporting the emergent needs of users, rather than ontologies or taxonomies which prioritise issues of language. Tags can be used for a range of documentation purposes, including content, context, organisation, attributes, opinions, purpose, facts, personal tags, and even self-references and enables a wide range of expression of the user's perceptions and perspectives.

There are a range of challenges regarding effective manual tagging. In addition to the labour required to annotate a large collection of content, tagging can be incomplete, can have different levels of precision, and tags can be incorrectly assigned to content. Even the language used to tag content can raise challenges: polysemy allows a word to have multiple meanings, such as "a field of knowledge" or "a field of flowers"; different lexical forms of words can exist that may be intended to refer to the same essential concept, such as “flower” and “flowers”; and synonymy allows for multiple words to refer to the same semantic meaning in certain contexts, such as "spade" and "shovel" (Deerwester, Dumais, Furnas, Landauer & Harshman,1990; Marchetti, Tesconi, Ronzano, Rosella & Minutoli, 2007).

A solution to this problem is to introduce the idea of semantic tagging (Marchetti, Tesconi, Ronzano, Rosella & Minutoli, 2007), which is the approach taken in this thesis. In semantic tagging, a short description is provided of the semantic concept that is intended to be captured, and users conscientiously select and assign these semantic concepts to objects as tags and use the concepts as queries to retrieve information. This attempts to avoid many of the complexities introduced by language, but requires support systems to help users locate the semantic concept they wish to tag or retrieve (Marchetti *et al.*, 2007). Semantic tags differ from normal tags in that “Images of Mountains” or “Mountain Images” are two different tags (lexically speaking), but refer to the same concept, and would therefore be merged into a single semantic tag.

Merging instances of duplicate semantic tags is an important challenge to overcome to take full advantage of the low cognitive effort of free tagging in semantic tagging systems. Rather than needing to spend time in consideration to determine the most appropriate semantic tag to apply, a user should ideally be able to write a semantic tag that expresses the idea they wish to convey, and if their tag duplicates the semantics of an existing tag, an automated method of reconciling (or merging) the tags would take effect. This is one of the key technical requirements explored in this thesis, particularly within Chapter 4.

Manual semantic tagging isn’t often feasible if it requires the users to input a significant amount of information to find or create an appropriate semantic tag, as users may be disincentivised to produce tags due to the high effort required. In automated semantic tagging, techniques such as Explicit Semantic Analysis attempt to create semantic mappings between keywords and concepts using resources such as Wikipedia by analysing how frequently the keyword appears in articles describing a particular concept. This allows a keyword-based tagging system to leverage many benefits from a semantic tagging system without requiring significant additional involvement by users (Egozi, Markovitch, & Gabrilovich, 2011).

3.2.4 Multimedia Semantic Networks

Multimedia semantic networks are networks formed by a collection of digital objects that have been connected together by common semantic tags. One can view the tags that have been associated with an object, and use the tags to navigate to semantically-related objects.

Each multimedia file is treated as a digital object to allow information represented in diverse forms (e.g. website, text, idea, video, audio, image, etc.) to be stored in the same semantic network. The actual data that comprises the multimedia file is not necessarily stored or accessed in the network; the semantic network is an abstract structure that references the already existing data, rather than storing copies of each file. This allows individuals or organisations (such as heritage institutions) to maintain control over hosting their material, while simultaneously enabling numerous multimedia semantic networks to reference these materials without causing interference to each other or the owner of the hosted materials.

The objects in these semantic networks are tagged with concepts that are short descriptions of a particular quality that the object is perceived to possess. This can include a vast range of metadata types such as artist, photographer, location, time created, historical significance, cultural significance, people present, topic, objects, event, and so forth. When a property is shared by multiple objects, indirect links are established between those objects in the semantic network.

Representing information in semantic networks exposes patterns in relationships which are difficult to detect otherwise, and analysing semantic networks enables a software algorithm to interpret and predict these patterns as a way to support the user's work. The software's objective is to determine the most important or productive changes to be made, and bring these changes to the user's attention so that the user can make a decision. This is designed to maximise the effectiveness of the human processing time that is consumed in performing the task. Multiple users can work in parallel, allowing contributions to be made in parallel on whatever scale is required.

3.3 Challenges

A variety of challenges face the application of multimedia information retrieval techniques to the field of digital heritage. Many of these relate to the nature of the heritage photographic data being analysed, but some are also introduced by the way in which this complex semantic knowledge is recorded and utilised.

This thesis seeks to identify and analyse these challenges, then propose viable solutions which allow these challenges to be circumnavigated or circumvented. These enhancements will be incorporated into a semantic similarity measure (the Semantically Annotated Graph Analysis or "SAGA" algorithm), which is specifically designed for this project. SAGA is systematically evaluated using an open multimedia information retrieval dataset (Huiskes & Lew, 2008) and several heritage datasets from the UNE Heritage Centre to demonstrate the efficacy of the approach taken (University of New England, n.d.).

3.3.1 Sparse Existing Resources

In general-purpose multimedia information retrieval research, millions of Creative Commons-licensed Flickr images are available for research projects. As a further benefit, these Flickr images are likely to have some tags and other metadata already associated with them. This information can be used for training autonomous computational algorithms that can perform a wide range of functions that benefit users, including making it easier to search and explore the image collections.

Unfortunately, the same does not hold true for digital heritage research. A limited selection of digital heritage image collections are openly available to researchers, and although this situation is likely to change in the near future with heritage digitisation projects being conducted around the world, there is another, even scarcer resource: complete and openly-available sets of annotations, linking images in heritage collections to the semantic information that is relevant to them.

Without this information, it can be difficult to search or selectively explore these significant image collections, and information retrieval research in this area must first use human-based methods, where participants provide information for an algorithm, before software-based information retrieval methods can be used to provide useful information to the user.

To remedy the absence of available annotations, users could use existing tools to record annotations and associate them with images. These tools can perform simple tasks relating to the annotations, such as allowing images to be sorted or searched by different fields. A spreadsheet application, as a simple example, can perform this level of functionality. However, this simple approach is limited in that these utilities do not accelerate or support the annotation task, so the process is, for the most part, a manual endeavour. This effectively creates a barrier of necessary annotation work between the digitisation of heritage material and the application of fully autonomous algorithms, which would provide many benefits to the researchers, experts and volunteers working with this material.

With this in mind, SAGE has been designed as a utility which allows the rapid annotation of image collections by groups of human participants, and which leverages the annotations that have been contributed to projects as data that software algorithms can analyse to support and enhance the annotation process. In typical usage, participants will annotate several images in SAGE, at which point SAGE can begin providing suggestions for new annotations that can be added. Accepting the correctly-identified suggestions provides new information that SAGE can use for further suggestions, creating a positive feedback cycle that improves suggestion accuracy while contributing an ever-growing number of annotations to the collection.

3.3.2 Digital Heritage Images

One potential alternative approach to solve this shortfall of annotations is to rely on techniques which leverage the content of images to perform analysis, search and organisation services, rendering a time-consuming annotation process unnecessary (Datta, Joshi, Li & Wang, 2008). These content-based information retrieval (CBIR) approaches have been shown to be effective in certain situations, but they are more difficult to apply when used on heritage image collections than on modern image collections, as they encounter a range of unique complications.

Heritage image collections are often made up of digitised film photographs and other visual media, such as lithographs, glass plates, slides, drawings and other depictions. As such, the metadata which is usually available for digital photography such as GPS coordinates of where the image was originally taken and time of photo capture are unavailable. This prevents metadata-based CBIR approaches.

The various processes used to create the heritage images lead to diverse colour profiles depending on the format, photographic technology and printing processes that were used. This complicates colour-based CBIR approaches.

As the photographs are varied in terms of original medium, and as the original photographs are likely to have aged or deteriorated over time, it is likely that the images contain visual artefacts which obscure parts of the photograph, or degradation which reduces the clarity of the image. This makes saliency-based CBIR, object and facial recognition, and region-based image retrieval more challenging.

Additionally, due to the social differences between the modern day and the period in which the images were taken, there can be style and appearance differences in everyday objects, fashion, vehicles and technologies that would require re-training of object recognition algorithms.

While a content-based image retrieval approach could be used, an annotation-based approach is preferable, since it operates at a higher level of semantic complexity, allowing objects to be compared based on the context as well as content of the images as perceived by users, rather than analysing the image content alone.

3.3.3 Scarcity of Experts

An annotation-based approach relies on the participation of human experts and volunteers who can identify the significance and context of the collections being studied. This approach introduces a valuable new source of information and insight, but also necessitates consideration of the factors introduced by relying on human judgement for the function of the system.

The research focus of heritage images is more often the context or background of the event, rather than the physical objects present, with the image serving as a focal point in a larger discussion or a “lens” through which the past event is being viewed (Čermáková, 2012). Due to the “out of scene” nature of the complex semantic information being recorded, experts will normally be significantly better at interpreting digital heritage photographs than any existing computer algorithm.

The background of the heritage collection being studied is likely to be an uncommon domain of knowledge. While some topics are widely studied due to national or historic significance, a vast range of topics are of a personal or community focus. The number of historians who can contribute new insights to these images will of course be much smaller due to the specialisation required. Some heritage images will be of people or subjects which are very difficult to determine, and it's possible that some information has been entirely lost.

Just as there is a limited number of experts for photo collections of specialised subjects, there are also a limited number of volunteers who support them. Volunteers are more likely to choose to work on collections that have personal significance or interest, rather than on a need-driven basis. The interests of experts and volunteers drives the research process, and can result in the prioritisation of preferred projects.

Experts are able to contribute exact knowledge about the subjects, locations, events and contexts behind collections, while volunteers may target a simpler level of semantic complexity, instead identifying what they can see in each image and any apparent information, such as street or shop signs, which they are able to read. This results in different levels of annotation complexity when experts and volunteers work on the same project.

In addition to differences in content of the annotations produced by experts and volunteers, there is also a difference in the structure of these annotations. Experts are likely to follow a structured form of annotation which is in keeping with their field and training. If volunteers are also required to follow the dominant ontology used, it can make it more difficult for them to contribute annotations, and creates a barrier of entry for new volunteers.

Finally, experts and volunteers working on a particular collection of heritage images may not be physically in the same location. This means that any paper or file-based means of recording annotations must be periodically exchanged in order for information to propagate from one group to another. Either such arrangements need to be made, or experts and volunteers must agree on an Internet-based information repository, either in the form of a shared cloud storage location or distributed application, by which the annotations can be more easily shared.

3.4 System Design

The University of New England Heritage Centre has a research focus on digitised photographs (University of New England, n.d.). Since digital image representations can be made of a wide range of cultural heritage artefacts including film photographs, maps, posters, slides, glass plates, physical objects, heritage-listed buildings and more, this multimedia approach is extremely versatile in terms of allowing a large range of materials to be conveniently represented and electronically accessed in computer software.

The approach explored in this research utilises human-centric computation practices, which place users as a central focus in the design process. Specifically, the primary goals of this system relate to its intended usage in addition to its functionality: that relevant information will be easily discovered, and also that usage of the system will enrich the information stored by the system. These ideas seek to capitalise on the unique perspectives that every user will have of the digital images in heritage collections as the means of acquiring varied, semantically-rich and interesting information.

Considering the data storage requirements of the system, we can quickly identify that in addition to the digital photographs themselves, our system must store the knowledge contributed by users about those photographs, as well as the links between each piece of knowledge and the photographs to which that knowledge relates. As different investigations work with separate collections of heritage images, a means of separating related photographs and information is highly desirable. Since different permissions will apply to the separate collections depending on the role a user has in these collections, information about users must also be collected.

These data requirements can be met by creating a small number of distinct but interconnected classes. Since these classes exist in an interaction-driven environment, an agent-based approach is an appealing and effective architectural style to use. Agent-based approaches treat each instance of a class as a container of information and a set of behaviours that are associated with this information. These behaviours are triggered by external events in the agent's environment, such as the addition of new agents, allowing the agent to update itself and interact with other agents in response to this stimulus until an equilibrium state is reached, providing a means of information self-organisation.

These agents could be stored in a centralised or decentralised manner. In a decentralised system, users would install software that networks their computers and allows for peer-to-peer (P2P) transmission of information between agents. In a centralised system, all agents would coexist in a single location maintained by an authority upon whom all other users would rely. In our case, an authority clearly exists, namely as heritage institutions that specialise in collecting, documenting, organising and disseminating heritage material. This encourages the use of a centralised system over a P2P system to grant these institutions the control they require to effectively carry out their work, managing concerns such as usage permissions, copyright, and responding to queries from the public.

Ideally, we want the system to be widely available to a large audience of users, but have a centralised database. It should be low in cost, rapidly deployed, easy to learn, simple to use, and be in a similar environment to where work is currently being conducted. The best choice meeting those criteria is to develop this system as a web application.

3.5 Agent Types

As identified in the system design, SAGE is made up of four different kinds of software agents: projects, digital objects, concepts and annotations. These agents are responsible for storing information relating to their concerns and interacting in a manner which steadily captures knowledge in a semantic network formed by annotation-linked objects and concepts, making this information available to end users to explore.

SAGE also represents the human users as agents in this multi-agent system. Each human user has a digital representation in SAGE that stores their identity and their permissions, which allows users to interact with certain projects in an authorised manner. This allows permitted interactions initiated by users through their user agent to communicate with other software agents, such as through creating new objects, concepts or annotations. The other agents can then attempt to respond to the user in an intelligent manner, such as performing suggestion calculations for whatever object or concept the user is currently viewing and interacting with.

3.5.1 Digital Objects

Fundamentally, we need a way of representing the items in the heritage collection being annotated. These items could be any form of text, document or multimedia object, referred to in SAGE as digital objects or more succinctly, objects. This still requires some piece of unique information that distinguishes one object from another, however.

One option for a unique piece of information would be a binary representation, the data of the object itself, though storing a large number of objects would exhaust the available storage capacity, and isn't strictly necessary if the data exists elsewhere and is simply being referenced.

A second approach would be to represent the object by an identifier such as a name, though this does not allow the object to be accessed conveniently. For example, we might associate an image with the name "GP_0333.jpg" with an annotation describing it as being "A Steam Ship With Passengers", but actually viewing the image requires navigating into the directory structure on a hard drive to find the image to open it.

Fortunately, an URL-based approach allows objects to correspond to a web-accessible resource that allows it to be uniquely identified and accessed on demand. Each object would store a URL like "http://www.example.com/images/GP/GP_0333.jpg" which both identifies the resource (URLs can only link to a single location) and provides a link that the user can visit in their browser to see the image in question. An URL-based approach is what is used by Objects in SAGE (see Figure 3.2).

Digital Object
• location:string • filename:string • thumbnail:string • project:Project • annotations:Annotations • concepts:Concepts
• thumbnail():Thumbnail • flatten(DigitalObject)

Figure 3.2 - Digital Object UML Class

Location

Since the URL provided for an object may be appropriate for the web, but not refer to the actual location of the resource which is being represented, a number of filters are run against new objects to determine if their URL should be adjusted. For example, a cloud data hosting service such as Google Drive might provide an URL that links to their web viewer for a given image, when what is desired in SAGE is a direct link to the image. Fortunately, this can be automatically reconciled by querying the image host for the actual image URL when this is detected.

Filename

Further to this location, we are accustomed to filenames being a way of identifying objects on our local systems. Including a filename field in our objects allows this to be conveniently stored and retrieved on demand.

Thumbnail and Thumbnail()

Objects can be of a multitude of multimedia types, and a common manner in which these objects can be concisely represented is in a thumbnail, providing an insight into what the object represents. Since the thumbnail, while associated with the resource being represented, is itself a separate resource with its own URL, we must add this field to objects in order to retrieve the thumbnail when needed.

Some resources can be represented quite easily as thumbnail images since they themselves are images. In some cases, it can be more challenging to represent a resource when that resource is complex, such as a database file, a webpage or a document. The choice of sensible default thumbnails is left up to a thumbnailing subsystem, which is described later in this chapter but is accessed through a method that the object holds.

Project, Annotations and Concepts

Objects need a project id to associate with their containing project. This allows objects to be grouped by their container project and restricted to authorised users, rather than all objects being visible to all users. An object also has direct access to the annotations that have been applied to it, as well as the concepts that are connected to it through annotations.

Flatten()

Objects are unique in a project based on their URL. If multiple objects with the same URL exists within a project, they will be automatically merged into a single object that retains the annotations assigned to the separate objects. This is particularly useful when importing objects from an external source, such as synchronising the information stored in SAGE with changes made outside of the application. It allows new objects and annotations to be introduced while preventing the creation of duplicate objects when they already exist.

Behaviours

While not part of the Object class itself, all objects can be wrapped inside a SimilarityAlgorithm-derived class to allow objects to be compared based on their similarity as detected by whatever algorithm is employed. Essentially, the SimilarityAlgorithm class provides the logic, but does not possess any data to operate upon. Wrapping it around an object provides that data, and any similarity measurements will be centred around the object that has been provided.

3.5.2 Concepts

Having stored objects, which are references to web-based resources located through URLs, we also need a means of representing the semantic information we are capturing from users that describe the characteristics and features of these objects.

SAGE uses a tag-based system for representing these concepts, with an emphasis on semantic tags where possible. Text descriptions used in these semantic tags can range from single-word keywords through to descriptive paragraphs, depending on the user's preference. These are referred to as concepts in this work (see Figure 3.3).

Concept
• description:string • project:Project • annotations:Annotations • digital_objects:DigitalObjects
• matching:Concepts • merge(Concepts) • flatten(Concept)

Figure 3.3 - Concept UML Class

Description

Concepts are expressed in uncontrolled language, which contrasts with the formal descriptions, hierarchies and ontologies which are often used to record information about heritage materials. While this means that the data collected in SAGE cannot be directly merged with archival records kept on a heritage collection, it provides a number of benefits to the users of SAGE.

Using uncontrolled language means that the contributors of annotations won't be expected to know and conform to institution-specific practices, and we do not need to implement various checks and safeguards to verify that the data collected from volunteers has a particular structure or consistency. These benefits of easier annotation collection lend themselves well to an exploratory or information gathering approach to research on a heritage collection, which is closely aligned with the goals of this project.

Project, Annotations and Digital_Objects

Concepts, like objects, need to store the identity of their containing project. While early designs of SAGE allowed concepts to be shared between all projects so that the concept only ever needed to be defined once, it was found to be too inflexible, as any changes or additions made to the concept to better suit it to one project may make it less suitable to all others in which it is used. For this reason, each project has their own, separate concepts. A concept also has direct access to the annotations that have been applied to it, as well as the digital objects that are connected to it through annotations.

Merge()

Two distinct concepts can be merged together when they represent a synonymous concept. Their annotations will be combined and their descriptions will be appended to form a single description that can be refined by the user.

Matching() and Flatten()

Concepts are unique within a project based on their description. If two concepts are created with a matching description, they will be indistinguishable from one another, and thus a flattening behaviour will be triggered by the concept that detected the match. This automatically combines the unique associations of the two concepts into a single concept. Correcting spelling mistakes in a description may also trigger a flattening operation. The same behaviour allows concepts to be selectively merged by updating the descriptions to match, such as when reconciling two synonyms into a single concept.

This behaviour is particularly useful when multiple users inadvertently use the same description to annotate the objects in the project they are working on. This is a simple way by which links to separate areas of work can be established without requiring any additional thought on part of the user, as one user's work will now be connected to the other user's work through the flattened concept.

Behaviours

Concepts can be wrapped in a SimilarityAlgorithm-derived class to allow them to be compared based on similarity, much in the same way that an object can be wrapped in a SimilarityAlgorithm. Concepts will usually require different logic for determining their similarity than objects, so each SimilarityAlgorithm will typically provide two specialised subclasses, one for objects and one for concepts.

Concepts in SAGE are semantic tags, as opposed to lexical tags, and their behaviour is designed in accordance with this. Specific characteristics that are expressed include that:

- Concepts are their own class, not simply metadata of another class.
- While editing a lexical tag will only change that individual tag, editing a semantic tag such as a SAGE concept will cause all tags sharing that description to be changed.
- If two concepts share the same description at any point in time, they will be merged into a single concept. Lexical tags are distinct from one another.
- To disambiguate two distinct concepts, different descriptions must be used so that the concepts can be distinguished as having unique meaning. Lexical tags can be polysemous, with ambiguous meanings determined by context.

3.5.3 Annotations

An object may have a number of different concepts which are relevant to it, and the semantic meaning behind a concept may be present in a number of different objects. All that is missing is a means of associating an object with a concept. In this research, these associations are referred to as annotations. Any object/concept pair may have an annotation linking them together (see Figure 3.4).

Annotation
• project:Project • digital_object:DigitalObject • concept:Concept • user:User • provenance:String

Figure 3.4 - Annotation UML Class

Project and User

Annotations have a containing project, since both the object and concept it associates will belong to the same project. Annotations also have a user, who is the one who originally created the annotation. This allows the origin of annotations to be identified.

Digital_Object and Concept

Annotations may either exist or not exist between an object and a concept. If multiple users simultaneously assert that an annotation should be made for a given object-concept pair, only the first assertion received by the system will be actioned. Subsequent requests will discover that the annotation already exists, and will have no effect. This helps to avoid duplications from being created when considering the asynchronous nature of requests sent from the web.

SAGE does not have an equivalent to a “negative” annotation. Instead, SAGE encourages the creation of a new concept that captures the semantics behind this negation. For example, images that contain a building may have the “Building” concept, and those that do not will not. Those images that do not have a building might have a “No Buildings Present” concept applied to them, if that is significant, or perhaps a more useful “Landscape” concept, if that holds true. One only has to consider the case where a single image in a collection contains a “Flamingo” to see that a policy of annotating every other image with NOT “Flamingo” would be both wasteful and tedious.

SAGE does not seek to have annotations confirmed by different users, as doing so provides little additional information and scales inefficiently to large projects with a small number of users. While this could help to resolve conflicting annotations (e.g. “8 out of 10 users think that this annotation is correct”), SAGE encourages conflicts to be documented in an annotation that highlights the controversy or uncertainty (e.g. “May either be Saumarez Homestead or a building with similar architectural style.”).

Provenance

Annotations can have different provenances depending on how the user added the annotation. Some annotations can be created directly on an image, while others will be added from suggestions made by the system to the user. Finally, some annotations may be created automatically in operations such as copying a project. These provenances are recorded primarily for evaluation purposes.

Behaviours

Annotations do not have any specific methods, but instead respond to events controlled from the agents they reference. Annotations may also become orphaned if the user that created them deletes their profile. Annotations are destroyed, however, if either the concept or object they link together is destroyed. This is because annotations have no meaning when one of these is removed, but is still useful even if the original annotator is no longer involved in a project.

3.5.4 Projects

It is possible that many separate heritage collections are being actively worked on in SAGE at any one time. We require a means of collecting objects, concepts and annotations into private groupings that have controlled access and permissions. This is achieved by using projects (see Figure 3.5).

Project
• name:string • notes:string • algorithm:string • parent:Project • children:Projects • viewer_key:string • annotator_key:string • contributor_key:string • administrator_key:string • concepts:Concepts • digital_objects:DigitalObjects • annotations:Annotations • users:Users • user_roles:UserRoles
• check_key(User,string) • generate_key(string) • reset_key(string) • clone(User) • sample(User) • pull(Project) • import_annotations • export_annotations

Figure 3.5 - Project UML Class

Name and Notes

Projects need to have a name that helps distinguish them from one another. Projects also have a notes section that allow the project's administrators to communicate welcoming and induction information to new participants in the project, such as copyright information, rules of use, general instructions for contributing and contact details.

Algorithm

Projects store the currently selected suggestion algorithm used by objects and concepts in that collection. This allows different algorithms to be used in different projects, and for this algorithm to be changed when desired by administrators. This is particularly useful for evaluations which compare two similar projects with different algorithms for usage and productivity differences.

Parent, Children, Clone(), Sample() and Pull()

Projects can have parent-child relationships with one another. A child project can be created as a partial copy (sample) or full copy (clone) of the parent project, and the child will store the parent project's identity when it pulls copies of data from the parent. This feature is useful when performing evaluations on a project, as it allows the original to remain untouched while a snapshot copy is produced for any potential modifications required by the evaluation processes.

Keys, Check_Key(), Generate_Key() and Reset_Key()

As projects are managed by a key-based security system (which is described later in this chapter), each project must store the currently valid key for each key type. Projects maintain records of who has redeemed keys and the positions they hold in each project in a separate UserRole class.

Users, User_Roles, Concepts, Digital_Objects and Annotations

Projects have access to the users, user roles concepts, digital objects and annotations that fall within or relate to the project. Project agents are able to compute project-wide statistics using this information that aren't directly accessible to individual objects, concepts or annotations, such as the popularity of a particular object or concept within the project. This allows project-wide statistics to be used as a useful element in inter-agent interactions such as finding relevant suggestions to provide to the user.

Import_Annotations() and Export_Annotations()

Projects have importing facilities to allow information to be drawn into the project rapidly and conveniently. Common import cases could include pulling a CSV backup of another project's data into the current project to consolidate the two projects, or accepting new information from an external source using CSV as a data interchange format. Conversely, projects provide an exporting facility to record its annotations in CSV format, both for use in SAGE and use by external programs.

Behaviours

Projects destroy their contents when deleted, preventing inaccessible, orphaned concepts, annotations and objects from populating the database. These orphaned entities would be completely inaccessible by users and would only serve to unnecessarily occupy space in the database. As users can be part of several projects, they are not affected when a project is deleted, though any roles and permissions they were granted for that project will be lost.

3.5.5 Users

While the other agent types can attempt to infer information from stored data, the user alone is granted the ability to contribute new information and confirm speculative connections between objects and concepts. The user is particularly valuable to SAGE as they are able to interpret objects with a degree of comprehension far beyond what is computationally possible, and users can draw on their past knowledge and experience to bring to light new facts about the collection being examined (see Figure 3.6).

User
• name:string • email:string • password_digest:string • user_roles:UserRoles • annotations:Annotations

Figure 3.6 - User UML Class

Name, Email and Password_Digest

To identify the user, we ask registrants to provide their name, email address and a password as a means of identifying and communicating with them, as well as supplying the information required to authenticate them on subsequent visits. A password digest produced using bcrypt-ruby (which includes a salt) is stored rather than a password, following good security practices (Hale, 2009). Password digests are like recognising a name rather than remembering it; you'll know when it's the right one, but you can never be maliciously compelled to reveal it, since you never memorised it in the first place.

User Roles and Annotations

Users must provide consent to participate in the SAGE research project as part of the terms and conditions of using SAGE, the fact of which is recorded implicitly in the act of creating the user account. Users who do not consent to the terms are not able to create an account. A user account permits keys to be redeemed in exchange for user roles that grant levels of access in project, allowing a user to contribute annotations which are linked with their account.

Behaviours

Rather than having much in the way of automatic behaviours like other agent types, users acting through their user agents are the source of the initial interactions which cause the other software agents to communicate with each other, much like a ripple effect in water, producing new information in the form of suggestions for the user to consider.

3.6 Interface Design

Having established the types of agents which will store the vast majority of the information contained by SAGE, we now need to investigate the manner in which users will access and contribute to this information.

Since we are developing for the web as a platform, we must consider two main aspects regarding how we design our interface. The first is to consider the routing scheme used to specify the path and action of HTTP requests, and the second is to consider how the individual pages will appear to the end user, including the information displayed in each, and the controls available to the user for further interaction.

3.6.1 Routing

Approaching the system design from an agent-based perspective makes it simple to design the routing interface by following resourceful routing practices (Rails Guides, n.d.), with each agent being treated as a resource. Resourceful routing allows a set of RESTful routes (Fielding & Taylor, 2002) to be generated for a resource, which permits HTTP verbs such as GET or POST to be directed to either a collection of resources or to an individual resource in order to indicate the action that should be performed on that resource. While most actions can be indicated using just this information, in some cases further path or query information can be provided for more complex or specialised actions, like adding annotations in SAGE.

In this way, we can interpret GET requests to /projects/1/concepts/123 to refer to the 123rd concept in the first project, or /projects/1/concepts/ to refer to a listing of all the concepts in the first project. POST requests to /projects/1/concepts/ would indicate that the attached data is intended to be used to construct a new concept.

In a more complex example, projects/1/concepts/123/add_object?object_id=321 would indicate that the 123rd concept will be annotated to the 321st object. Along with a small amount of data from the user's browser to confirm that they are logged in as an authenticated user of SAGE, this provides everything needed to perform this kind of complex action.

These examples present a somewhat simplified version of the actual URL, as the host would precede the relative path. This would depend on the hosting decisions made for the web application. For example, the full URL for accessing the first project might look like: http://sage.example.com/projects/1

3.6.2 Views

With the crucial decisions already made regarding routing, we can now investigate how each of the individual user interface pages will be laid out. These pages are referred to as being *views* of the resources they represent. SAGE is composed of several important views, including a static, non-changing collection of site pages that provide general information about SAGE, as well as dynamic, changing views for users, projects, objects and concepts.

Some of the agents and supporting components discussed in the previous section, such as thumbnails, user roles and annotations, are not accessed directly by the user via web interfaces, but are instead created and updated implicitly during the interaction of users with the other software agents in the web interface. For example, annotations can be added or removed from the object or concept views, and as such annotations do not require views of their own.

Site Pages

The site pages include a welcome screen, registration and login forms, user guides, feedback and contact details. These pages are designed to encourage users to discover what SAGE is, create a user profile (refer to Figure 3.7), learn how to use SAGE and participate in the project as a contributor of new information.

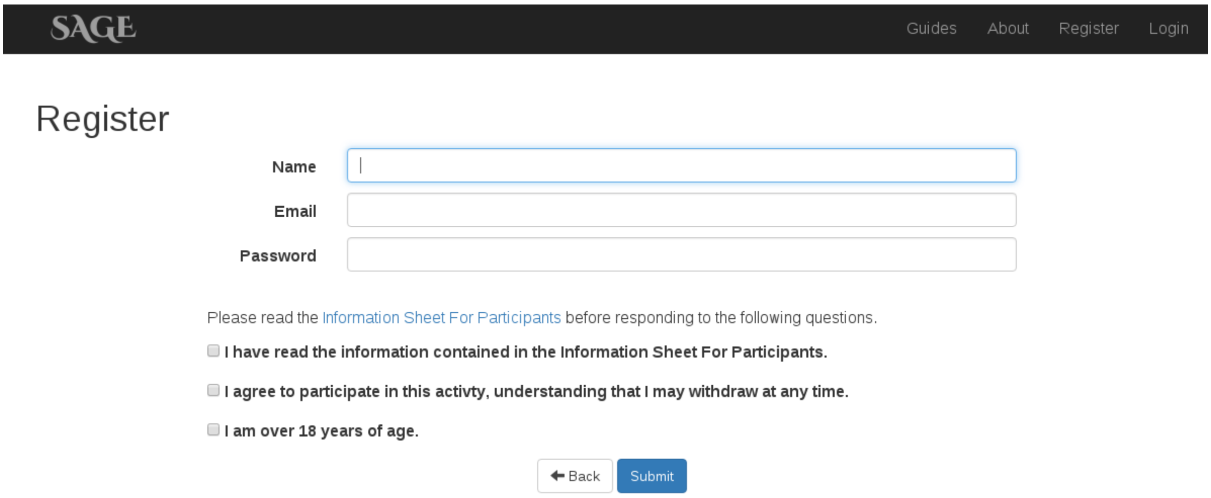


Figure 3.7 - SAGE Registration Form

Users

The user profile provides a view of the details stored by a user agent. It allows a user to review this information and make corrections where necessary (see Figure 3.8).

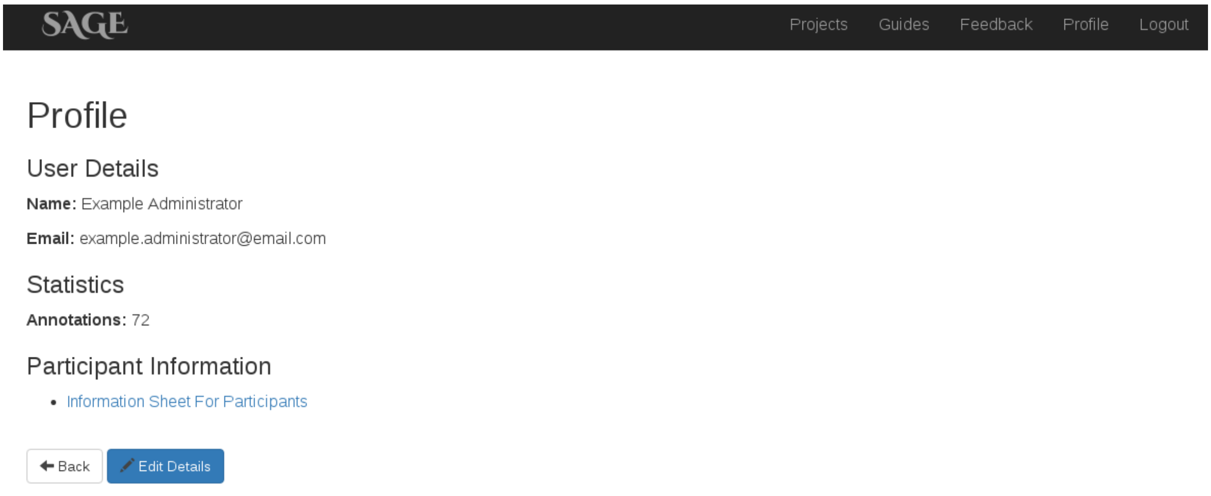


Figure 3.8 - SAGE User Profile

Projects

The project listing allows a user to see the projects they are involved in, create new projects or redeem keys they have been granted for existing projects. From this page, users can select to leave projects they are no longer interested in, which may result in those projects being deleted if the user is the last administrator of those projects (see Figure 3.9). Users can generate an administrator-level key to establish a suitable successor in order to avoid this happening.

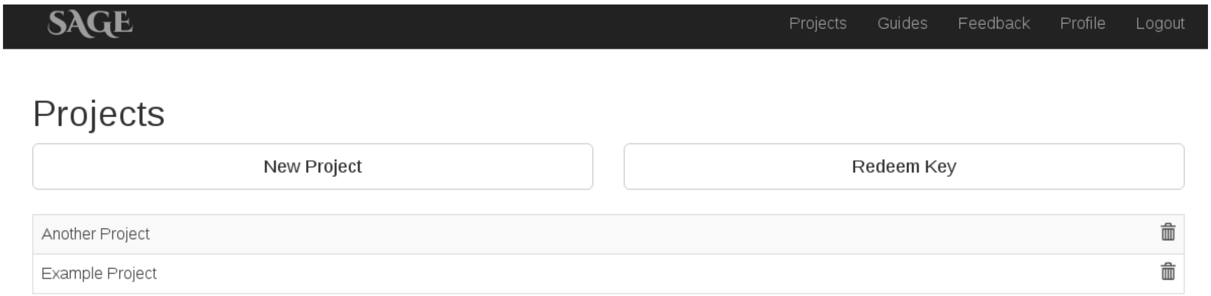


Figure 3.9 - SAGE Projects Index

Selecting a project brings up the project notes, which provides an overview of the project for viewers or contributors, and gives administrators fine-grained control over who is participating in the project and how new participants can access it (see Figure 3.10).

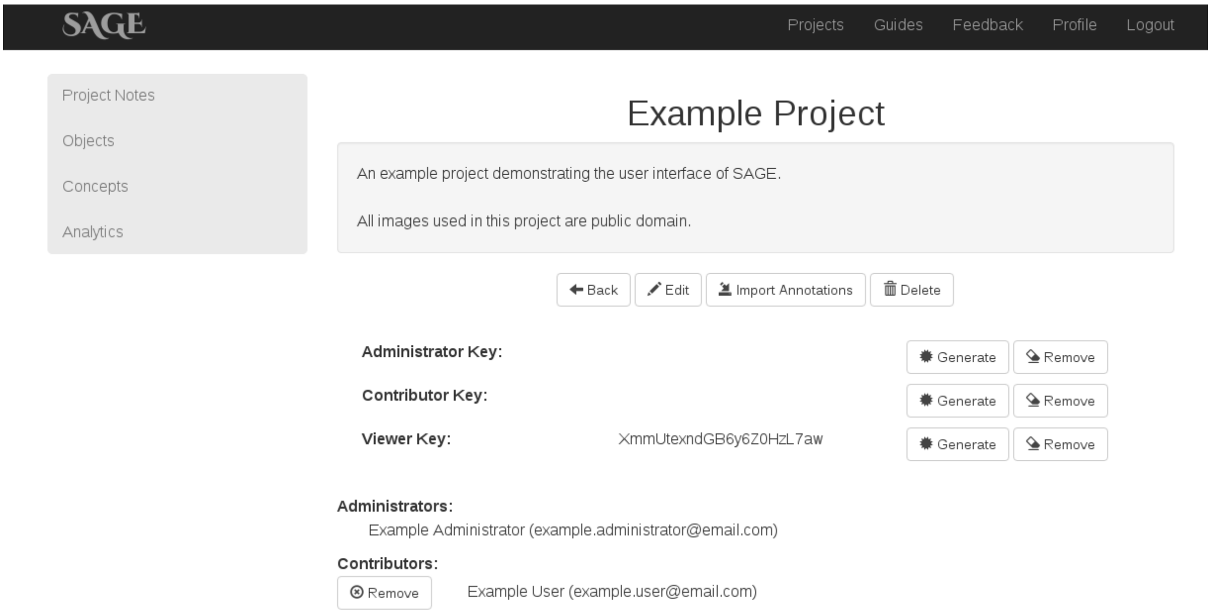


Figure 3.10 - SAGE Project, Administrator View

Providing access to new participants is achieved via the key-based security system. Selecting the “Generate” or “Remove” buttons next to the different key fields either produces, alters or removes the key, depending on which option was selected and whether a key already existed for that field. Administrators are able to see who has redeemed keys and the roles they fill. If an administrator wishes to remove a selected user, they can do so from this screen by pressing the “Remove” button next to their name.

In designing SAGE, the decision to not publicly list projects was made in order to preserve the privacy of personal projects. This means that there is no comprehensive list of all the projects in SAGE. Since users cannot self-enrol in a project anyway, this has no impact on the usability of the system, and recruitment of additional users for a project is on a word-of-mouth basis. Thus, the person who initially tells a user about a project is either able to give the user a key directly, or can pass along the contact details of someone who can. This friend-of-a-friend recruitment strategy helps to establish trust and accountability among participants.

Users of the system are also restricted to seeing only the names and contact details of the project administrators, intended for communication purposes. The identity of co-contributors is hidden for privacy reasons, as while the administrator has control over who is invited to the project (via giving out keys) and therefore implicitly shares their contact details with new participants, a co-contributor of the project does not have a chance to approve or reject this implicit permission as they are not responsible for selecting the group working on a given project. This privacy measure means that all communication must go through administrators, unless administrators reveal who else is working on a project to allow for cross-communication between contributors.

Selecting a project also brings up the project-specific navigation bar, which allows users to move between the project notes, objects, concepts and project analytics. This is the sidebar visible on the left in Figure 3.10 and in following figures.

Objects

The object index provides a list of all the objects that have been added to a project sorted in reverse chronological order of addition, so that the newest objects are the first to be shown in the listing. This helps users keep track of ongoing additions to a project and also directs attention to the new objects, which are most likely to need further annotation work (see Figure 3.11).

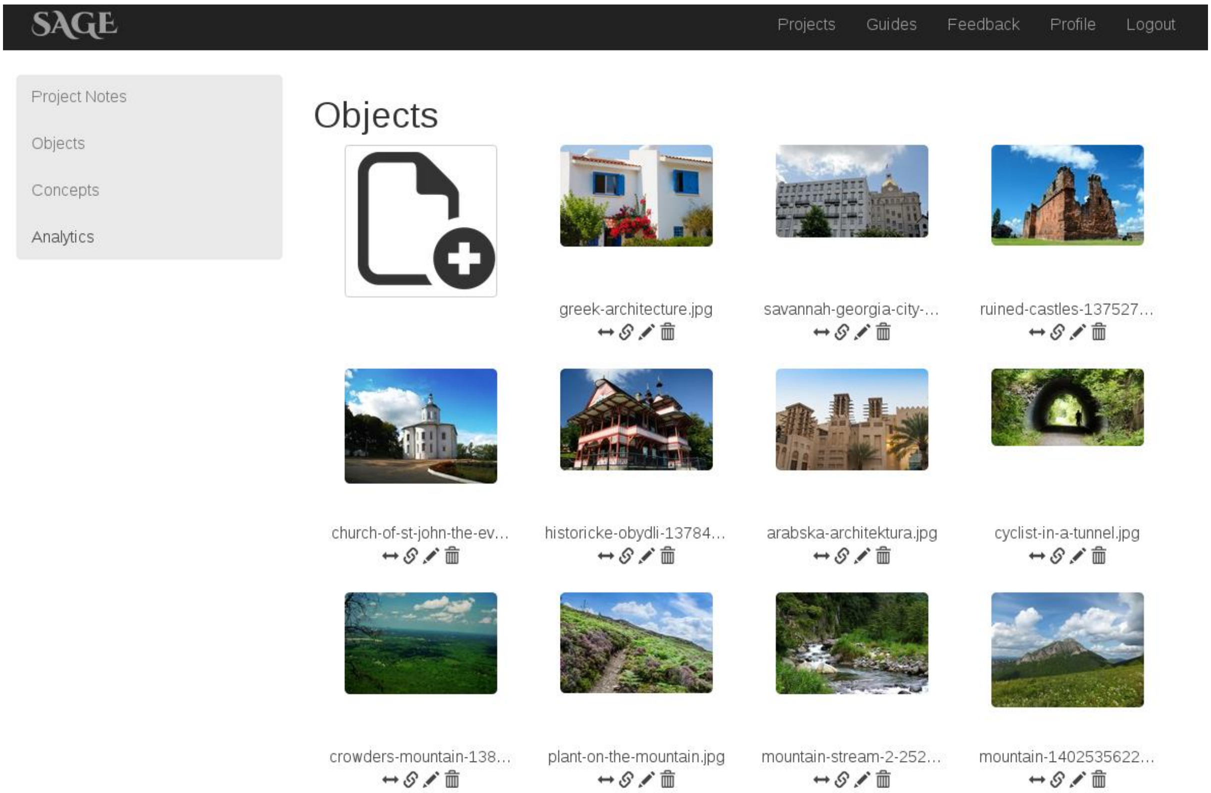


Figure 3.11 - SAGE Object Index

The layout of this listing is presented in the style of a gallery. Each object has a representation made up of its thumbnail and filename that links to the object's specific view. Longer filenames are truncated to prevent them from overflowing the object's tile in the gallery, as can be seen in several of the filenames in Figure 3.11 that end in "...".

The thumbnails are displayed in a fluid container that adjusts the number of columns depending on the size of the device used to view it, with smaller devices such as smartphones displaying the list in a compressed layout (refer to Figure 3.12), and larger devices like laptops and desktops showing the gallery across several columns.

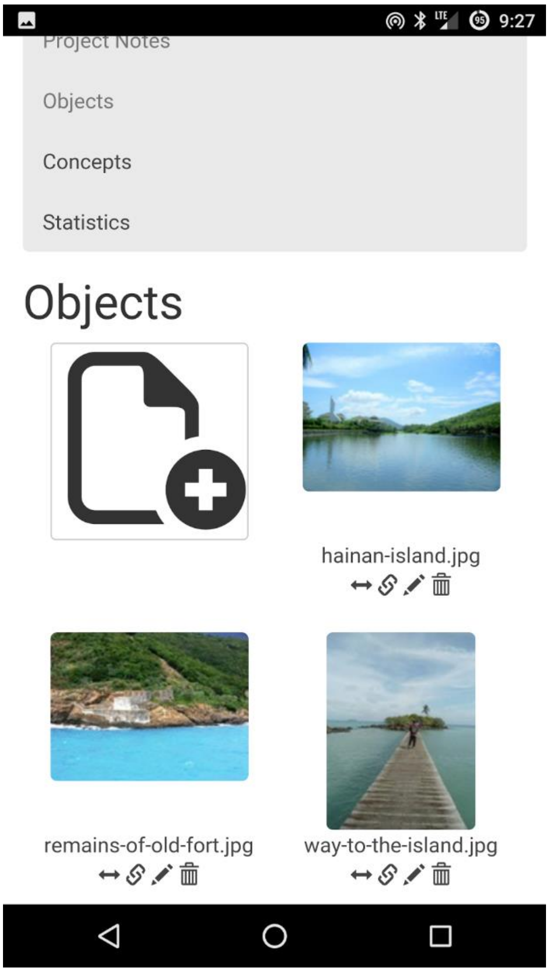


Figure 3.12 - SAGE on a Smartphone

As the number of objects in a collection can be quite large, the listing is broken into pages. For all but the last page, a “Next Page” button is shown, and similarly for all but the first page, a “Previous Page” button is shown. In the event that the collection can be shown on a single page, neither button will be present. Users can manually specify a page by modifying the URL if they wish, though if they attempt to visit a page outside the normal range between the first and last page, the page they access will display no objects.

The very first item in the gallery is a button allowing users to add new objects. New objects can be specified by their URLs, with multiple URLs separated by new lines to allow a batch import. Users are also able to temporarily authorise tightly-restricted access to their Google Drive account so that SAGE can be provided with the file ID of a public folder, from which it can import all the files it contains into the current project. This is useful when an organisation has hundreds or thousands of heritage images in a Google Drive folder, which would take a large measure of time to import one by one.

Selecting an object will show that object along with a set of controls for editing or deleting the object. The controls also provide the user with a means of moving between previous and subsequent objects to view them one-by-one rather than selecting them from the index.

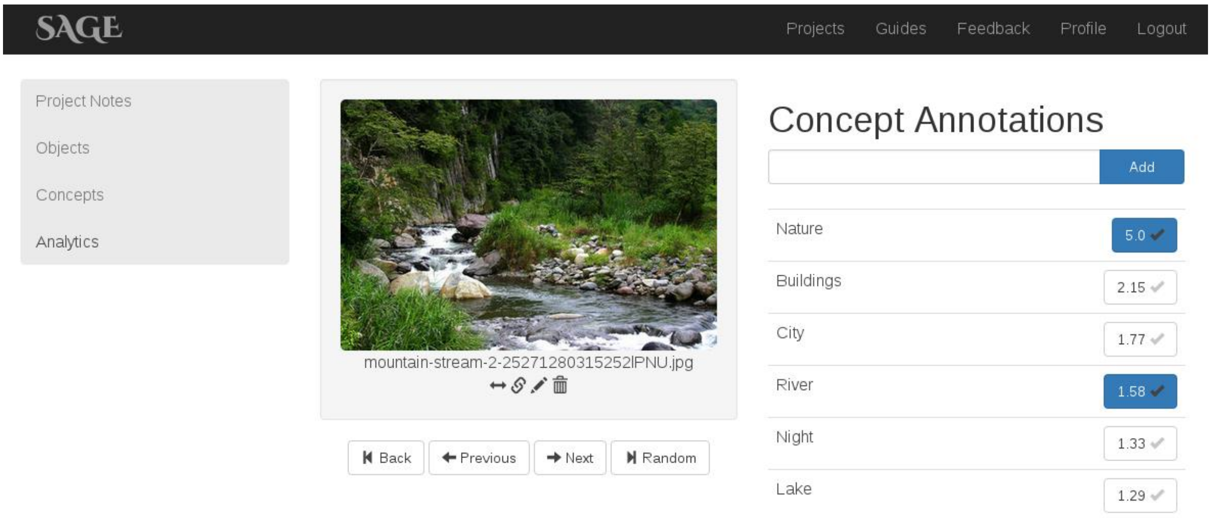


Figure 3.13 - SAGE Object View

The annotations and suggestions for new annotations will also be shown in the object view (see Figure 3.13). The suggestions provided will depend on the suggestion mechanism being used in the project. Each annotation will have an opaquely shaded button next to it which can be clicked to remove the annotation. Likewise, suggestions will have a transparently shaded button that can be clicked to add the suggestion as an annotation. The number in the button reflects the score that the suggestion mechanism calculated for that suggestion, which will vary depending on the suggestion mechanism used.

Users are also provided with a simple form at the very top of the suggestions and annotations list to submit and annotate entirely new concepts to the object, particularly relevant to the early stages of annotation or when novel objects have recently been added and the concepts needed to describe them must be created. This reduces the cognitive load on users as they do not need to remember the details of the image and visit the concepts listing to create new concepts, and they can conveniently examine the object while adding concepts instead.

Concepts

The concept views in a project are very similar to the object views. In the concept index, all of the concepts that have been created in the project are displayed in alphabetical order as a list, with each line of the list linking to an individual concept. Particularly long concept descriptions are truncated (see Figure 3.14).

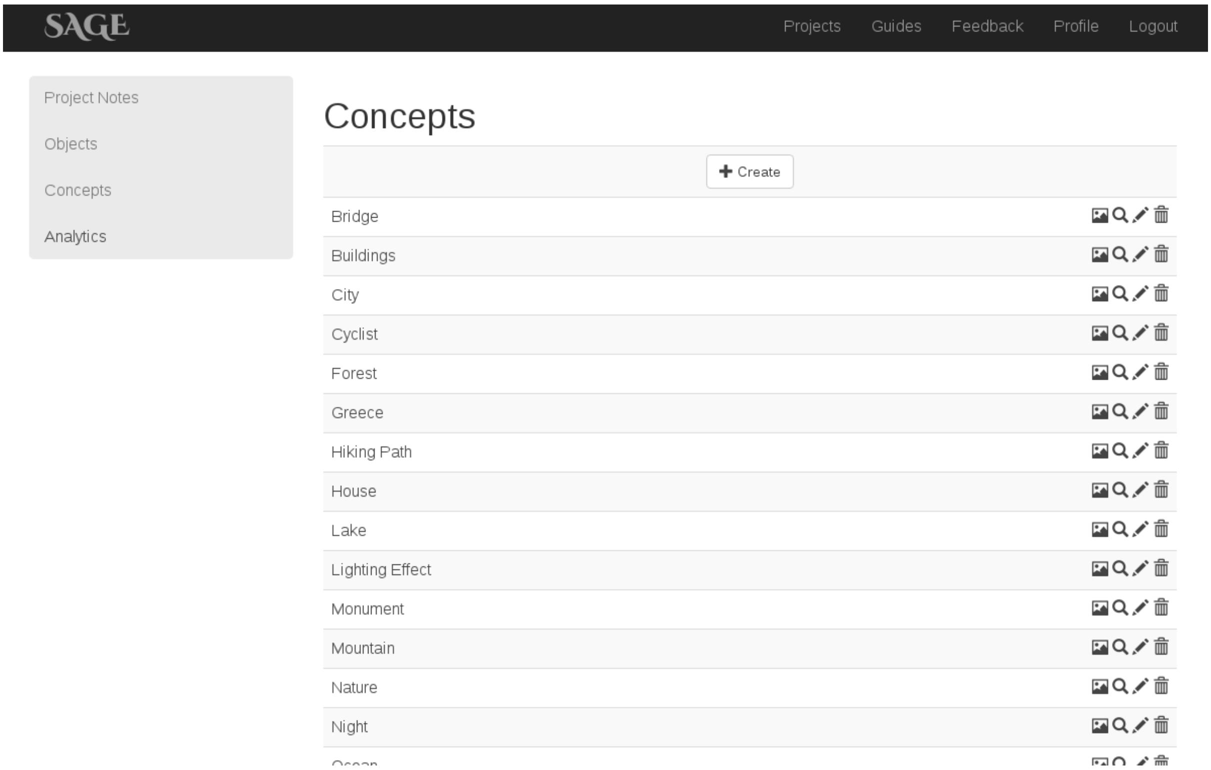


Figure 3.14 - SAGE Concept Index

In the event that a large number of concepts exist within a project, the list is broken into several pages which can be navigated using a “Next” and “Previous” button, similar to the object index.

The very first row of the concept listing shows a button to add new concepts. Concepts are added one at a time, with the submission form requiring a non-blank description of the concept to be provided by users. Upon submitting a new concept, the user is directed back to the concept creation screen to allow for multiple new concepts to be added in sequence. Since the description field is highlighted on loading the page, and since pressing Enter will submit the current description, multiple concepts can be added quite quickly.

Each concept has a set of quick controls accessible in the index or in the concept’s individual view. These controls allow a concept to be edited and deleted, and also allow the concept to be used as the search query for a Google search or Google image search. This provides a means of conveniently drawing on information available from the wider Internet to expand or enhance a SAGE project.

Selecting a concept shows that concept and a set of controls that allow the concept to be edited or deleted (see Figure 3.15). Navigation controls are provided so that concepts can be viewed one-by-one rather than by selecting concepts from the index.

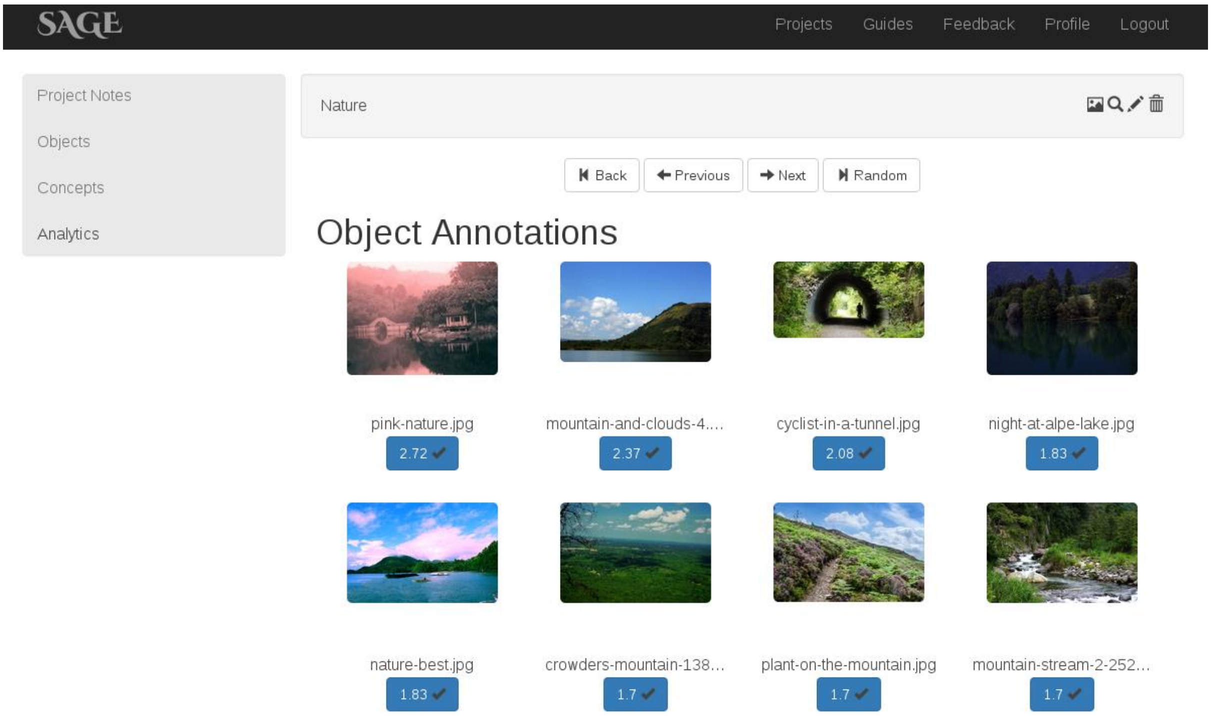


Figure 3.15 - SAGE Concept View

Below the concept controls all the objects that the concept has been annotated to are shown, along with suggestions for additional objects that the concept might be relevant to whose accuracy will vary depend on the suggestion mechanism used. A transparent or opaquely shaded "Add" and "Remove" button system is employed to add or remove annotations in a manner similar to the object views. Scores are also shown as calculated by the suggestion mechanism.

The annotation and suggestion objects are shown in a gallery with similar properties to the object index, specifically that the number of columns dynamically adjusts to suit the size of the display. Mobile devices will show the gallery in a single column, whereas desktop browsers will show it in a grid.

3.7 Ancillary Systems

Beyond the agents and their views, a number of supporting systems were utilised or developed to allow SAGE to carry out its intended function. These ancillary systems include application hosting, data storage, thumbnailing, project importing and an authentication system. These are commonly used by web applications, as they provide supporting functionality relevant to a wide range of application ideas.

3.7.1 Web Hosting

SAGE is built using a full Ruby on Rails stack hosted on Heroku using a Heroku PostgreSQL database (Heroku, n.d.). Heroku is a popular cloud-based Platform as a Service provider for Ruby on Rails web applications, allowing them to be easily maintained and hosted on the Internet, and Heroku's PostgreSQL database is the preferred method of efficiently storing large amounts of data when using Heroku. SAGE requires two processes to function: a web process to handle and serve incoming HTTP requests from the web, and a worker process to perform delayed jobs used by several other ancillary systems, such as generating thumbnails. Both of these worker types are specified in a configuration file and managed automatically by Heroku.

When performing operations using the evaluation subsystems, one-off processes are used rather than running the evaluations on-line through the web interface. This helps to avoid blocking traffic to the application while the evaluation finishes, which can take a number of minutes for larger datasets. The one-off processes are the default behaviour on Heroku when spawning an interactive Ruby shell (the irb application) in the terminal or when performing Rake tasks to run evaluations (similar to the automated tasks that can be performed in makefiles using the make utility of many Unix-like systems).

When running particularly large user-oriented operations, such as the experiments conducted in the following chapters, Heroku can be configured to provide additional web or worker processes. Additional web processes help deal with higher traffic, while worker processes are particularly useful when importing large collections of objects in order to generate their thumbnails in a timely manner using the image manipulation library RMagick (RMagick, n.d.). This makes it easy to temporarily scale SAGE to handle large-scale user evaluations.

Heroku runs web applications following the principles of a 12-factor app, a set of guidelines that apps can follow to be extremely scalable (Wiggins, 2012). One such principle is that no process should store persistent local data, since this data is very likely to be destroyed when the website administrator scales or restarts virtual machines, and the local data can't easily be shared between virtual machines. This necessitates additional subsystems to handle the shared, long-term storage of data, such as a PostgresSQL database for application data, and external multimedia data storage like Google Drive for original images and Amazon's Simple Storage Service (S3) for thumbnails (Amazon, n.d.b).

3.7.2 Data Storage

When working with large collections of photographs or other multimedia files, we need a way to make them accessible to SAGE for importing. Data which is already present on the web poses no issue, since it already has a URL that can be used in SAGE, but if we wish to work with local files, we first need to upload them to a web-accessible location.

There are several requirements that direct our choice of web storage systems. The chosen system need to provide a large amount of cost-effective storage, since collections can be hundreds of gigabytes in size. The chosen system should also be user-friendly with a reliable upload mechanism, ideally which can be run as an interruption-safe background process that avoids failed partially-complete uploads. Finally, the chosen system needs to have a suitable security setting that allows data to be made accessible on restricted terms, specifically to only those users who know the file's unique and hard-to-guess file identifier so to preserve the potentially confidential or restricted nature of the heritage images.

There are several prominent commercial systems available to select from that satisfy these requirements to various extents, including Google Drive, Dropbox, OneDrive and Box (Google, n.d.a; Dropbox, n.d.; Microsoft, n.d.; Box, n.d.). Google Drive provides the flexible security settings and storage space that are required by SAGE, and additionally is a mature cloud storage platform, thus was selected as SAGE's primary data storage system.

Users are able to upload images to their personal Drive, mark them as public, and then provide that file's shareable URL in the SAGE new object form to create a new object linking to that resource. Additionally, users can upload a collection of images to a folder in Drive, mark the entire folder as public, and then grant SAGE permission to scan the folder for each objects' shareable URL for inclusion in the selected SAGE project. This allows entire folders to be imported into SAGE in one action, and for any new files in those folders since the previous import (if this import is not the first) to be used as the basis for new objects, which safely avoids creating duplicates.

Google requires that web applications identify themselves as part of this process for billing and security purposes. In return, Google allows applications to retain permissions granted by users to simplify subsequent actions on the user's files. SAGE opts for a single-use permission system that persists only as long as the user is logged in to SAGE, as import requests are infrequent. Users can grant this permission again in subsequent visits.

3.7.3 Thumbnailing

While the approach of using Google Drive to upload and host content on the web solves the logistical issue of data storage and data accessibility, downloading full images when showing gallery views of the data stored in SAGE isn't feasible when each high-resolution photo could be several megabytes in size. While Google Drive does produce thumbnails for images stored in Drive, a system was needed that could support files hosted outside of Google Drive, as well as files that Google Drive doesn't normally produce a thumbnail for.

The solution to this was to develop a thumbnailing system using RMagick and Amazon's S3 cloud storage service. When a file is added to SAGE, it checks the file type and if the file is an image, it scales it to a thumbnail and stores it in S3 with an URL that ties it back to the object. When that object is displayed, the thumbnail is retrieved rather than the original object itself. For non-image objects, this allowed a custom or generic thumbnail to be set.

When an object requests a thumbnail of a particular source with specific X and Y sizes measured in pixels, the thumbnail class attempts to determine if one matching those requirements has already been generated. If not, a new one is scheduled for creation and the thumbnail system will return an URL referring to a placeholder processing thumbnail.

Thumbnails have an X' and Y' size as well as their actual X and Y size. The X' and Y' size is used when a thumbnail of a specific size is needed, but to preserve aspect ratios either the actual X or actual Y coordinate will almost always be smaller than the X' or Y' size. The exception is when the actual X and Y sizes are in a perfect ratio to the requested sizes. For example, if I have an image that is 300x600 pixels in size and I want it to fit a 150x150 pixel box in the gallery, I would create a thumbnail that is 75x150 pixels in size to fit. The requested X' and Y' dimensions are both 150 pixels, but the actual X dimension is 75 pixels while the actual Y dimension is 150 pixels out of necessity to preserve the image's aspect ratio.

Thumbnails can be flipped. This is particularly useful when the scanning or photographic processes have resulted in landmarks or text, for example street signs or shop signs, from being represented in reverse.

Thumbnails also must preserve their original source and the location where the thumbnail is stored on Amazon S3 storage in order for it to be retrieved and accessed. Some URLs will be “local” for generic filetype-specific thumbnails. These thumbnail images are stored in the SAGE image assets rather than Amazon S3.

S3 can be accessed using API keys stored in the application’s hosting environment (i.e. on Heroku). These keys cannot be viewed by outside parties, and ensure that only SAGE has permission to modify the thumbnail images stored in S3. Whenever SAGE requests that a thumbnail image be committed to S3, it must also provide the API key.

3.7.4 Project Importing

When other systems are being used alongside SAGE, a simple data exchange format is very useful for saving a great deal of manual effort to transcribe data. SAGE allows data to be imported in the form of a CSV file, where the first column refers to an object URL and the second column refers to a concept description, and every pair implying an annotation.

The import feature is available in the project view along with the other administration commands. The CSV can be written in the provided text area, or provided as text via a copy and paste operation. When the CSV file is submitted, it is first parsed for errors by SAGE, and if any are encountered, the user is redirected back to the import page with the CSV file returned to the text input area and the error reported in a notice.

In the event that the CSV file was parsed without errors, the import process begins and SAGE processes the CSV file one line at a time. A typical line will be an [object, concept] pair.

When an object or concept is encountered that already exists, SAGE uses the existing entry. Otherwise, a new object or concept is created. Using the specified object and concept (whether they are new or existing), an annotation is then formed between the two. In this manner, an entire set of annotations can be imported into SAGE for an arbitrary number of [object, concept] pairs.

When a solitary object or a solitary concept needs to be imported, the line in the CSV file will simply be blank in the companion column. This allows [object,blank] and [blank,concept] lines to be provided for objects and concepts with no annotations. Of course, if that concept or object has already been referenced in an [object, concept] pair, there is no need to include it by itself, since it will have been already created if it doesn't yet exist in the database.

Project imports can be performed under three modes. The process detailed above is the most permissive mode, since any object or concept not already in the project will be created.

The second mode is a more restrictive mode that will not create new objects from lines in the CSV file. Annotations and concepts will only be created if the object they relate to is already in the project. This is analogous to a whitelist, where only items already on the list will be permitted, and those not on the list will be rejected. This mode is useful when the details exported from another program will unavoidably include unnecessary [object, concept] pairs that are not desirable as additions to the project database. This allows for opportunistic additions of new concepts and annotations from CSV files exported from other projects, since importing from the CSV file will only add relevant entries if the object exists in the project.

The final import type is a broader variant of the second mode, and seeks to match objects based on filenames rather than full URLs. If an object with the specified filename does not exist, then the object is not created. This allows files that have not been previously hosted online, which will consequently not have a full URL, to be matched with their counterparts in a storage service like Google Drive. This is particularly important when working with local files in a local application to create annotations (such as museum software like PastPerfect (PastPerfect Software, n.d.)), then seeking to import those annotations into a SAGE project that features those same files hosted in the cloud.

3.7.5 Authentication and Security

SAGE runs using a custom-built authentication and authorisation system. Users who register for SAGE can provide their login credentials to access their user profile, which has its ID number encrypted and stored in the user's session cookie. When a user attempts to access a resource, their profile ID is used to retrieve the roles they have been assigned in that project, allowing them to access any action which they are authorised to perform.

Every user has a role in any project that they are able to access. This role grants them a number of privileges within that project and also within any resources, such as objects or concepts, that are part of that project. Roles such as Administrator, Annotator and Contributor allow the user to make lasting changes to the projects, while Viewers may simply view what others have created.

The exact actions which the user is permitted to perform on each resource is defined within that resource's *controller*, which is the component of the routing system which performs any additional functionality required before the view is rendered. This includes responsibilities such as authentication and authorisation checks, along with other tasks such as preparing data to be displayed.

Each controller groups actions into categories depending on the level of privilege needed to perform that action. Simple display actions require only Viewer level of access, for instance. Higher levels of privilege completely subsume the privileges granted to lower levels, so an Annotator can always do anything that a Viewer can do, a Contributor can do anything that an Annotator can do, and an Administrator can do everything that the lower roles can do.

The user's role also affects the controls that are visible to them. This helps to prevent users with a lower level of access, such as Viewers, from selecting to perform actions that require a higher level of privilege, such as creating new annotations.

If the user manually enters the URL required to perform an action they are not permitted to perform, they will be presented with an error message. Similarly, users with no role in a project will be presented with an error message when attempting to perform any interaction with that project, and users who are not logged in will be redirected to the login page along with an error message asking them to log in before performing any actions.

3.8 Extensibility

Using the system outlined above, we can capture, organise and work with a range of heritage data. SAGE also encourages extensibility in suggestion mechanisms, analytics and evaluations. This allows several different suggestion mechanisms to be stored and used within SAGE, for statistics regarding different projects to be gleaned through project analysis, and for competing approaches to be compared against one another in terms of various metrics.

SAGE provides common analytics and five different evaluations by which suggestion mechanisms can be compared in terms of commonly used metrics based around a binary classification system of suggestions.

3.8.1 Suggestion Mechanisms

Suggestion mechanisms use the information captured by a software agent to find recommendations for new information that can be added by users, enriching the data stored in the system in a user-supervised positive feedback loop. Suggestions accepted by users provide new information upon which to base additional suggestions, allowing annotations to be contributed with very little effort on the part of the user. Suggestion mechanisms built for SAGE are not restricted to using an agent-based approach, although an agent-based approach works well with the way the data has been structured in this web application.

Consider the process of developing a custom suggestion mechanism. The simplest suggestion mechanism would be one that returns nothing. When viewing an object, all we would see is the control to create a new concept and annotate it to the object, with all existing annotations being unreported and effectively hidden.

An improvement on this simplistic approach would be to report what has been annotated to an object, but provide no further suggestions. This increases functionality significantly, as it means that we can view the information that has been contributed to objects, navigate to that concept, and also remove any erroneous concept annotations.

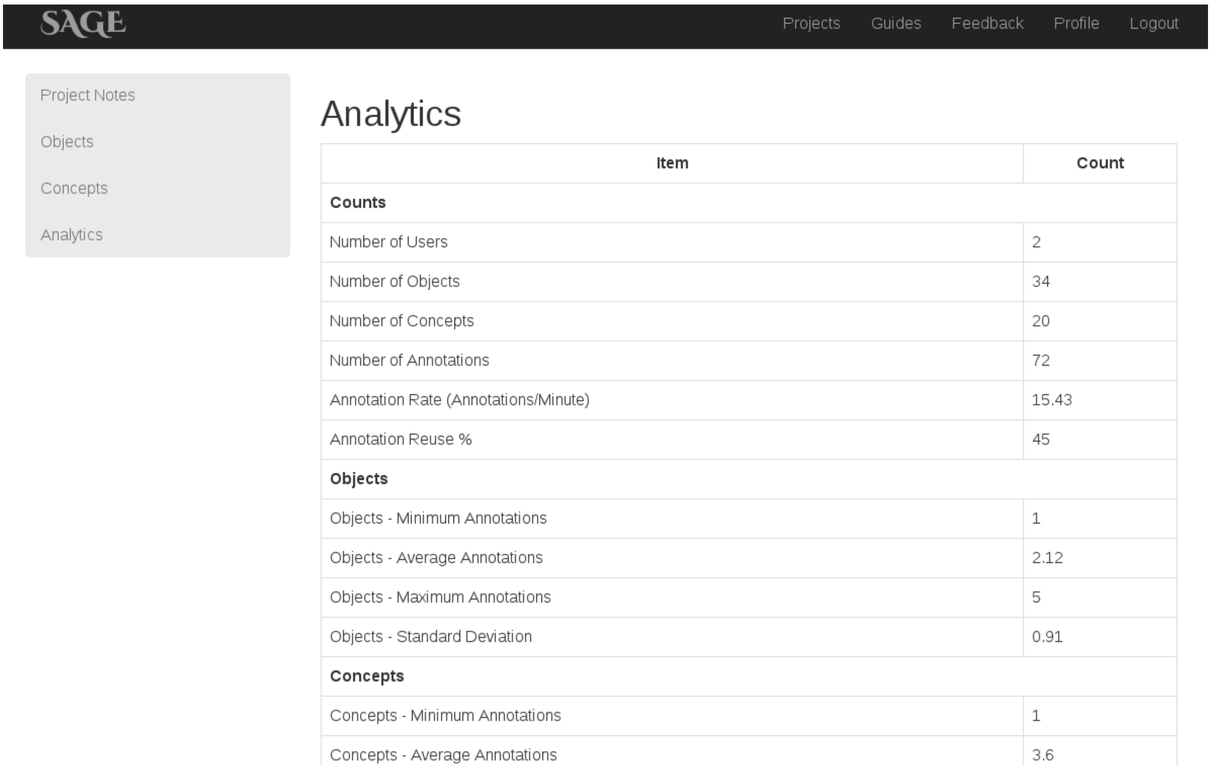
A further improvement would be to show all annotations and also show all other concepts in the database as suggestions, with the option to add the suggestions as annotations or remove annotations and return them to the pool of suggestions. This adds an additional level of functionality by promoting the re-use of existing annotations rather than creating new ones, which requires less effort to be expended and also implicitly provides more information in the form of concept co-occurrence and popularity when concepts are reused. The only real drawback of this approach is that when dealing with a large number of concepts, a large number of non-useful suggestions will be provided.

Finally, we reach the third improvement on these simple models, where the list of suggestions is sorted in a meaningful way, and where suggestions below a threshold of meaningfulness are possibly hidden. This is the target level of complexity for suggestion mechanisms employed and evaluated in SAGE, and is the principal subject of the following chapter in this thesis.

Each project can have a separate project-wide suggestion mechanism which is used by the agents contained within that project. This allows different algorithms to be used and tested in parallel, either using separate projects which may have different subject matters, or in duplicate, cloned projects where the only difference is the algorithm used to make suggestions.

3.8.2 Analytics

SAGE has a built-in analytics subsystem that provides information on the size and complexity of projects. This subsystem helps to describe the characteristics of the projects being used for evaluations and identifies quantitative differences between them (see Figure 3.16).


SAGE Projects Guides Feedback Profile Logout

Project Notes
Objects
Concepts
Analytics

Analytics

Item	Count
Counts	
Number of Users	2
Number of Objects	34
Number of Concepts	20
Number of Annotations	72
Annotation Rate (Annotations/Minute)	15.43
Annotation Reuse %	45
Objects	
Objects - Minimum Annotations	1
Objects - Average Annotations	2.12
Objects - Maximum Annotations	5
Objects - Standard Deviation	0.91
Concepts	
Concepts - Minimum Annotations	1
Concepts - Average Annotations	3.6

Figure 3.16 - SAGE Project Statistics Page

Most importantly, it provides information on the number of objects and the number of concepts, which helps to distinguish larger projects from smaller projects. It also provides information on the number of annotations between objects and concepts, which helps to distinguish more comprehensively annotated projects from superficially annotated projects.

3.8.3 Evaluation

SAGE has a range of built-in evaluation subsystems accessed through an interactive Ruby shell or through custom-built Rake tasks. These methods allow Ruby code to be entered and interpreted (in the case of the interactive Ruby shell) or for existing Ruby scripts to be executed (in the case of Rake) as a means of interacting with SAGE outside of the functional constraints imposed by the application’s web pages.

These evaluation subsystems are used for evaluating the effectiveness of suggestion algorithms and providing a richer analysis of projects than the project's analytics web page can provide. These systems are initially described here, but are explored in more detail in subsequent chapters of the thesis where they are used.

Performance Evaluation

This process creates a clone project based on an existing project and modifies it to have a designated level of partitioning between training and evaluation data. The performance evaluation compares the suggestions provided by a range of algorithms using a binary classification process that categorises suggestions made by that algorithm as true positives, true negatives, false positives and false negatives (Powers, 2011).

Using these classifications, the performance evaluation produces overall measures for each algorithm describing it in terms of Precision, Recall, F-Score (Powers, 2011) at three weightings for F-Score favouring or balancing precision and recall, Phi Coefficient (Chedzoy, 2006), Mean Reciprocal Rank, Precision@5 places and Success@5 places (Sigurbjörnsson & Van Zwol, 2008).

Acceptance Evaluation

This process examines the annotations that were provided by volunteers for sample projects created from a single existing project. The process contrasts the number of annotations that were suggestions accepted by users with the number of annotations that were created manually.

This allows a range of algorithms to be compared based on how frequently their suggestions were accepted in the first, second, third and fourth quartile of the annotation process as well as their overall acceptance rate. A suggestion mechanism that frequently produces good suggestions is highly desirable, especially if it can produce useful suggestions given limited initial information, such as during the early annotations in a new project.

Productivity Evaluation

The productivity evaluation uses the same samples created for the acceptance evaluation and investigates the quantity of annotations that were created per minute using the different algorithms. It accounts for time spent actively annotating and interruptions to the user's work detected by periods of low activity.

This allows algorithms to be compared based on differences to the rate of annotation, the quantity of annotations and connectedness of objects and concepts following these annotations. A suggestion mechanism that results in a large quantity of annotations while maintaining the level quality of those annotations is highly desirable.

Complexity Evaluation

Complexity evaluation examines the distributions of annotations in both concepts and digital objects. It identifies how comprehensively a project has been annotated and the level of internal connectedness between concepts and objects to quantify the complexity of each project's annotation data.

Subgraph Evaluation

In this evaluation, the links between objects and concepts are traversed to identify how many unique subgraphs have been formed by disconnected groups of concepts and objects. The subgraph evaluation also records the size of each subgraph to help identify if these are trivial subgraphs (e.g. a new object with a single annotation) or complex subgraphs (e.g. many objects and concepts that are separate from the main graph). This provides an indication of whether a purely graph-based method of moving between objects and concepts is likely to encounter significant difficulty in accessing the disconnected items.

3.9 Conclusion

The system outlined in this chapter, SAGE, is able to effectively capture and organise the data involved in a heritage dataset through using a purpose-built multiagent system, and many users are able to simultaneously view and contribute to this data using an easily accessible web interface. Through the use of suggestion mechanisms, we allow users to easily contribute new information to their projects as they use the system.

SAGE addresses the major challenges identified in this chapter, which are encountered when applying multimedia information retrieval techniques to digital heritage photo collections. Specifically, SAGE seeks to address the shortfall of annotations available for digitised heritage collections by providing an effective utility where these annotations can be captured and explored. SAGE is resilient to missing metadata and visual artefacts in the source material, achieved through employing a human perception-driven technique for producing annotations, and SAGE allows groups of researchers and volunteers interested in a given heritage project to collaborate securely through an online system available anywhere in the world.

SAGE's design allows several competing suggestion mechanisms to be compared with one another in terms of performance for the purpose of advancing research into effective approaches to the community memory building problem. In the subsequent chapters, we investigate the performance of several suggestion mechanisms and compare it to our own, evaluate the process of interaction with the user, and apply the complete SAGE system to a case study that has projects featuring several digital heritage photo collections.

Chapter 4: Semantically Annotated Graph Analysis (SAGA)

4.1 Overview

A key element to effective annotation of the items stored in digital repositories is a similarity mechanism that can be used to find other subjects that are similar to the subject that the user is viewing. Not only does it make navigating large collections easier, but it also assists in the process of contributing new knowledge to the repository by supporting users when they are annotating objects through suggestions.

The approaches that can be used to find similarity are dependent on the information available about each object in a collection. Content-based information retrieval is one approach, using the implicit presence of the object's content to find similarity (e.g. comparing two visually-similar images), while metadata-based information retrieval can be used when additional information has been provided as annotations or data fields on objects (e.g. comparing two images with the same photographer or location). Semantic annotations such as descriptions are particularly useful if present, as they can express the significance, content and context of an object, which might not have been captured by any other method.

A similarity measure is not automatically available whenever semantic annotations are used to document objects in a digital collection; semantic annotations are informative and useful to human viewers of a collection, but do not innately convey similarity to a computer algorithm beyond simple observations, such as when two objects share an identical annotation. An algorithm is needed to analyse annotations and detect by some means when two are similar before we can offer assistive features, such as providing suggestions to human annotators.

This chapter examines the information that a metadata-driven similarity mechanism has available to it when working with a digital collection, and then examines four methods that utilise this information in prior works (Sigurbjörnsson & Van Zwol, 2008). Several key challenges are identified that impact similarity measures in digital heritage applications, and a new method, Semantically Annotated Graph Analysis (SAGA), is presented to address these concerns.

The four methods plus SAGA are evaluated and compared using a large, open dataset to determine if the improvements in SAGA have led to improved theoretical performance compared to the existing methods. SAGA and one of the existing algorithms from prior works go on to be compared in practice during an evaluation of the SAGE environment in the following chapter.

4.2 Related Work

The key related works in this section are those that have explored a synergistic approach employing human and software agents, particularly those that investigate areas such as using semantic tags, quantifying the similarity between objects, ranking those objects, and using these results as a means of recommending suggestions for additional tags.

The appeal of using annotations as a means of determining when two images are similar is that these tags represent the high-level conceptual information present in an image. This means that using them allows conceptually similar objects to be retrieved, rather than visually similar objects. Three popular methods by which annotations can be used to detect similarity between two objects are tag co-occurrence, tag popularity and text similarity.

4.2.1 Tag Co-Occurrence

Tag co-occurrence, where a set of tags occurs in two or more objects and imply that the two objects share similar properties, is a fascinating metadata-based information retrieval and alternative to content-based information retrieval for detecting similarity between multimedia content (Sigurbjörnsson & Van Zwol, 2008).

When two multimedia objects share many of the same tags, it is likely that the two are conceptually similar to one another. Consider the tags “wooden”, “furniture” and “four legs”. These three tags are likely to co-occur in objects such as chairs and tables, which are closely related concepts.

Tag co-occurrence presents an effective way of using the information contributed by users to a collection (Sigurbjörnsson & Van Zwol, 2008). When objects share many tags, they create informal groupings. By examining the tags assigned to members of the group, the tags which are not already common can be proposed as suggestions.

Considering the previous example, if “hand-made” was added to one wooden furniture object, it would be a reasonable suggestion to make as an additional tag for the other wooden furniture objects. This propagates the new tag throughout the images, wherever it holds true. Following this, if a new object was created and its sole annotation was the “hand-made” tag, we could also predict that it might be “wooden” or “furniture”.

Tag co-occurrence is very flexible in nature. As new relationships emerge from the tag associations in images, suggestion mechanisms which use them can offer more complex and interesting recommendations. Consider if an object added to the example collection was “hand-made” and “furniture”, but also “metal”. We can now make suggestions for any hand-made furniture to identify if it is “wooden” or “metal”.

Tag co-occurrence is also robust, and all hierarchical relationships that form emerge from simple correlations in the tags rather than being rigidly specified. This allows tag co-occurrence to adapt to a huge range of topics, and also allows the co-occurrence structures to be instantly reorganised when tags change or are removed.

We can consider whether tags are generally popular within a dataset or are specific to a group of neighbouring objects and make recommendations accordingly. A careful strategy of promoting popular tags and suggesting informative tags that help to distinguish groups of neighbouring objects leads to a rapidly increasing number of tags in a network. Novel methods of suggestion that leverage tag co-occurrence as a source of information can then be applied to this rich collection of tags (Sun, Bhowmick & Chong, 2011).

4.2.2 Tag Popularity

When a concept is applicable to a wide variety of objects, it is likely to have been annotated to a larger number of objects than other concepts. For example, a broad topic such as "people" or "sky" is probably going to be more commonly annotated than a more specific topic such as "Albert Einstein" or "cumulous clouds".

By recommending more popular concepts as suggestions for objects, we have a greater than random chance of making correct suggestions. As an added benefit, this technique is not sensitive to the number of annotations an object has, allowing it to work effectively for new objects with no annotations. Once one or more popular tags have been added to establish a rough picture of what the object is about, other techniques can be used to suggest more specialised tags.

As an example, if 80% of the objects in a collection are annotated with the tag "furniture", with 20% annotated with "metal" and 40% annotated with "wooden", we would suggest "furniture", "wooden" and "metal" as first, second and third suggestions for a brand new, unknown object in order to maximise our chances of making a successful recommendation. As tags are annotated to the object, we can transfer weight from popular tag suggestions to tag co-occurrence suggestions to improve suggestion accuracy.

4.2.3 Word Co-Occurrence

Words are the basic building blocks by which meaning can be conveyed in written communication. Tag co-occurrence treats tags as a discrete entity, but since each tag is made up of words, we can analyse “word co-occurrence” within tags as an additional source of information with which to detect when two tags are similar in meaning.

To determine the similarity between two sets of keywords, Manning, Raghavan & Schütze (2008) explain a strategy for finding the cosine similarity score between two vector spaces, which provides a floating point value between zero and one indicating the extent of two spaces matching. These vector spaces can be formed from collections of keywords by counting the number of times each distinct keyword appears in the collection. Thus, two collections where the same words appear in roughly equal proportions will be detected as similar, while two collections with entirely different keywords will be recognised as being distinct.

Some approaches implement a stemming process such as suffix-stripping (Porter, 1980) or lemmatisation (Manning, Raghaven & Schütze, 2008) to improve the matching process. Suffix stripping removes the suffix applied to a word, such as “quickly” becoming “quick”, allowing both words to be detected as being related, but can result in poor performance for certain distinct words such as “wand” and “wanderer”, which would be incorrectly reduced to a single stem. Lemmatisation is a more complex approach that converts related terms into a single base term, which allows for robust detection such as recognising “better” and “good” as being related.

Word co-occurrence can be used in addition to the already described tag co-occurrence as a means of finding how closely two concept descriptions match one another. Word co-occurrence does not necessitate that a lemmatisation or stemming process be used; on one hand, additional computation time can be exchanged for robustness when faced with multiple lexical forms of the same base word, but on the other hand, not performing these processes offers the advantage of simplicity when this additional robustness is not needed, such as in the case of this thesis.

4.2.4 Neighbour Voting and Ranking

When considering the suggestions provided through tag co-occurrence, we may also want to quantify the weight of confidence in each suggestion and use that as the basis for ranking the suggestions in a list. A common approach is neighbour voting, where a suggestion's ranking is determined by the sum influence given to it by its neighbours. The identity of these neighbours along with the proportions of influence provided by each neighbour can be determined in many ways, such as by content similarity, annotation similarity or a combination of both (Truong, Sun, & Bhowmick, 2012).

The tags contributed to an image can encompass a diverse range of subjective and objective information. To improve the ability of algorithms to determine which tags are objectively present in an image, Li, Snoek & Worring (2009) devised a neighbour voting algorithm based on visual similarity. If many visually similar images have the same tag, then that tag is likely to relate to the visual content of the image. Conversely, if a tag does not appear in visually similar images, it is more likely to be a subjective tag. This provides an important method for software algorithms to interpret the tags that users have annotated to images.

Tags on social photo sharing websites such as Flickr are presented without explicit ordering or ranking, as each tag assigned to an image by users has equal intrinsic weight. A technique was devised by Liu, Hua, Yang, Wang & Zhang (2009) which allows these tags to be ordered by first estimating the initial relevance of tags to an image, then incrementally refining their weights using a random walk process to find the true weights and ranking. This provides another important bridge between the manual tagging efforts of users and the ability of software algorithms to estimate the importance of each tag.

Both of these examples demonstrate that by analysing the distribution of tags across images in a collection through the process of neighbour voting, we can uncover insights which were not immediately obvious when looking at individual images, and have a level of semantic complexity which we would normally expect from the product of human perception and reasoning.

4.2.5 Evaluation Strategies

While a range of techniques for detecting similarity between tags and objects have been identified, we still require a means of determining when one approach or combination of approaches outperforms another in terms of suggestion accuracy. We explore how rankings produced by similarity measures can be validated using automated evaluation techniques which determine their accuracy. A standardised strategy (and ideally a standardised data set as well) allow approaches described by different authors to be compared with one another in a fair and consistent manner.

The evaluation of this PhD project follows a similar strategy as described by Shani & Gunawardana (2011). We begin by evaluating the algorithm's performance in an offline setting, then evaluate the application's performance with a small group of users, and finally consider the application in context of a case study in digital heritage. Shani & Gunawardana promote this approach as a better indicator of the algorithm's usefulness than measuring offline suggestion performance alone.

In a survey on evaluation methods (Herlocker, Konstan, Terveen & Riedl, 2004), choices of dataset and both performance and non-performance related metrics were identified as being core issues when evaluating recommender systems. Essential to designing experiments that can be compared to one another is establishing a shared set of data, truths, metrics and methodologies to produce the results (Hare & Lewis 2010). The MIRFlickr25K dataset was chosen in this PhD project because it provides both data and truths, and is highly suitable as it is both openly available and allows a wide range of content and metadata approaches to be used for suggestions (Huiskes & Lew, 2008).

McParlane, Moshfeghi & Jose (2014) discuss a number of issues with tag recommendation evaluation datasets used in other research in their paper. Issues that were highlighted include ambiguity and the lack of normalisation in collections, the lack or misuse of ground truth, poor image quality, low subject diversity and copyright restrictions. While synthesising a dataset aimed at minimising these issues addresses these concerns, new issues are introduced by the synthetic nature of the collection. For example, should a recommendation technique actually be tolerant of noisy, absent or ambiguous tags provided by users? This presents a case for using different types of test collections to account for the conflicting factors.

4.3 Challenges

Approaches which use tag co-occurrence and other annotation-based methods for detecting similarity in heritage applications encounter two challenges which are common for recommender systems, namely the issue of cold start and a long tail in result lists.

4.3.1 Cold Start

A common question in recommender systems is how to address the issue of cold start, where suggestions cannot be made by the system as the system contains insufficient information to be able to make an informed estimation on what may be relevant

(Schein, Popescul, Ungar & Pennock, 2002). This is also highly applicable to digital heritage applications, and applies to both the project and digital object levels.

At the project level, the cold start problem is encountered for every new project that is started. Each heritage project's subject material is unique, so to use inter-project information to address the cold start is complicated due to uncertainty over relevance. Identifying similar projects for the purpose of using their concepts as initial suggestions either explicitly, or implicitly by analysing information such as the project's description, raises issues of privacy in personal, unshared projects, as concepts could include sensitive or identifying information.

At the object level, an object with no annotations is unconnected to the project's annotation graph, the network created by following the annotation links between objects and concepts. As nothing is known about the object, the ability to accurately predict relevant concepts is limited. Being able to make suggestions at an early stage in heritage projects requires annotators to first identify a number of concepts which are relevant to the objects and project being examined. This slows the initial annotation process down.

4.3.2 Long Tail

The other common question in recommender systems is how to address a long tail in suggestion results (Yin, Cui, Li, Yao & Chen, 2012). It is quite usual for a suggestion mechanism to find a few, extremely high-scoring suggestions and a longer list of modestly-scoring suggestions. The high scoring suggestions are shown as recommendations as well as some of the modestly-scoring suggestions, but this raises the question of where the cut-off threshold should be, as a result which is not shown as a suggestion is effectively being treated the same way as one which has been detected as not relevant.

A long tail is undesirable in digital heritage applications that rely on participants for annotation as annotators expend a small amount of time and effort in considering each suggestion. This is likely to return a good quantity of hits towards the start of the list where the suggestions are highly relevant, but asking a volunteer to consider many low-scoring suggestions in the hope of occasionally finding a hit towards the end of the list is not an optimal strategy, as the annotator's time may be better spent considering the set of suggestions available for the next object. As a project grows with many objects, concepts and annotations being introduced, the size of these lists can continually grow, resulting in additional time being wastefully spent on these low-relevance suggestions instead of pursuing the "low-hanging fruit" elsewhere in the collection.

The other consideration regarding the long tail problem is that every annotation provides new information with which new suggestions can be calculated. It is expensive to compute the similarity scores of every object to every concept and thus identify the optimal areas to direct users, and this issue is compounded by having to re-calculate all similarity scores whenever a suggestion is accepted, resulting in constantly changing scores as one or multiple users collaboratively work on a project.

Projects can contain mutually-exclusive concepts, a simple example being the album that a digital object was sourced from. If an object was created using an image from album 14, then we know that the object isn't from album 16, 28, 34 or 40 (for instance). However, if these albums share many common characteristics, it is very likely that they will be recommended as suggested concepts, which doesn't negatively impact other suggestions, but may be inconvenient for the user. Establishing complex rules of mutual exclusions to eliminate these suggestions would divert the user's attention away from the annotation task, and while it can gain some performance improvements in retrieval, it loses out on the benefits provided by the simplicity of the original approach (Deng *et al.*, 2014).

4.4 The SAGA Algorithm

Existing similarity measures suffer from a number of shortcomings. These include sensitivity to the cold start problem, where few suggestions are available for new collections; long tail in suggestion results, where a filtering method for low-likelihood suggestions is absent; requirements for training data before the algorithm behaves optimally; and discrete graph bridging issues, where one user's work is separated from another user's work until a bridging tag is added (Bonchev & Buck, 2005).

The SAGA (Semantically Annotated Graph Analysis) algorithm proposed in this research overcomes these issues through a novel design. Concisely put, SAGA determines that neighbouring images are indicated by the co-occurrence of similar semantic tags, and the tags which neighbouring images possess can be provided as suggestions for other neighbours.

The approach encourages early, speculative connections to form between concepts, and between objects through them. Users may accept the relevant suggestions among those returned by the algorithm to enrich the information that has already been captured in the network, which in turn helps to discover additional relevant annotations. This process is explored in detail in this section.

4.4.1 Object Suggestions

When starting with an object with the goal of finding additional concepts to associate it with, we first consider the information available to that object, which acts as a software agent with knowledge limited to itself and its immediate neighbours. The object knows which concepts it has been annotated with, and also that all concept annotations are, as far as it is aware, equally relevant.

The object is able to query each concept and pose each of them a question; "which concepts are similar to you?" In this way, the SAGA suggestion mechanism uses similarity between concepts to find suggestions for new annotations.

The object also passes an evenly divided quantity of influence points to each of its concepts. This voting influence will be divided between the suggestions made by each concept in a proportion that best represents that concept's confidence in each suggestion. The initial amount of influence points is proportional to the number of concepts in the project, allowing for the possibility that every concept in the project could be an equally relevant suggestion.

Ranking the final set of suggestions is then a process of identifying each unique suggestion returned by the queries made to each of the object's associated concepts and tallying the number of points given to the suggestions, as a number of different concepts may have returned the same suggestion. A quantity of points accumulated during the process are also distributed to concepts that are popular in the project, boosting those results.

At the end of the influence aggregation process, results which have not met the minimum threshold are removed from the set of suggestions (though the object’s annotations are never removed from the results list, even in the unusual case that they score poorly). The ordered list ranked on points of influence is the final result.

This process can be represented in the following pseudocode. Note that references such as “(2a)” indicate a call to another function, with the suffix of “a” reserved for object functions and the suffix of “b” for concept functions.

(1a) Initiating SAGA for an Object

Define Influence as the count of Concepts in the current Project.

Define Propagations as 3.

Find Results using Self as the starting point (Influence, Propagations) (2a).

Replace "Popular" in Results with Popular_Concepts (3b).

Aggregate Results, eliminating duplicate entries and accumulating scores.

Filter Results of scores < 1.0 that aren't annotations.

Sort Results by score.

Return Results.

The value of propagations is initially set to 3 to specify how "far away" the termination point of this operation should be. A value of 0 would reach the initial object, while a value of 1 would reach that object's concepts. These two pieces of information are already directly available to the object, so are not particularly useful. A value of 2 would reach the initial object's neighbours (those objects having one or more shared concepts with the initial object), while the value of 3 reaches all concepts possessed by the object's neighbours, a set of concepts which includes the concepts already linked with the initial object, and hopefully a subset which have not yet been linked to the initial object which can be presented as suggestions.

If the number of propagations is set to a higher odd number, such as 5, concepts will still be returned, but these will be less closely related to the initial object. Similarly, higher even numbers, such as 4, will find "distant neighbours", which are objects relating to the initial object in an increasingly indirect manner. The use of the SAGA algorithm to find these sets has not been pursued for the purposes of this research, but may be useful in a different application.

The process described in (1a) is followed for the object asking for suggestions, but what if an object is asked to provide suggestions? If an object receives a suggestion request, the object will respond in a manner determined by the information it receives as part of that request, which includes an amount of influence points as well as an indication on whether the object is the termination point of the search for suggestions, or is an intermediary agent in the process, with more propagations of the influence points yet to come.

If the object is the termination point, it will greedily assign all influence to itself and return that as its response. If the object is an intermediary, it will distribute influence evenly between its concepts and query them in turn for their suggestions, while holding back a single equal portion of influence which it assigns to “Popular”. This is illustrated in Figure 4.1.

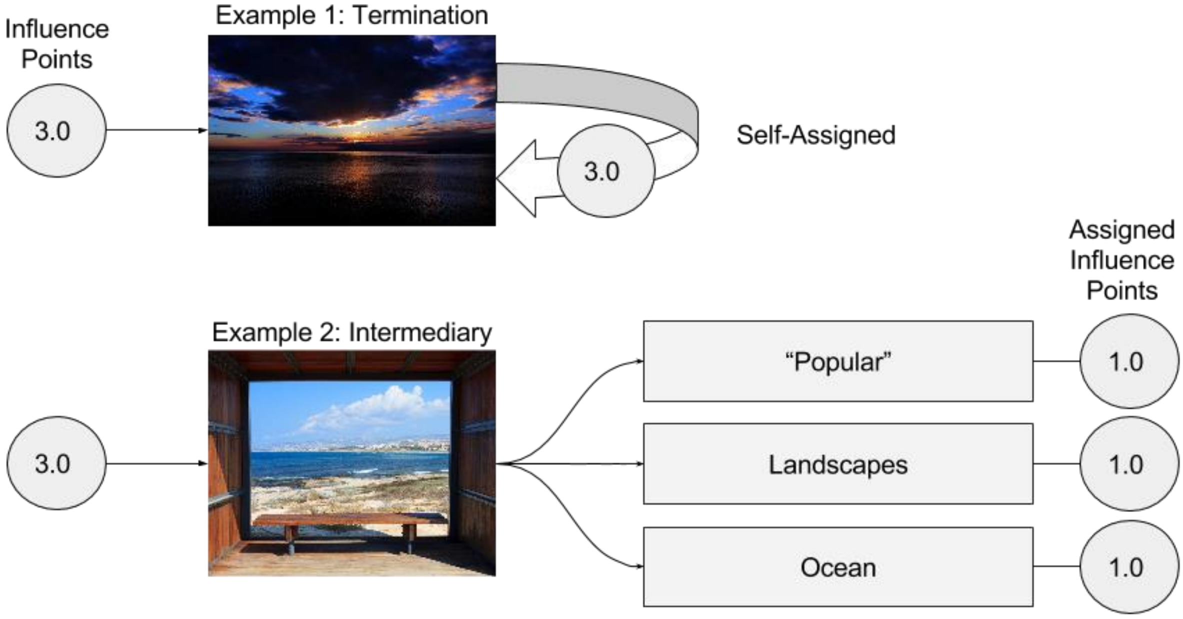


Figure 4.1 - Object Influence Distribution

Considering this method of influence allocation, if the object has a large number of concepts annotated to it, then the amount of influence given to “Popular” will be very small. The object is well-connected to the semantic network and does not need to rely on what is popular to make a meaningful suggestion. Conversely, if the object has few annotations, a relatively large amount of influence will be given to “Popular”, allowing a poorly-connected object to still make a best-guess as to what a set of good suggestions will be.

A critical case to consider is when an object has no annotations whatsoever, which will occur when a new object queries itself for suggestions as part of the algorithm shown in (1a). In this case, 100% of the influence it has will be given to "Popular", as this is the best suggestion that it can make while being disconnected from the main semantic network.

This process can be represented in the pseudocode:

(2a) SAGA for Objects
Receive Influence and Propagations as arguments.
If Propagations == 0:
    Define Results as { Self => Influence }.
Otherwise:
    Decrease Propagations.
    Define Proportion as Influence / (Annotations + 1).
    Define Results as { "Popular" => Proportion }.
    For Each Annotation:
        Add SAGA Response for that Annotation to Results,
        passing Proportion and Propagations as arguments.
    End.
End.
Return Results.

4.4.2 Concept Suggestions

When starting with a concept with the goal of finding additional objects to associate it with, an almost identical process is followed as for objects. The main difference is that the amount of influence initially available will be proportional to the number of objects, rather than concepts.

This process can be represented in the pseudocode:

(1b) Initiating SAGA for a Concept
Define Influence as the count of Objects in the current Project.
Define Propagations as 3.
Find suggestions using Self as the starting point (Influence, Propagations) (2b)
Replace "Popular" in Results with Popular_Objects (3a).
Aggregate Results, eliminating duplicate entries and accumulating scores.
Filter Results of scores < 1.0 that aren't annotations.
Sort Results by score.
Return Results.

Concepts have an additional source of information that can be used to find suggestions: the text that makes up their description. This allows a text similarity measure to find lexically similar concepts as an additional step in allocating influence points in concepts. The benefit of doing so is that it creates speculative links across the project that can link the work being conducted by one user with the work being conducted by another. This encourages them to annotate objects with concepts created by the other, boosting the effectiveness of tag co-occurrence strategies by creating highly-interconnected semantic networks.

This additional step occurs right after the concept receives influence from another agent. The concept will find all concepts in the project that have a degree of text similarity (including itself, which will be a 100% match) and distribute influence to each concept accordingly. Consider an example, where three concepts have text similarity. One will be 100% (itself), one might be 20%, and one might be 10%. The vast majority of influence will still go to itself, but a small amount of influence branches out to those speculative concepts which might or might not be relevant. If another concept has a high matching score (for example, 90%), then it is highly likely that the two concepts are quite similar, and might even be good candidates to merged together.

After sharing influence, the concept will operate in a similar manner to objects, and the concept allocates an even portion of influence to each of the objects it has been annotated to (if it is intermediary agent), or it will greedily allocate all of the influence to itself if it is the termination point.

This process can be represented in the pseudocode:

> *(2b) SAGA for Concepts*
> *Receive Influence, Propagations and possibly Dispersed as arguments.*
> *If Dispersed is not received:*
>   *Disperse Influence among all concepts proportional to text similarity with Self (Proportional Influence, Propagations and Dispersed given).*
> *Otherwise:*
>   *If Propagations == 0:*
>   *Define Results as { Self => Influence }.*
>   *Otherwise:*
>   *Decrease Propagations.*
>   *Define Proportion as Influence / (Annotations + 1).*
>   *Define Results as { "Popular" => Proportion }.*
>   *For Each Annotation:*
>   *Add SAGA Response for that Annotation to Results, passing Proportion and Propagations as arguments.*
>   *End.*
>   *End.*
>   *Return Results.*
> *End.*

Bringing it all together, the suggestion process for an object will follow the process outlined in Figure 4.2.

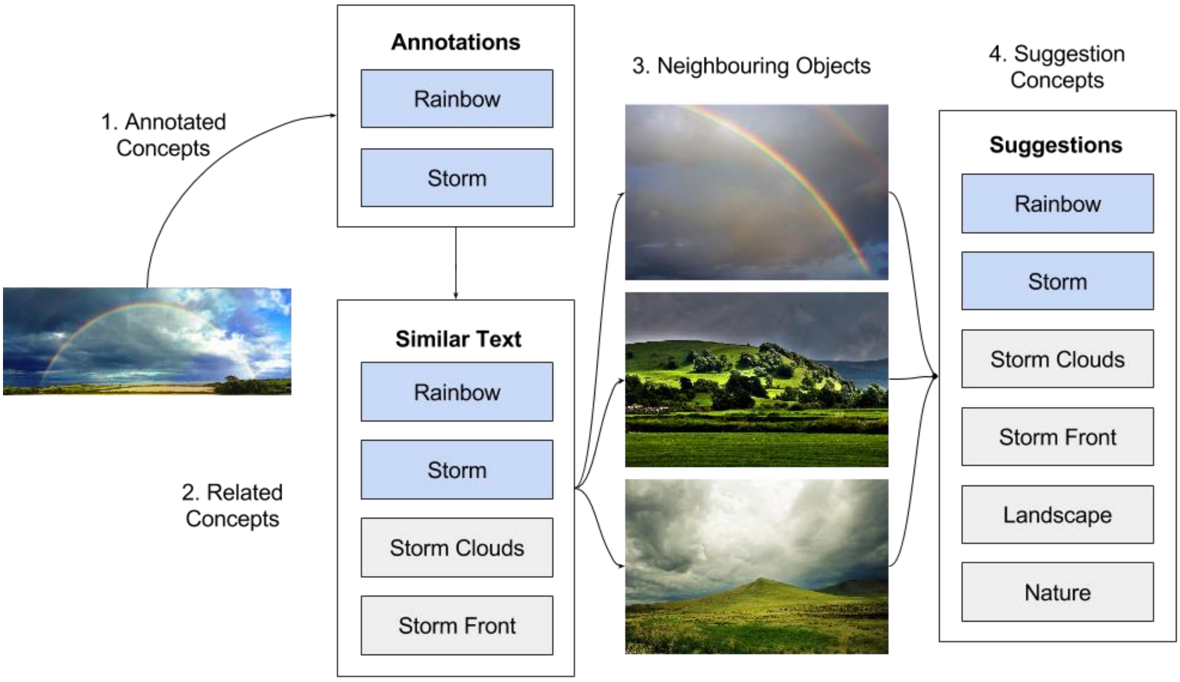


Figure 4.2 - SAGA Suggestion Discovery

First, influence will be passed to the concepts that have been annotated to the initial object. Second, influence will be reallocated to all concepts that have similar text to these concepts. Third, influence will be passed to the neighbouring objects associated with these concepts (called "neighbours" as they share similar concepts with the initial object). Finally, influence will be divided between all of the neighbouring object's concepts, and an aggregated results list will be obtained from all neighbours. The results are then ranked in order of strongest to weakest suggestion.

4.4.3 Weights

Weights are assigned equally when working with a list of associated concepts, since no concept is more relevant than another in this context. Consequently, popular/general concepts that provide many suggestions contribute only a small amount of voting weight to each suggestion, while uncommon/specific concepts consolidate their voting weight into a small number of high-quality suggestions. This is desirable since the suggestions from a popular concept are likely to be a weaker indicator of semantic relevance than suggestions from an uncommon concept with only a few suggestions.

Contrasting with the above, weights are assigned proportionally when using a text similarity measure, since a quantified measure of relevance is available in this context. This is the case when finding lexically similar concepts, since they will have a text similarity score represented as a value between 1.0 (exactly the same) and 0.0 (with no similarity). If one concept is a 60% match (0.6) and a second concept is a 40% match (0.4) then 1.5 times the influence will be given to the first concept as to the second (see Figure 4.3).

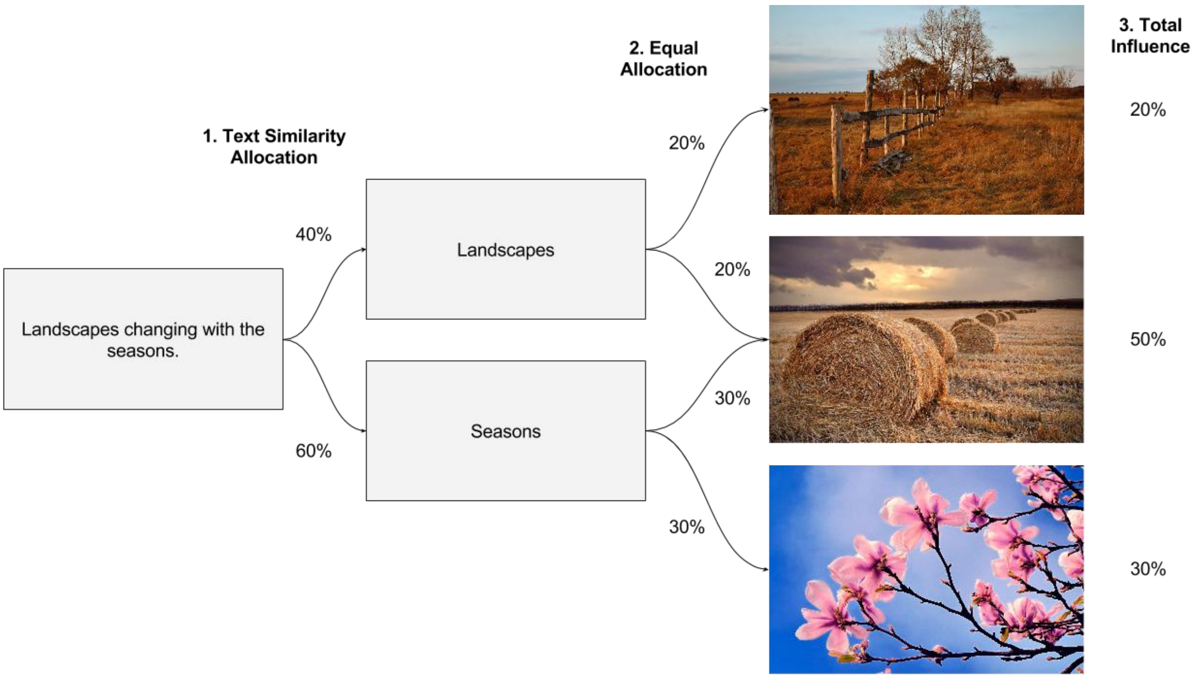


Figure 4.3 - Concept Influence Distribution

In Figure 4.3, we move from a user's query concept to two concepts with a 60% and 40% text similarity score in step 1. As one concept matches more strongly than the other, it receives a larger percentage of influence points. In step 2, the influence each concept received is distributed equally to the objects it has been annotated to. Since the middle object was associated with both concepts, it receives a share from each concept, and therefore becomes the overall strongest suggestion as shown in step 3.

4.4.4 Popularity

Popularity is calculated by project agents and provided as a service to the objects and concepts that request it. The popularity of an object or concept is equal to the number of times it has been annotated in the project, with a single additional point given to every object or concept in the project.

In SAGA, for both objects and concepts, a certain number of influence points will be allocated to "Popular", a placeholder for the popular objects/concepts in that project. These influence points will be distributed in proportion to the popularity of each object/concept. For example, if we have three objects {A => 3.0, B => 6.0, C => 1.0}, and 20.0 points of influence allocated to "Popular", A will have 6.0 points, B will have 12.0 points, and C will have 2.0 points after allocation.

The single additional point given to objects and concepts means that objects and concepts that have not been used in any annotations are still represented on the list of popular objects or concepts. This satisfies the special case where there is one new object in a project (for example), and ten concepts, with no annotations having been made. If the object is used as part of the SAGA algorithm, it will assign 10.0 influence points to "Popular" (as it has no concepts to give influence to), and these 10.0 points will then be distributed to the ten concepts, with an end result of each of the ten concept being equally relevant with 1.0 point.

This process can be represented in the pseudocode:

(3a) Finding Popular Objects
Receive Influence as argument.
Define Results as the map of all Objects to their Annotation Count + 1.
Allocate Influence to Results based on each Result's score.
Return Results.

(3b) Finding Popular Concepts
Receive Influence as argument.
Define Results as the map of all Concepts to their Annotation Count + 1.
Allocate Influence to Results based on each Result's score.
Return Results.

4.4.5 Filtering

Filtering is the final step before displaying a list of results as suggestions, and is a strategy employed in SAGA that helps to avoid a long tail in results. Individual results that have obtained less than 1.0 point of influence will be removed. This helps to keep the results list concise, particularly as a counterbalance to the many low scores that are introduced through the popularity component of SAGA.

Filtering values below 1.0 was chosen with consideration of the case when no annotations have been created. This can be illustrated when finding suggestions for an object. If there are *n* concepts, the total amount of influence available is *n*, and the influence will be evenly distributed between the identically-likely concepts. Thus each concept receives 1.0 point of influence. All concepts will be presented as suggestions and no filtering will occur, since all concepts are detected as equally likely by the SAGA algorithm.

Once some annotations have been made, a group of these concepts will emerge that are slightly more likely as suggestions than others. However, the total amount of influence available in the SAGA algorithm will be unchanged. Therefore, the more likely suggestions will receive more than a single point of influence, and less likely concepts will be allocated less than a single point of influence. Filtering will occur to remove the less likely concepts from the list of suggestions.

As the actual value of influence received by each suggestion is dependent on the number of concepts in a project, the influence received by a suggestion cannot be directly compared to the influence received by a suggestion in a different project. Consider a concept with 2.0 points of influence, and a concept in a different project with 80.0 points of influence. If the first project only contained five concepts, while the second contained one million concepts, the first concept is actually proportionately more likely to be relevant as a suggestion than the second. For this reason, filtering by arbitrary values of influence is not provided as a configuration variable in the SAGA algorithm.

4.5 Methodology

In the first set of experiments, we focus on evaluating the suggestion quality provided by the SAGA algorithm in a series of offline experiments that are independent of SAGA's usual user environment, SAGE, and therefore are independent of user interaction.

As a benchmark, we use the Vote, VotePlus, Sum and SumPlus algorithms, a set of well-studied general purpose suggestion mechanisms that uses tag co-occurrence to make recommendations (Sigurbjörnsson & Van Zwol, 2008). The aim of these experiments is to determine whether the domain-specific enhancements used in SAGA provide improved performance over the current state-of-the-art algorithms.

A positive result in these series of experiments would provide supporting evidence for a number of research outcomes, specifically:

- That the SAGA algorithm provides suggestions with a high level of quality, comparable to the current state-of-the-art.
- That the SAGA algorithm outperforms the state-of-the-art on the MIRFlickr25K open dataset, and that;
- The issues identified and accounted for have led to this improvement.

4.5.1 Algorithms

The closest algorithm in current literature to SAGA is the algorithm referred to as VotePlus (Sigurbjörnsson & Van Zwol, 2008), the best performing algorithm from a series of tests that also resulted in the creation of the Sum, SumPlus and Vote algorithms. VotePlus has been used in a range of application areas including assistive tagging (Wang, Ni, Hua & Chua, 2012) and event detection in social media (Firan, Georgescu, Nejdl & Paiu, 2010). However, digital heritage is an area where it has not yet been applied.

SAGA is readily comparable with the Vote, VotePlus, Sum and SumPlus algorithms. All algorithms use tag co-occurrence as the primary means of making suggestions, and SAGA, VotePlus and SumPlus use global statistics such as a concept's popularity as a means of modifying the results to present the strongest candidates first.

Vote and Sum are two basic techniques that employ tag co-occurrence to find suggestions, with the two differing in the voting strategies they use. While the Vote algorithm scores suggestions based simply on the number of times they co-occur in concepts shared with the initial object, the Sum algorithm also normalises each suggestion based on the number of times it has been used in annotations.

VotePlus and SumPlus seek to boost the best suggestions found by the Vote and Sum algorithms respectively by promoting suggestions that rank well in the Vote/Sum algorithms, that do not occur rarely (and are unlikely to be useful), and that do not occur too frequently (and are too general to be useful).

The VotePlus and SumPlus algorithms also use the top 25 co-occurring tags to find suggestions, rather than 10 as in the Vote and Sum algorithms, which results in the VotePlus and SumPlus algorithms typically retrieving a broader list of suggestions than Vote and Sum.

SAGA's primary differences from these four algorithms are that it draws in more speculative information when making similarity judgements, and that it automatically adapts to filter low-confidence results without requiring any training or configuration variables to be set.

SAGA uses global statistics to make suggestions for new images, where the lack of prior information would typically make it difficult for an algorithm to provide suggestions (the Cold Start problem). This problem is common in sparsely-annotated heritage projects, which are common due to the volume of information being annotated and the small number of experts and volunteers who are available to provide these annotations.

SAGA uses text similarity to discover concepts that have similar descriptions, and will automatically follow these links when calculating suggestions to assist with linking work being performed independently by separate researchers. This also allows images that have been annotated in different styles (tags, semantic tags, individual descriptions, etc.) to be used together within SAGE projects, potentially allowing annotations produced by several external sources to be combined in SAGE.

SAGA employs an agent-based method for distributing weights, and provides a filtering mechanism to remove low-weight suggestions from the list of suggestions returned to the user. This helps to cut back the number of suggestions that users are asked to consider (the Long Tail problem), and controls the workload being assigned to participants in projects where the quantity of manpower is limited.

SAGA does not use configuration variables that must be optimised before use. This makes it simpler to implement than an algorithm that relies on training, and allows the same configuration of the algorithm to be used on varied datasets which would otherwise require re-training. As heritage projects are often quite small, the opportunity to train algorithms using large quantities of data is not present, so this is a desirable advantage.

4.5.2 MIRFlickr25K Dataset

The largest openly-available image dataset with relevant annotations is the MIRFlickr25K dataset (Huiskes & Lew, 2008). This collection consists of 25,000 images sourced from Flickr, and the annotation data for the 24 most common tags that have been applied to these images. Together, these pieces of data allow a complex and complete network to be constructed. Table 4.1 describes the high level statistics of the MIRFlickr25K collections.

Table 4.1 - MIRFlickr25K Statistics

MIRFlickr25K Statistics	
Objects	25000
Concepts	24
Annotations	92902
Average Object Annotations	3.72 (0-14, σ: 2.0)
Average Concept Annotations	3870.92 (259-10373, σ: 3124.41)

Some examples of images in the MIRFlickr25K collections are illustrated in Figure 4.4. This images were sourced from the MIRFlickr website.

Figure 4.4 - MIRFlickr Example Images

The images specified in the MIRFlickr25K collections are identified by ID number rather than an image file itself. While the images can be downloaded separately, as neither SAGA nor the VotePlus algorithm use image content data to make suggestions, the ID numbers themselves were used to populate a new project with 25,000 text objects. Using the ID numbers, objects were linked to the 24 concepts according to the annotations specified in the MIRFlickr25K data set (refer to Figure 4.5).

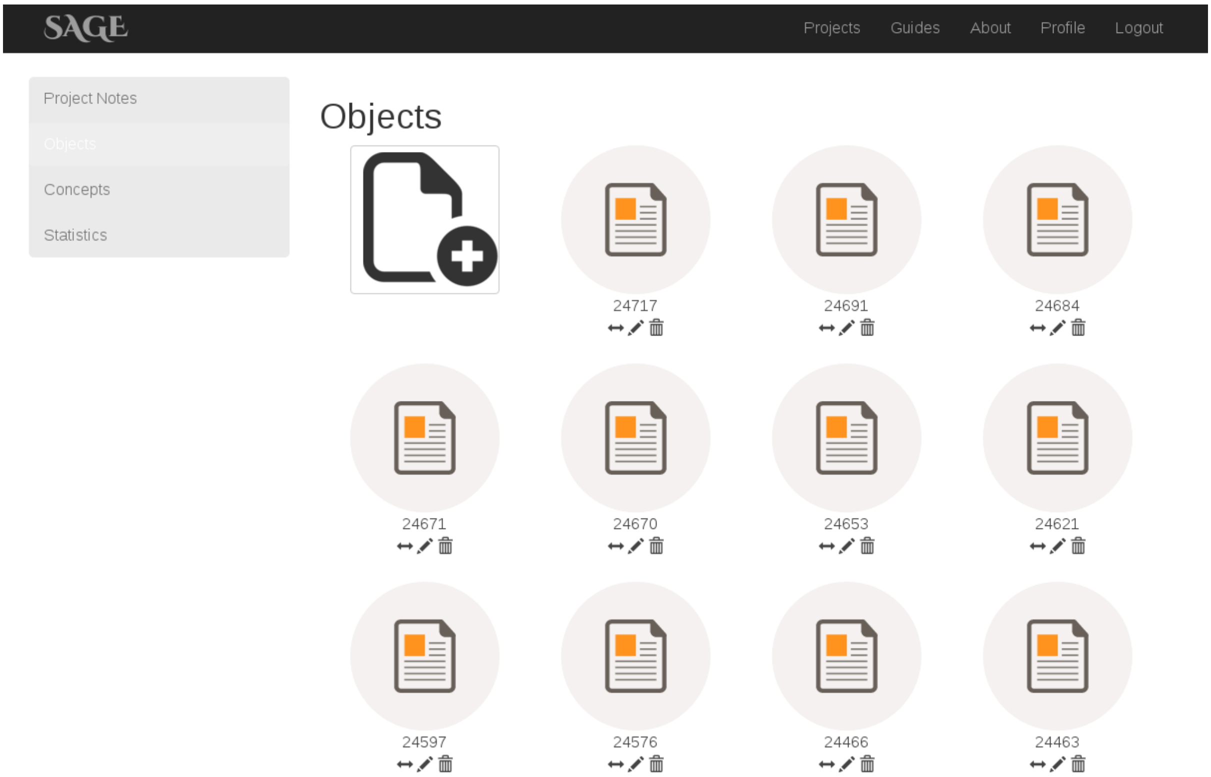


Figure 4.5 - Text Representations of Objects in MIRFlickr25K

While not a heritage dataset, the MIRFlickr25K collection allows SAGA to be compared with VotePlus on a general-purpose collection which has been widely used in recommender systems evaluation, such as in ImageCLEF 2009 (ImageCLEF, n.d.) for fully-automated visual concept detection and annotation.

4.5.3 Performance Evaluations

SAGA and the Vote, VotePlus, Sum and Sumplus algorithms all rely on the annotation information present in a project to make suggestions. None of the algorithms are hard-coded to detect certain objects or events, and no algorithm is capable of creating a new concept without human intervention. Therefore, the MIRFlickr25K collection needs to be partitioned, with a certain proportion to be used as a training set, and the remainder to be used as a validation set to check the accuracy of suggestions.

The proportion used for training has an effect on the ability of the algorithms to make suggestions. A small training set would simulate a newly created collection, while a large training set would simulate a well-annotated collection. To explore this variable, the MIRFlickr25K was partitioned at 20%, 40%, 60% and 80% in four scenarios. A partitioning level of 40%, for instance, means that 40% of the original annotations were preserved for the training set, and the remaining 60% would be for validation.

Each scenario contains annotations that are selected randomly and independently of the other scenarios. To ensure that all algorithms were given equal footing, all comparisons between algorithms were performed using identical scenarios (i.e. training sets) on identical target objects, with suggestions produced for 30 randomly-selected target objects at each partitioning level and averaged to give an overall measure. The sample size is large enough to account for variability in individual results, and also avoids the need for cross validation, as suggestions are made on a per-object basis, rather than providing a population-wide set of suggestions that would need to be investigated for overall fit.

As no algorithm in this experiment employs a training process that would change its configuration to optimise its performance, overfitting, where an algorithm becomes increasingly optimised for a particular dataset over many runs, giving an inflated estimation of its discretionary power when making suggestions for that validation dataset, was not an issue.

The suggestions provided by the algorithms were analysed using a binary classifier, which categorised them as either true or false depending on the validation set. This provided four measures; true positives (hits), false positives (type I errors), true negatives, and false negatives (type II errors, a.k.a. misses).

These measures can be used to calculate a variety of measures commonly used to represent information retrieval system performance, including precision, recall, F-score, phi coefficient and two measures used in the original VotePlus paper, success@n and precision@n (Sigurbjörnsson & Van Zwol, 2008). The mean reciprocal ranks were also calculated by inspecting the positions that correct suggestions occupied in the overall list of suggestions.

Precision and Recall are two common information retrieval measures. Precision refers to the proportion of the suggestions made by an algorithm that were actually relevant, while recall refers to the proportion of relevant suggestions the algorithm detected and made as suggestions (Powers, 2011).

F0.5, F1 and F2 are F-scores, a harmonic mean of precision and recall, with the number component representing the weight favouring either component, with F1 being a balanced score between the two. It is commonly used as a means of simplifying precision and recall into a single measurement (Powers, 2011).

The Phi Coefficient (also known as the Mean Square Contingency Coefficient or Matthew's Correlation Coefficient, and is similar to Pearson's Correlation Coefficient) is a measurement that accounts for differences between predicted and actual binary values, such as those produced using a binary classifier on suggestions (Chedzoy, 2006). It has the advantage over F-scores in that the phi coefficient accounts for the likelihood of randomly guessing the correct answer. A phi coefficient of 1.0 is perfect prediction, 0 is no better than random selection, and -1.0 is complete disagreement with the truth.

In the original paper that introduced the VotePlus algorithm (Sigurbjörnsson & Van Zwol, 2008), the Precision@N and Success@N measurements were used to determine the precision within the top N results and the chance of the top N results including at least one correct suggestion respectively. Mean Reciprocal Rank (MRR) was the other measure used in the paper, which is the inverse average rank at which the first correct suggestion is made (e.g. a MRR of 1 suggests that the very first suggestion was correct in every sample, ½ suggests that the second suggestion on average was correct, etc.).

4.6 Results

The following results were obtained for the Vote, VotePlus, Sum, SumPlus and SAGA algorithms using the MIRFlickr25K dataset. Evaluations were run at 20%, 40%, 60% and 80% partitioning between the training set and the test set (e.g. at 60%, the majority of annotations in the dataset were retained and the remaining 40% was reserved for validating suggestions).

Recall, Precision, Phi Coefficient, Precision@5 and MRR were selected as the measures most indicative of the algorithm's performance and were reported here, along with discussion regarding F-score and Success@N. The complete results can be viewed in Appendix A.

4.6.1 Recall

All algorithms scored extremely well on the well-connected MIRFlickr25K collection, with recall improving from 93% through to 100% for all five algorithms as the amount of training material was increased. This result is encouraging, but not particularly surprising, as the MIRFlickr25K collection provides a large number of annotations for algorithms to analyse even at lower training proportions (refer to Table 4.2).

Table 4.2 - Algorithm Recall

	20%	**40%**	**60%**	**80%**
SAGA	0.93	0.97	0.97	1.0
Vote	0.93	0.97	0.97	1.0
VotePlus	0.93	0.97	0.97	1.0
Sum	0.93	0.97	0.97	1.0
SumPlus	0.93	0.97	0.97	1.0

Results are illustrated in Figure 4.6. Note that the results of all algorithms follow the same line in the graph.

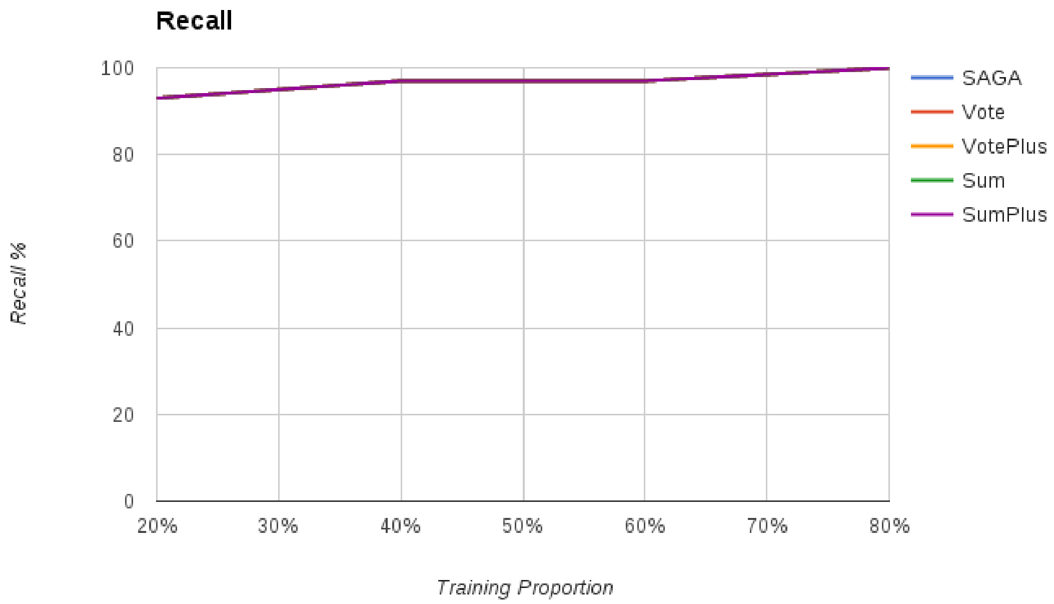


Figure 4.6 - Recall vs Training Proportion

4.6.2 Precision

Of all of the algorithms, SAGA scored best on the precision metrics, followed by Sum and Vote, trailed by VotePlus and SumPlus. This shows that SAGA's filtering method outperformed the more restrictive selection performed by the Sum and Vote algorithms, all of which fared better than the broad list suggestions produced by the VotePlus and SumPlus algorithms (refer to Table 4.3).

Table 4.3 - Algorithm Precision

	20%	**40%**	**60%**	**80%**
SAGA	0.25	0.34	0.35	0.34
Vote	0.27	0.29	0.28	0.25
VotePlus	0.16	0.17	0.16	0.15
Sum	0.27	0.29	0.28	0.25
SumPlus	0.16	0.17	0.16	0.15

Notably, the Vote and Sum algorithms slightly outperform SAGA at the 20% mark, after which SAGA regains its position as the best-performing algorithm in precision. Considering that the algorithms display equal recall at all training proportions, this implies that SAGA has made more speculative suggestions when given a small amount of information, as compared to the number of suggestions provided by the Vote and Sum algorithms.

While the additional suggestions proved to be irrelevant, as more information is made available to the algorithms, SAGA provides less irrelevant suggestions, which is evidenced by its growing precision. This may be considered as being generally desirable behaviour; at low levels of certainty, speculative suggestions are provided (as opposed to showing fewer suggestions, which may slow the process of acquiring more information), but as certainty grows, the list of suggestions becomes increasingly succinct and accurate.

This is shown in Figure 4.7. Vote and Sum share a line, with VotePlus and SumPlus also sharing a line. This shows that the differences in the Sum or Vote methods of distributing influence among suggestions has not made a difference to the results, while the broader list of suggestions in the Plus algorithms has.

As all algorithms obtain near-perfect recall on the MIRFlickr25K database, the F-score of each algorithm follows a near-identical trend to the precision, with F-scores that favour recall over precision obtaining a higher score than those that favour precision. As such, they have been omitted from these results, but are available in Appendix A.

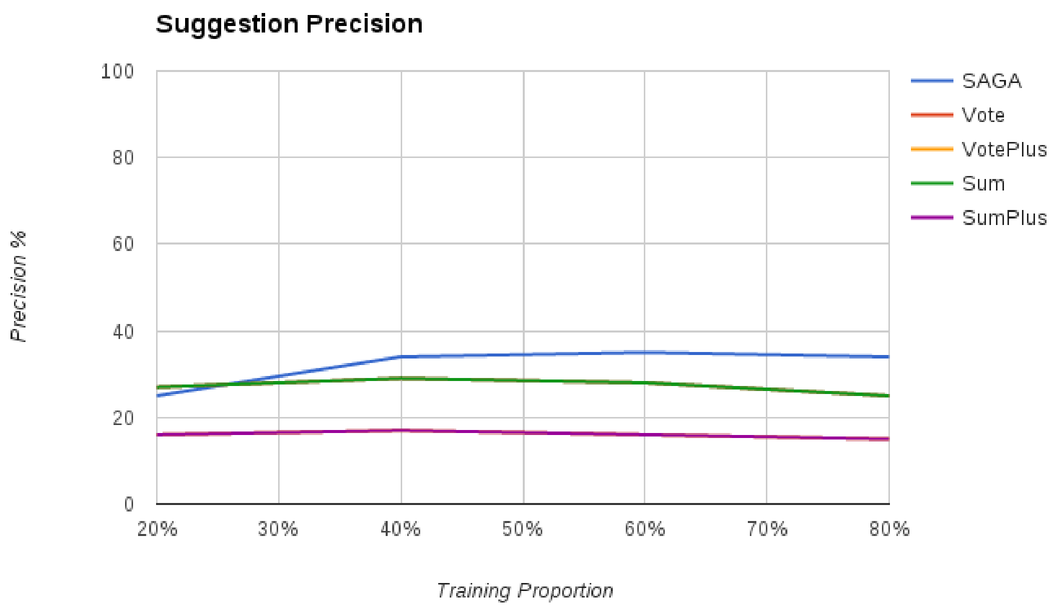


Figure 4.7 - Suggestion Precision vs Training Proportion

4.6.3 Phi Coefficient

The Phi Coefficient again shows SAGA typically outperforming the other four algorithms, followed by Vote and Sum, which obtain similar results, then VotePlus and SumPlus at similar near-zero scores (refer to Table 4.4).

The reason for these low Phi Coefficient scores in the VotePlus and SumPlus algorithms is that they make many more suggestions than the other algorithms. Since they make so many suggestions, their score is comparable with random chance (which has a 0 Phi Coefficient), particularly on a large, well-connected dataset with a small number of concepts, such as this one. No algorithm receives a negative score on this measure, which would indicate disagreement with reality.

Table 4.4 - Algorithm Phi Coefficient

	20%	**40%**	**60%**	**80%**
SAGA	0.32	0.43	0.45	0.46
Vote	0.35	0.36	0.36	0.34
VotePlus	0.02	0.0	0.0	0.0
Sum	0.35	0.36	0.36	0.34
SumPlus	0.02	0.0	0.0	0.0

Figure 4.8 shows the results obtained for each algorithm on this measure. Vote and Sum share a line, while VotePlus and SumPlus also share a line. SAGA has lower precision at 20% until about 25% due to its slightly lower precision, during which the Sum and Vote algorithms briefly outperform it, but not by a large margin.

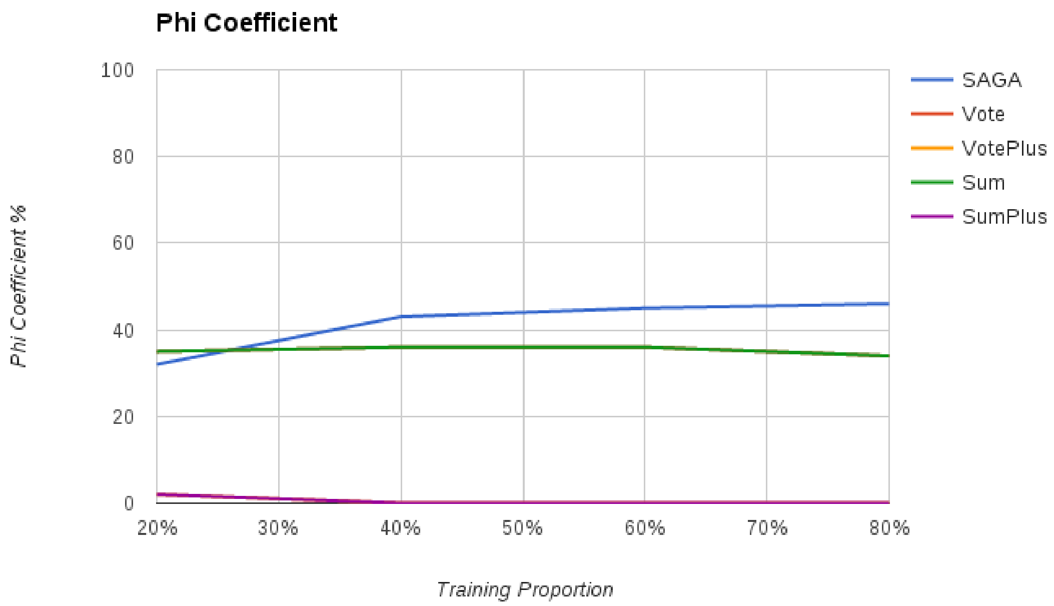


Figure 4.8 - Phi Coefficient vs Training Proportion

4.6.4 Precision @ 5

The precision @ 5 positions results control for the length of the list of suggestions made by each algorithm, and it is here that we see the VotePlus and SumPlus algorithms make a resurgence. SAGA does comparably well, with the SumPlus algorithm taking the lead with a small margin. The Vote algorithm is an outlier, with consistently lower results compared to the other four algorithms (refer to Table 4.5).

Table 4.5 - Algorithm Precision @ 5 Positions

	20%	40%	60%	80%
SAGA	0.7	0.68	0.63	0.58
Vote	0.34	0.35	0.32	0.31
VotePlus	0.61	0.57	0.59	0.56
Sum	0.71	0.71	0.67	0.62
SumPlus	0.71	0.71	0.7	0.63

These results are illustrated in Figure 4.9. All algorithms tend to perform slightly worse at higher training proportions, with results dropping from approximately 70% to approximately 60% accuracy across the proportions for the SAGA, Sum and SumPlus algorithms. This evaluation suggests that the ranking method employed by the Vote algorithm does not effectively place correct suggestions in the top 5 positions.

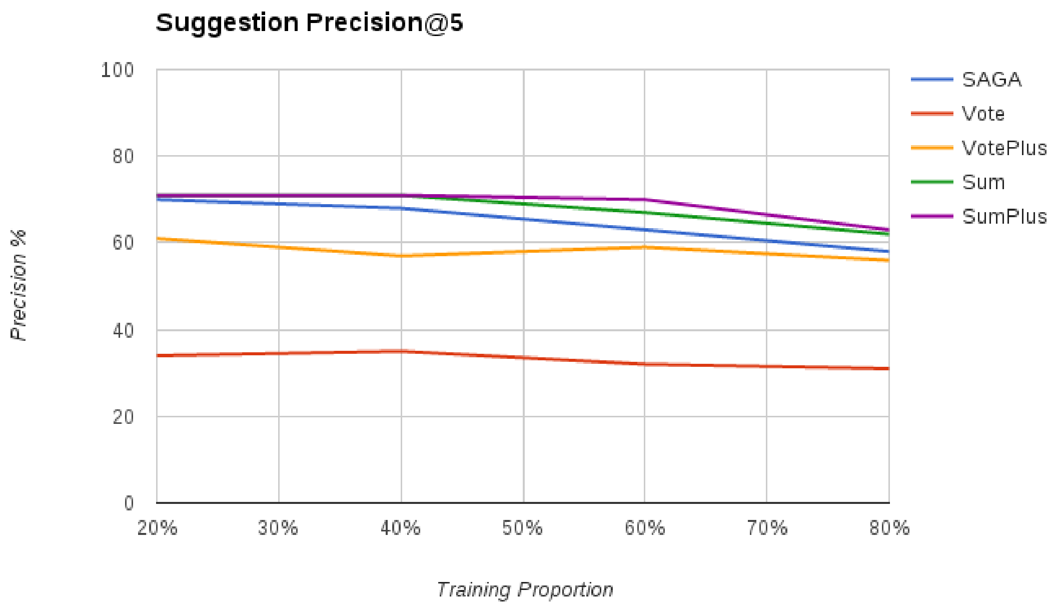


Figure 4.9 - Suggestion Precision @ 5 vs Training Proportion

4.6.5 MRR

The Mean Reciprocal Rank corresponds to how effectively each algorithm places a correct suggestion in the upper suggestions of a list. In this task, SAGA, Sum and SumPlus obtain excellent results, with VotePlus not far behind.

The Vote algorithm, similar to its results in the precision @ 5 evaluation, is not shown in these results to be particularly effective at placing a correct suggestion at the top of the suggestion list, with a 0.34 result suggesting that the first correct suggestion is positioned on average in 3rd place, which improves to approximately an average of 2nd place as more training data becomes available (refer to Table 4.6).

Table 4.6 - Algorithm MRR

	20%	**40%**	**60%**	**80%**
SAGA	0.93	0.97	0.97	1.0
Vote	0.34	0.49	0.49	0.44
VotePlus	0.93	0.95	0.95	0.96
Sum	0.93	0.97	0.97	1.0
SumPlus	0.93	0.97	0.97	1.0

These results are illustrated in Figure 4.10. SAGA, Sum and SumPlus share a line. Not included in this results section are the scores obtained in the Success@N tests, as those results are highly comparable to the results obtained in these MRR results, with Vote having low scores, and the other four algorithms obtaining very similar, excellent results.

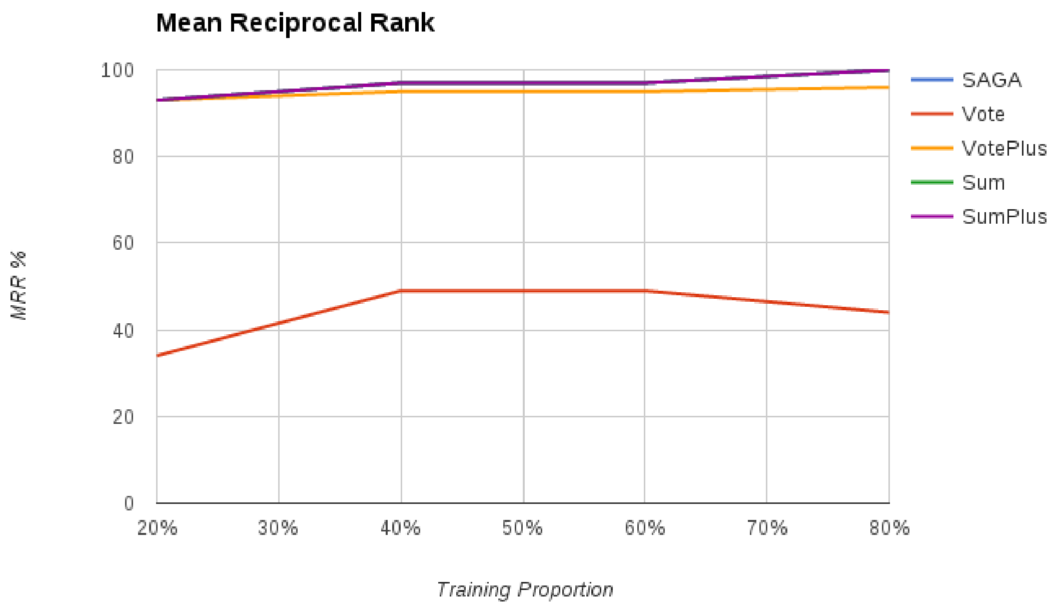


Figure 4.10 - Mean Reciprocal Rank vs Training Proportion

4.7 Discussion

From these results, we can see that SAGA obtains superior suggestion performance when compared to the Vote and VotePlus algorithms. While all three algorithms obtain near-perfect recall on the large, well-linked MIRFlickr25K dataset, SAGA is able to make substantial improvements regarding precision through its filtering process, which allows it to reach a higher Phi Coefficient and F-Score at all weightings.

SAGA obtains mostly favourable suggestion performance when compared with the Sum and SumPlus algorithms. While Sum and SumPlus provide slightly better performance when one constrains results to the first five positions using the Precision @ 5 condition, SAGA obtains better results overall on an unconstrained precision measurement, leading to a better Phi Coefficient and F-Score at all weightings outside of a narrow range where limited training material favoured the Sum and Vote algorithms.

Overall, SAGA obtains very strong results when compared to the other four algorithms on unconstrained and constrained tests. This is particularly interesting when one accounts for the @ 5 condition for success or precision, or the MRR, as this shows that even after eliminating the filtering process employed by SAGA as a variable, SAGA provides a competitively accurate suggestion mechanism.

Out of the other four algorithms, SumPlus obtains slightly better results than its nearest competitor, VotePlus. While the Sum and Vote algorithms obtains better results in measurements where the length of the list of suggestions penalises algorithms that make many low-probability suggestions, the SumPlus and VotePlus algorithms have significantly better success at boosting high-probability suggestions towards the top of the list of suggestions. This is desirable when interacting with human users, as the users are likely to focus on suggestions towards the start of the list, and ignore the majority of the suggestions when they become less helpful towards the middle and end of the list.

VotePlus does possess one notable advantage over the SumPlus algorithm: it is the algorithm that has been recommended by its authors, and obtained higher performance in their performance tests. This means that those using the work conducted in that paper are more likely to employ a VotePlus algorithm in their own work, and for this reason, VotePlus was favoured in this PhD thesis as the competitor to SAGA in the following chapters.

It is important to recognise that this disagreement in test results indicates that when working on different datasets, or when using different metrics to determine an algorithm's success, the ranking of algorithms may change. This highlights the importance of conducting user acceptance testing and a case study application, presented in the following two chapters of this thesis, as these additional experiments will provide a better indication of an algorithm's suitability within a digital heritage application area than can be determined through algorithmic testing alone.

Chapter 5: Semantic Annotation by Group Exploration (SAGE)

5.1 Overview

In this chapter, we describe how the SAGA algorithm can be used as a component of SAGE to support groups of users who explore a project and contribute new information to it. SAGA supports users by helping them find the information that they are interested in, and encourages them to contribute new annotations through a process of suggestion acceptance.

As part of this process, we identify the challenges and benefits that a diverse range of users are likely to bring to a project, including topics such as trust and expertise. We examine user behaviour patterns and how these behaviours impact the user's interaction with SAGE. This affects the design of the user's work environment, specifically the user interface of SAGE and the kinds of operations it supports.

The second half of the chapter focuses on evaluating SAGE and the user acceptance of the SAGA algorithm. This process employs evaluations of both the implicit and explicit acceptance of users to produce a complete picture of user interaction with SAGE. From this data, we calculate measures of suggestion acceptance and user productivity for both the SAGA and VotePlus algorithms to contrast and analyse how varying algorithms in the same work environment (SAGE) affect results.

5.2 Related Work

The key related works examined in this section focus on the individual and social factors which influence collaborative work using formal or informal information management systems, which will also impact this project when users utilise SAGE to explore digital heritage collections.

We also review research that provides techniques for evaluating a user's acceptance of a given technology as a means of quantifying whether SAGA and SAGE are effective approaches for producing annotations in heritage collections.

5.2.1 Social Factors

The challenges of social computing include issues relating to the group of users, such as the formation and stability of social groups, motivating users to contribute, verification of information, establishing authority and trust, reputation, governance of social groups and the management of intellectual property and other legal concerns (Parameswaran & Whinston, 2007).

Of primary concern is the formation and stability of social groups and motivating users to contribute to the group. As the amount of information that can be collected through social computing is directly affected by the number of participants and the level of engagement the participants have with the project, a number of studies have been conducted in this area to identify the factors which affect engagement.

Sutanto, Kankanhalli & Tan (2011) investigate the effects of the *sense of virtual community*, the feeling a user has towards a group which influences their motivation to contribute to and remain within the group. The authors concluded that the most important factors for maintaining this sense of virtual community is to satisfy the informational, instrumental, entertainment and self-discovery needs of members, while social enhancement needs, such as receiving recognition from their peers, were unexpectedly less significant.

This suggests that users seek information and instrumental assistance from groups to complete tasks relating to their personal interests, that users seek entertainment and to learn more about themselves from interaction with other members of the group, but gaining reputation and status within a group is not a strong motivator. It was noted that this result may be specific to larger social groups, and that social enhancement may still be a motivator in smaller groups.

In addition to the sense of virtual community, Carroll (2010) identifies awareness of group activeness as another important factor for collaborating as a group. This relates to the sense that other members of the group are actively contributing to areas relating to the group's interests, as opposed to merely being idle members of the group that benefit from the efforts of others. Methods proposed for enhancing the perception of group activeness includes building systems which allow members to passively contribute to the group through their interactions rather than requiring an active sharing process to be initiated by the user (consider social bookmarking and filtering), as well as providing systems by which activity can be made visible to other members of the group (such as usage statistics, notifications and RSS feeds). These are key technical requirements for SAGE to address.

Nov & Wattal (2009) identify an additional factor that affects the motivation of users to contribute information: privacy concerns. In their study, they determined that a number of factors relating to privacy affected a user's willingness to collaborate and share, including their trust of the other members of the group, the duration that users have been members of the group, the level of control users have over what is shared and who it is shared with, and the social position the users have within the group. Users were most willing to share when they were engaged with the community over a longer period of time, able to only share information they were comfortable with sharing, and were receiving benefits from sharing information with the group. This suggests that control over the users who can access individual projects in SAGE is an important concern which must be addressed as part of encouraging user contributions.

5.2.2 Individual Factors

Additional challenges that can arise for individuals wishing to participate in social computing include the limitations of the technology that is used, issues relating to individual motivations, and potential issues that can arise between individual members in a group (Wang & Fesenmaier, 2003).

Limitations of technology range from ineffective interfaces to facilitate a task, the transfer of data between the user's computer and an online data repository, the transfer of data between users, indexing and accessing information, effectively presenting information, managing security, maintaining privacy and adhering to legal requirements in different countries (Wang & Fesenmaier, 2003).

Simple examples of where technology can limit a user's participation include slow network access, where a user may not be able to stream rich media such as videos at an acceptable rate over their Internet connection, or interfaces that neglect to account for the disabilities that a user may have, such as visual impairment.

Issues with an individual's motivation can include untrustworthy behaviour and an exaggerated perception of the user's authority on a subject matter. These issues can lead some users to seek to influence the collaborative efforts of the group for a malicious or self-serving purpose (Wang & Fesenmaier, 2003).

For example, some users who perceive that they have greater expertise or better communication skills than other users may seek to aggrandise themselves in an online community. Trust models and trust systems can be developed to mitigate these effects, such as a user-based rating system which allows advice provided by these individuals to be identified as inaccurate (Leavitt & Clark, 2014).

Two or more individuals in a group can sometimes have conflicting personalities, which can lead to arguments that threaten the group's cohesion and may make users hesitant to continue their participation in the group. Addressing these issues require facilities for resolving disagreements and moderating dissenting or conflicting opinions on a subject matter (Wang & Fesenmaier, 2003).

These are very difficult issues to address in SAGE. The SAGA algorithm does not detect and treat inaccurate or dissenting information differently from any other piece of information, and while the presence of this type of information will not prevent SAGA from working, it will negatively influence suggestion results in a manner which affects their usefulness. For this reason, SAGE relies heavily on the judgement of the coordinators of individual projects, who are able to moderate disputes and resolve disagreements using channels of communication that lie outside of SAGE, and can enforce their decisions using the facilities that SAGE provides.

5.2.3 Evaluation Strategies

In their evaluation strategy paper, Shani & Gunawardana (2011) highlight the comparative difficulty of collecting user data compared with running offline experimentation on an algorithm, which does not require the participation of any users. They recommend that tests are kept short, focussed, and are checked for inconsistency, and that these small-scale user tests collect a range of information at a high level of granularity to allow for unforeseen avenues of investigation to be later pursued using the data collected as part of the experiment.

The Technology Acceptance Model (TAM) (Davis, 1989) is a theory from an information systems background that attempts to evaluate systems based on two crucial factors, perceived usefulness and perceived ease of use. This model can be used in conjunction with a TAM survey to analyse the responses provided by responders to determine how likely the system is to be accepted and utilised by users. The TAM has been revised twice to incorporate an increasingly complex understanding of the factors that influence a user's perceptions of whether a system is useful (Venkatesh & Davis, 2000; Venkatesh, Morris, Davis & Davis, 2003).

From a systems engineering background comes the Software Usability Scale (SUS) (Brooke, 1996; Brooke, 2013), which is used to provide a "quick and dirty" subjective evaluation of a system's usability. It was originally designed as a tool for improving the software usability of office systems, but has since gained a wider following in usability engineering. The SUS model is conducted at a more subjective level than the TAM model, which allows it to be applied on a wider range of systems which can then be approximately compared. However, it lacks the benefits provided by the underpinning theory of the TAM model, which limits a researcher's ability to analyse why a system received a particular SUS score.

5.3 Challenges

The performance of an agent, whether they are software or human, is not always perfect. Users will provide a diverse range of types and quality of information when interacting with information systems, and while a diverse range of perspectives, knowledge and experiences is likely to benefit a group, to naively assume that their performance is guaranteed to be without flaws can lead to unfortunate consequences which can compromise the integrity of the information captured by the multiagent system.

Factors that have a significant influence on the quality of an agent's work are expertise and certainty. Examining these factors helps with designing techniques which limit the ways in which inaccurate information can be introduced into a system. These inaccuracies can be in the form of misinformation, from erroneous or uncertain calculations; or disinformation, from deceptive communications (Santos & Li, 2010).

5.3.1 Expertise

Some agents may be able to fill a role more effectively than other agents, which is especially true for human agents. We can consider these agents as having differing levels of expertise; some being novices, others being experts. Should an expert make a judgement that goes against the judgement of a novice, we would be inclined to side with the expert. This introduces the concept of agents having a certain level of authority. Authority can be viewed as metadata that can be contributed along with certainty and trust values to an agent's decision.

Users with a wealth of knowledge or experience in a relevant project are likely to provide high-quality contributions that are useful and interesting to other users. This said, users with a basic knowledge of the subject matter of the collection can still contribute valuable information by identifying characteristics about the objects stored in the repository. A simple example might be classifying images as "country" or "city", or identifying some of the things they see in the images such as "house", "car" or "people". This provides information that SAGE can use to provide better suggestions when expert users are exploring and contributing to the collection. For instance, when identifying a well-known historical house in the city, "house" and "city" concepts annotated to an object will help to improve suggestions substantially.

To determine if a human agent is an expert or a novice in a role, we can explicitly evaluate their experience by asking the agent to produce a rating for itself, ask a manager agent to provide a rating, or ask the agent's peers to produce a rating. This can be likened to an interview or resume used during the job application process. Alternatively, we can evaluate the agent as they perform their role through profiling (Budescu & Chen, 2014).

Profiling an agent requires some means by which to determine if the decisions being made are "correct". This can take the form of a predetermined set of responses, analogous to an exam, or comparing the agent's responses with other agents currently filling or previously filling the role. We would assume that an agent who agrees with agents previously identified as experts is themselves an expert.

The difficulty with profiling is that we must presume that there is a correct expert response, although in reality there may be several correct and defensible stances which experts might take. Additional flexibility (at the cost of complexity) can be introduced in this system by allowing an agent to occupy additional levels of authority between a novice and expert, or employ a ranking system where all agents occupy a position in a hierarchy (Noll, Au Yeung, Gibbins, Meinel & Shadbolt, 2009).

5.3.2 Certainty

Agents may be uncertain about the correctness of a piece of information they are contributing. When there is uncertainty over whether a concept should be annotated to an object, an amendment to that concept or a new concept that implies uncertainty is very useful. For instance, “May be a famous scientist” will partially match the concept “famous scientist”, allowing SAGE to make the connection for the purpose of suggestions while still representing the degree of uncertainty held by the annotator.

Systems which treat information as existing in a binary state of true/false are not flexible, adaptable and robust in the face of information which may be conflicting or controversial, which can lead to unexpected complications in the system’s operation. A mechanism used to deal with this is the field of Fuzzy Logic (Zadeh, 1965), which allows information to exist in a quantified uncertain state between true and false.

Using this quantity, certainty isn’t just a means of dealing with potential problems that are caused by the binary true/false model, but it is also an additional piece of metadata that can be stored with the information that has been contributed by the agent, which can be useful in further calculations. For example, if an agent is frequently more confident in information contributed on the topic of hiking, but less confident regarding information contributed on the topic of cycling, we could detect that the user is possibly a recreational hiker without explicitly asking them.

5.3.3 Trust

Agents may choose to engage in deceitful behaviour to further their own agendas. Alternatively, agents may make a genuine mistake. Both situations result in incorrect information being contributed to the multiagent system, so having a means of detecting this kind of information is desirable. Achieving this detection through a reputation system is made more challenging by the fact that deception is typically a rare incident, and should an agent's history be honest, they may be deemed trustworthy at the time when they perform their first deception, and so they will not be detected (Santos & Li, 2010).

One method suggested by Johnson *et al*. (2001) to overcome this is to model the expected behaviour of the agent and, if an anomaly is detected from the expected behaviour, formulate and test a possible hypothesis as to why this occurred. A challenge to implementing this approach is that in multiagent systems, agents may not individually possess sufficient information to "see the bigger picture" and create an accurate behaviour model with which to confidently detect deception being perpetrated by another agent.

Santos & Johnson (2004) propose that a prediction system based on the correlation of multiple agent's past opinions provides a potential measure by which an individual's deception can be detected. Deceit is signalled by an unexpected change to the level of correlation with the agents that the agent previously correlated strongly to. While this approach may not be able to accurately distinguish misinformation from disinformation, it has a suitable detection rate for the combined forms of incorrect information and significantly improves upon the results achieved by other systems (Santos & Li, 2010).

5.3.4 User Engagement

Computer literacy is one of the main hindrances to user engagement (Wang & Fesenmaier, 2003). If a user perceives the technology used in SAGE to be difficult and frustrating, they will be less inclined to participate in research projects and will instead pursue the approaches they are familiar with. A user who is confident with computers and web applications is likely to contribute at a faster rate than one who needs additional practice or is cautious.

User engagement is crucial for both experts and volunteers in SAGE. Every participant in a digital heritage project has the opportunity to contribute a unique and useful perspective on the materials being examined, and a participant's awareness of this is important (Ling *et al.*, 2005). A volunteer's perspective can effectively supplement and augment the domain-specific tags or subject headings provided by experts (Rolla, 2011), while experts are capable of maintaining opinionated positions irrespective of the group consensus (Meade, Nokes & Morrow, 2009), which may influence the group towards more strongly-supported theories. Both types of contributions are therefore complementary and valuable.

5.4 The SAGE Environment

We have presented SAGA, an algorithm that uses text similarity, tag co-occurrence and neighbour voting as a means of providing high-quality suggestions early in the semantic network's development to assist with the annotation process. SAGE is a system which captures user-provided information in the form of annotations, and uses SAGA to make suggestions to prompt users to contribute additional information. In this manner, SAGE is the multiagent environment in which SAGA and users participate in a positive feedback loop, where the suggestions accepted by users provide new information which the system can use to provide further suggestions.

The initial sparsity of information when a semantic network is being developed poses a major challenge for suggestion mechanisms that rely on annotations such as semantic tags to provide relevant results. This is of particular interest in application areas such as digital heritage, where a large number of images need to be annotated by a limited number of volunteers and a suggestion mechanism would provide an effective way of rapidly annotating the collection.

By analysing the behaviours of users during the information seeking and contribution process, and by investigating the controls that can be implemented to guide and coordinate the interaction between human and software agents, we allow contributions to be securely and reliably obtained from a diverse group of users.

5.4.1 User Behaviours

While the SAGA similarity algorithm is the heart of SAGE, the other key element to the annotation process are the users themselves. By viewing objects and concepts, users automatically apply the SAGA algorithm to what they are viewing and are presented with the opportunity to quickly and easily contribute new information back to the repository either using these suggestions, or by adding entirely new information.

Depending on the circumstances under which users browse the content of a project, users can express a number of information-seeking behaviours. As part of developing a supportive software environment, the design of SAGE examines these behaviours and seeks methods by which they can be accommodated and augmented. Analysing the support provided for these behaviours also presents a possible avenue by which SAGE could be evaluated for effectiveness.

Exploring

When a user is exploring a collection, they view the content of a collection with no information-seeking goal in mind. In SAGE, users can achieve this by viewing the listing page of either objects or concepts or by selecting an individual object/concept as a starting point and using the next and previous controls to view each subsequently until one catches their interest.

During this process, users will be able to view the annotations that have been applied to each of the items they visit and see suggestions which encourage them to contribute information to what they are passively viewing. These annotations also provide ideas for topics that the user might want to investigate in more detail. In this way, SAGE supports users while browsing, but simultaneously provides encouragement for users to alter their behaviour towards either a searching or contributing activity to actively participate in the project.

For example, refer to Figure 5.1. In this image, users can see several travel-related topics that might interest them, such as “Holiday Destination” or “Island Cruises”. Users might also be able to identify the exact island in the picture using their local knowledge, which would be of substantial value as metadata for the image.

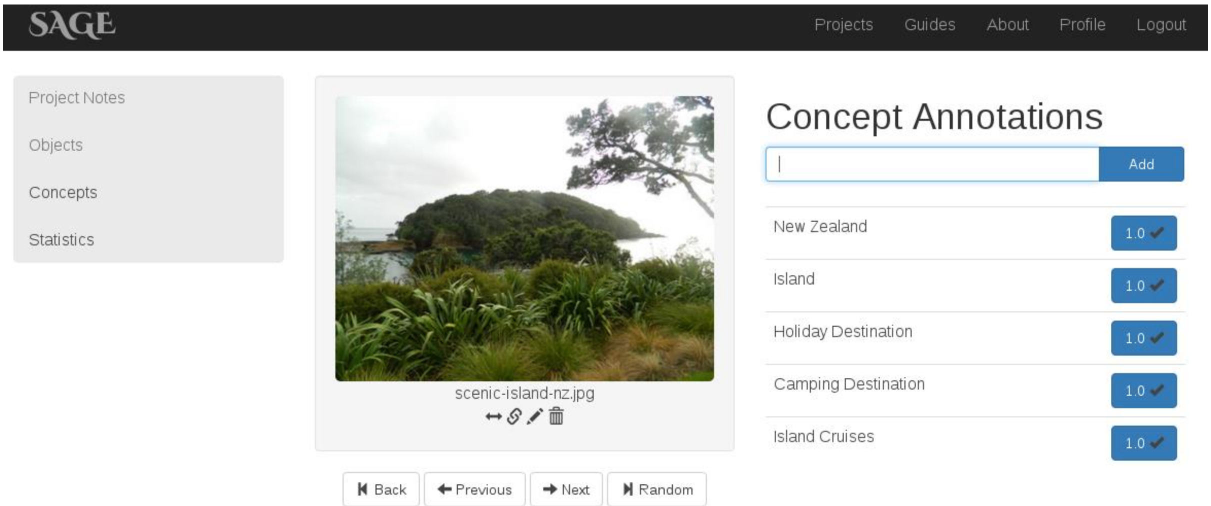


Figure 5.1 - Browsing a SAGE Project

Searching

When the user has a rough idea of what they wish to search for, using the annotations that have been assigned to an object or concept that is an approximate match of what they seek is a technique by which they can navigate to the exact concept or object they are looking for.

For instance, if the user starts with a concept for "Islands" they will see the objects that have been annotated with this concept. If they see one in particular that they are interested in, they can navigate directly to that object to see what other, more specific concepts have been annotated to that image, such as "Caramoan Island" or "Hainan Island". This is illustrated in Figure 5.2, where the user has arrived at their desired object after following a link from the "Island" concept.

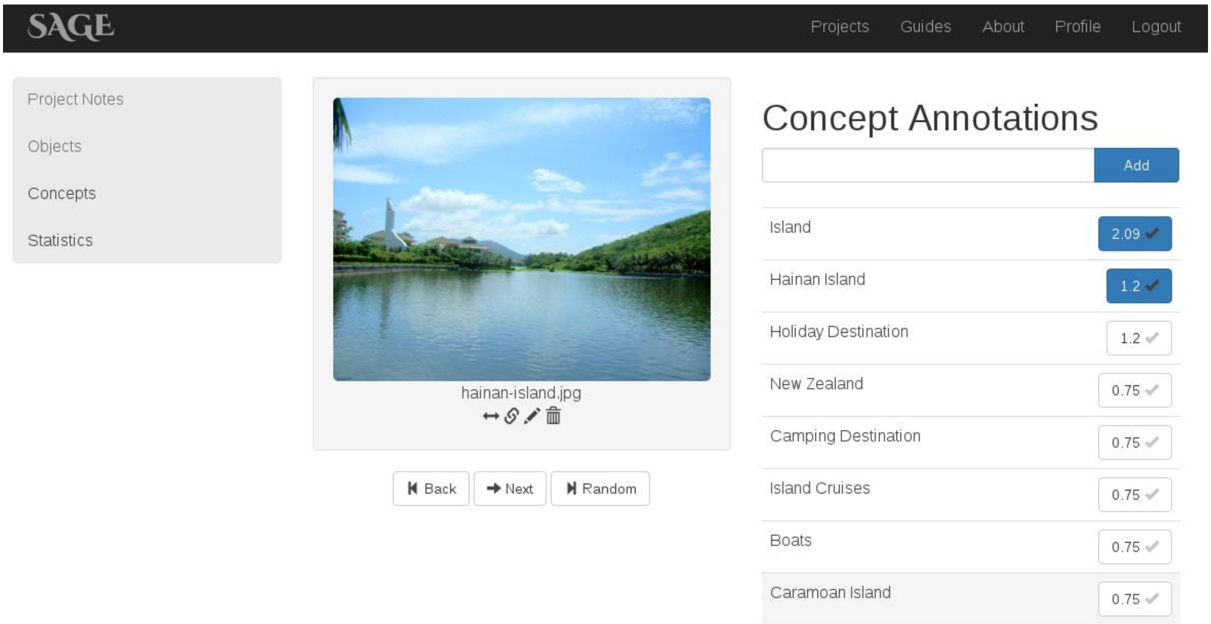


Figure 5.2 - Search Refinement in a SAGE Project

If the user knows precisely what they wish to find, they can create a new concept describing what they seek. If that concept already exists, they will be redirected to the merged concept and their search will be complete. If the concept is unique, users will be shown initial suggestions which they can accept to refine the suggestions provided by

SAGA until they have collected together the materials that they are looking for, at which point their search will be complete.

As this concept and the objects the user has annotated to it persist in the SAGE project, it is very easy for the user to access the results of this search process in a subsequent visit by navigating directly to it. Perhaps more importantly, it has contributed the perfect response to provide to a different user who is searching for the same kind of material as the original user.

In the case that two users perform this process separately from one another and two similar concepts are formed sharing many (but perhaps not all) of the same concepts, changing the description of both concepts to be identical will trigger the concept's flattening behaviour and merge the two concepts.

In this way, SAGE supports users who wish to search either passively, for what already exists, or actively, by contributing new information in the form of a concept to the project. This follows human-centric computing practices giving users freedom of action while encouraging them to contribute new knowledge to the project.

Contributing

While a user is viewing the objects or concepts in SAGE, they are continually presented with existing annotations and new suggestions determined by the software user agents through the SAGA suggestion mechanism. A user can contribute to the project and improve their suggestion results by correcting mistaken annotations and accepting relevant suggestions. This allows noncommittal users to gradually contribute as they perform other tasks.

In Figure 5.3 below, the user can see useful information about a memorial on Catalina Island. However, they may notice that "Wrig**ley** Memorial" has been misspelled as "Wrig**ely** Memorial", and a contribution they can make is to correct this spelling error.

Alternatively, users may add new suggested annotations such as “Island”, which are relevant to this object.

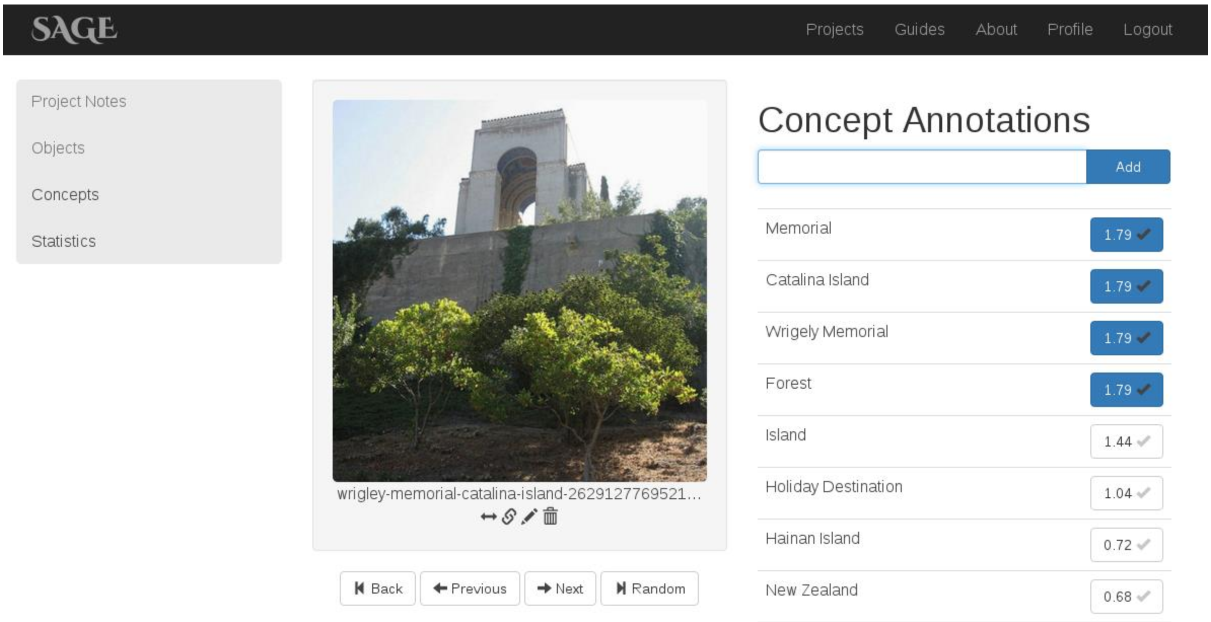


Figure 5.3 - Search Contributions in a SAGE Project

Some users will actively contribute additional objects as part of their tasks, which makes these objects available to other participants in the project. Some projects consist of a definitive set of objects that make up the project’s collection, in which case a user can add all of these objects in a single full import at the very beginning of the project, while other projects may have a steady stream of new materials that can be used as the basis for objects.

In either case, the user adds new objects to SAGE for their own purposes, but these new objects can indirectly help other users with what they are working on, and these other users may in turn help the original user by annotating the objects for them. If a user simply wishes to be helpful and provide annotations for newly added objects, these can conveniently be found on the object index page, as the index page shows objects in order of recency of addition to the project.

If multiple users are working on separate sets of objects that share some semantic concepts, the likelihood of these concepts colliding (i.e. a direct match) or at least sharing text, is high. This provides an opportunity for SAGA to suggest to one user a concept created by another user, and once initial connections are formed between the separate work being performed by the two users, an increasingly large number of suggestions can be drawn from each user's work, allowing for a period of rapid annotation progress as relevant suggestions are accepted by both users.

Even without these initial links being speculatively formed between concepts, popular concepts are likely to be made as suggestions anyway. This has the effect of linking new contributions to the well-established core contributions of the project, which helps to provide better suggestions. So long as users link their recent work to at least one concept in the well-established portion of the project's semantic network, their new work will be reachable by tag co-occurrence.

Administrating

Users who have the administrator level of privilege in a project are likely to invest some of the time they spend using SAGE in the process of administrating their projects. Tasks administrators might perform include recruitment, communicating with participants, monitoring the changes made by contributors, and promoting users to higher levels of access privilege.

While it might seem like the administrators are spending time on upkeep tasks which aren't a major part of SAGE, they are actually in a very unique position, which is a rare example of one user agent functioning as a manager for other user agents.

The process of generating a contributor key and sharing it with colleagues in their department via email, for example, will (hopefully!) lead to new users in a project. By delegating a degree of authority and responsibility to each new participant, the team is capable of cumulatively achieving a goal of considerable scale, and one which the original administrator may not have been able to complete alone.

5.4.2 Access Control

Access is controlled in SAGE by a key system. Administrators of a project may choose to generate keys for any of the available roles and share these keys using a system outside of SAGE, such as email or even just writing it down on a piece of paper and handing it to a coworker.

The available levels of access are Administrators, who can edit both the users and content of a project, Contributors, who can edit the content of a project, Annotators, who can annotate samples of projects, and Viewers, who can view the content of a project.

The hierarchy is shown in Figure 5.4 below, showing the (typical) linear progression in privilege from viewers to contributors to administrators. Annotators are a special case between viewer and contributor; while they have annotation privileges, their privileges are in a separate sample project accessible only by them and the administrators of the main project. Trusted viewers (who build trust via external channels such as email discussions) may be directly promoted to contributors, or may be given an opportunity to prove the value of their contributions as an annotator before eventually being promoted to a contributor in the main project.

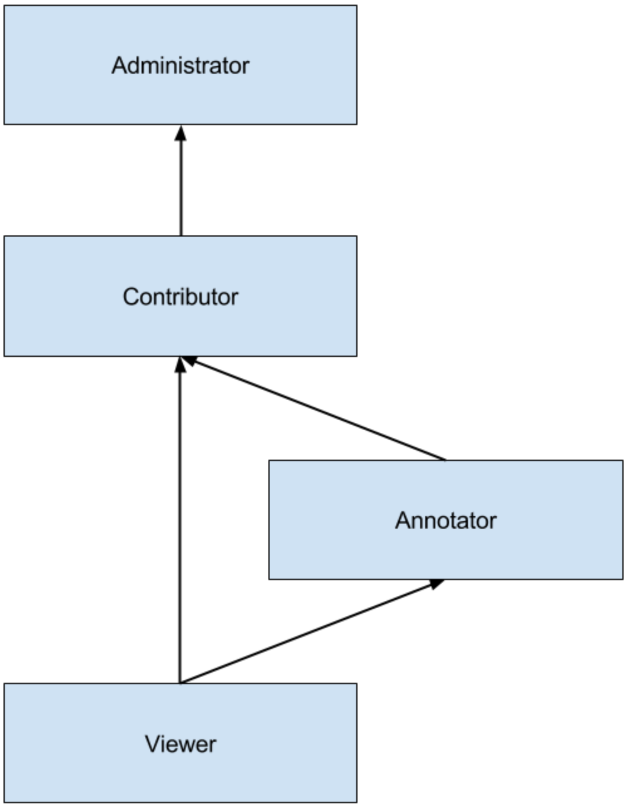


Figure 5.4 - Role Hierarchy

Keys can be redeemed an unlimited number of times to grant that level of control in that specific project, so low-risk keys such as the Viewer key can be shared publicly on forums. In the event that Administrators need to tighten security on the keys that have been issued, they can rescind the key entirely or re-generate the key. This is analogous to barring a door, or alternatively rekeying a lock, so that only users who have been provided with the new key will be able to redeem it for access.

To prevent the formation of orphaned projects, which are inaccessible to all users but still reside in the database, when the last administrator leaves a project, the project is deleted. Transferring a project to a new administrator is as simple as providing the new administrator with an administrator key, waiting for them to redeem it, and then having the original administrator leave the project.

5.4.3 Quality Control

SAGE assumes that all contributors to a project are equally trusted. Anyone with the rights to modify the project has full control over the addition, modification or deletion of objects, concepts and annotations, and their contributions are given full weight by the software agents. Because of this, SAGE is intended to be used by small, close-knit groups of contributors with a wider group of viewers, rather than having universal contributor access available to the public, which would leave projects vulnerable to quality control issues such as vandalism.

SAGE offers a special user role, annotators, to address issues of quality control for users who have not yet earned the trust of the project's administrators. Users who redeem an annotator key are given a sample subset of the project to which the key belongs. The annotator is unable to select or modify which subset they have been given, but are able to annotate the subset using concepts of their own creation.

The administrators of the original project are able to view progress being made by annotators, and can import annotations from the sample projects if they deem them to be of an appropriate level of quality. This provides annotations, but also gives administrators of a project the opportunity to see if a new user, acting as an annotator, might be suitable for promotion to a fully-fledged contributor of the project.

Once an annotator has finished with their subset, they can redeem the annotator key again to be assigned a new subset of objects from the original project to continue contributing to the ongoing work. Multiple annotators can work in parallel, allowing for large-scale, secure crowdsourcing. Objects assigned in each sample are chosen randomly from the objects that have not been assigned to that user in other samples, ensuring users are annotating objects that they have not seen before.

The annotator role is also useful when conducting user-based experimentation to compare algorithms. As this role restricts a user to focus on the annotation task on a predefined, fixed-size sample, it provides a means of controlling several variables which could impact the comparability of two samples annotated under more inconsistent conditions.

5.5 Methodology

In the second set of experiments following on from the algorithm performance evaluations conducted in Chapter 4, we evaluate SAGA as a component of SAGE, a larger system that introduces a new variable: the *end user*.

For any human-centric system, a critical issue to address is user acceptance (Davis, 1989; Thong, Hong & Tam, 2002). This acceptance can be measured both explicitly and implicitly. Explicit acceptance is analogous to directly asking the user whether they perceive a system to be useful, while implicit acceptance is determining whether the user utilises a system in practice based on observing their actions. If users voluntarily select to use a system, then the system is much more likely to have real-world impact than a system which users are coerced into using as a requirement of a job or task.

A positive result in this set of experiments would provide supporting evidence for the following:

- SAGE does not have significant user acceptance issues which would impact negatively on the results obtained from a case study,
- SAGA provides suggestions which are actually accepted by users and which has an acceptance rate that increases as more concepts are identified and established, and that;
- SAGA provides suggestions using its domain-specific algorithm which are accepted more frequently than suggestions made by a generic-use algorithm.

5.5.1 Initial Training

Prior to data collection for this series of experiments, users who participated in this experiment were provided with a standardised set of instructions and tasks to complete as part of an initial training activity. The acceptance testing participants needed to demonstrate each skill in the activity before the training progressed to the next stage.

Each user worked on their own workstation while a live demonstration was shown by overhead projector, and questions could be asked at any stage during the demonstration to account for the individual's learning style differences (tactile, visual or auditory learners).

As the number of users who participated in this experiment were larger than the computer lab in which it was conducted, the acceptance testing participants were divided into three roughly equal groups who determined among themselves a day and time when all members would be free. The computer lab was booked for a three-hour session at that time to ensure that the participants would be uninterrupted by members of the public.

Users were first asked to create and name a new, personal project and populate the project with approximately 25 images from a Google image search of a topic that interested them. This was done using the “Add Object” form, and users were encouraged to experiment with adding a single object, or adding a batch of objects with URLs on separate lines. The user was then asked to work through each image and contribute new concepts as they identified them, or annotate existing concepts when they were provided as relevant suggestions. Users were also taught how to modify or delete unwanted concepts (e.g. spelling errors) and to remove incorrectly assigned annotations.

Next, users were asked to return to the project list page and select to redeem a key provided in an information handout. This familiarises them with the ease by which keys can be redeemed, which is particularly important for the main data collection exercise. Redeeming the key gave users contributor access to a shared heritage project, the Nickson AIF collection. This is a modestly large heritage collection provided by the UNE Heritage Centre, selected as suitable in size for a large group of users to collaborate on. Here, they were encouraged to visit randomly selected objects and continue the annotation task as a groupwork exercise.

Finally, the user acceptance participants were asked to fill out a survey before proceeding to the main data collection exercise. This survey consisted of user experience self-reporting, a Technology Acceptance Model questionnaire, and a Software Usability Scale questionnaire (see Appendix B for the complete survey).

5.5.2 Experience Self-Reporting

In the first section of the survey, users were asked to rate their experience level by selecting one of three statements designed to estimate their confidence with using the SAGE application. The description provided for each of the three experience ratings can be referenced in Table 5.1 below.

Table 5.1 - Reported Experience Descriptions

Reference: Experience Level Chart	
1	I can perform simple tasks in SAGE, but I'd like to have someone available who can help me if I get stuck.
2	I can comfortably perform a variety of tasks in SAGE, and I could generally work without supervision.
3	I can confidently perform a variety of tasks in SAGE, and I'd be able to help a new user learn how to use it.

Ideally, users would select 2 or 3, indicating by self-assessment that they have gained sufficient experience from the training. This self-assessment, along with the competency-based training progression, allows us to place higher confidence on the assessment made by the user acceptance participants regarding SAGE in the following questionnaires.

This also acts as a control for the user acceptance participants' learning curve, which is particularly relevant in the set of experiments on user productivity. As these users will have had ample opportunity to learn and practice with the system before commencing these experiments, we control the likelihood of the user being significantly delayed by inexperience or lack of confidence.

5.5.3 TAM and SUS Surveys

User acceptance was measured explicitly using the widely-used Technology Acceptance Model (TAM) (Davis, 1989) and Software Usability Scale (SUS) (Brooke, 1996) surveys. The Technology Acceptance Model places particular emphasis on the factors leading to the acceptance and use of software, while the Software Usability Scale emphasises the quality of the interface used by the software.

The standard set of questions from both the TAM and SUS surveys were slightly modified to include the name of SAGE and specify the “job” context asked in some of the questions to be heritage annotation tasks using SAGE. Otherwise, the questions were left unchanged and asked in the typical order using a five-point Likert scale ranging from Strongly Disagree through Strongly Agree (Likert, 1932).

Survey responses were requested of all users of SAGE, who fall under two main groups. The Heritage Centre Participants are both experts and interested individuals with affiliations to ongoing, real research projects in the UNE Heritage Centre. The seventeen Acceptance Testing Participants are users specifically recruited for evaluating SAGE in this experiment, and while they have limited experience with heritage projects, they provide a large amount of test data on the usability of SAGE’s user interface.

The surveys were deployed to users through Google Forms, and the results were collected in Google Sheets (Google, n.d.b). A separate but almost identical survey was given to each of the two groups, with the only change being that the self-assessed training question was removed for the heritage centre participants’ survey as they did not participate in the same formal training as the acceptance testing participants.

The use of two online survey forms allows for the surveys to be kept completely anonymous, which the users were advised of before completing the surveys, but still allows each group to be kept separate for the purposes of identifying informative differences between the two group’s perceptions of SAGE.

5.5.4 The Saumarez Collection

The Saumarez Collection is a collection of several thousand heritage images from the New England, NSW and beyond, taken by members and friends of the White family.

The collection has kindly been provided by the UNE Heritage Centre for use in this research project, and is one of the largest digitised collections in the centre (see Figure 5.5 for some examples).

Figure 5.5 - Saumarez Collection Example Images

Refer to Table 5.2 for statistics about this collection. Note that the number of concepts and annotations in this new SAGE heritage project is initially zero.

Table 5.2 - Saumarez Project Statistics

Saumarez Combined Statistics	
Objects	1411
Concepts	0
Annotations	0

After completing the initial training and surveys, each volunteer was given an annotator key for the Saumarez Collection. This allowed participants to receive small sample of 25 images to annotate using either the SAGA or VotePlus algorithm, with the algorithm alternating for every new sample. The initial algorithm was randomly selected for every participant.

Participants would annotate the images with the help of the algorithm provided for that sample. Once they had completely annotated their sample, a participant would redeem their annotator key again to obtain the next sample of images with the alternate algorithm, and they would begin the process again.

Participants were instructed to work individually, quietly, and to focus exclusively on the annotation task. Users were, however, encouraged to ask questions or indicate any issues they were experiencing in order for these to be corrected and the annotation task to proceed as smoothly as possible.

Participants were also instructed to annotate an even number of samples, ensuring the same number of samples were collected from each user for the both the SAGA and VotePlus algorithms. Measurements relating to both the process and product of annotating the samples were collected and analysed to provide the results for this chapter.

5.5.5 User Productivity

One of the key measures by which the SAGA and VotePlus algorithms can be compared is to measure whether either algorithm leads to improved user productivity in terms of either producing more annotations per minute, or producing a better-linked network of objects. Improvements to user productivity in either regard would be a strong motivator to favour one algorithm over the other.

Productivity metrics were tracked during the annotation of the Saumarez Collection samples, specifically the number of annotations and the time interval in which the annotations were added. As each annotation in the database has an automatically-created timestamp of its creation, times intervals could be calculated using this data.

To control for unexpected interruptions, the work of users was recorded in *clusters*, which collects annotations into groups where each annotation is within a 2 minute interval of the previous annotation. This time was selected as being long enough to account for a very slow annotation (such as needing to check the spelling of a location or person's name), but short enough for any significant distraction to be detected. For example, if a user annotated half their sample and then had to leave the computer lab due to a fire drill, they could return 5-10 minutes later after the drill and resume their work with the delay being automatically excluded from the results. If the number of clusters are high, this would indicate that users were highly distracted while performing the annotation work. Ideally, all users would be uninterrupted and complete each sample in a single cluster.

To measure productivity in terms of work rate, we simply need to calculate the average number of annotations per minute within that cluster. The time spent in each cluster is measured in seconds, therefore:

$$\textit{Cluster Rate} = \textit{Number of Annotations in Cluster} / \textit{Time Spent in Cluster} * 60 \qquad \textbf{(5.1)}$$

To find the average cluster rate, we take the average of all cluster rates in samples that use the same algorithm.

Average Cluster Rate = Sum of Cluster Rates / Count of Cluster Rates **(5.2)**

To determine how well-connected a sample is, we compare the leaf count in the sample to the branch count. The "leaf" and "branch" terminology is common when describing features of a network tree. Leaves are objects that are connected to the rest of the network by a single connection, while branches are objects that have multiple connections to the rest of the network. Thus:

Leaf Count = Count of Objects with 1 Annotation **(5.3)**

Branch Count = Count of Objects with > 1 Annotation **(5.4)**

Branch Ratio = Branch Count / (Leaf Count + Branch Count) **(5.5)**

To determine the average branch ratio, we take the average of all branch ratios in samples that use the same algorithm.

Average Branch Ratio = Sum of Branch Ratios / Count of Branch Ratios **(5.6)**

A high average branch ratio signifies that objects in samples using a particular algorithm are well-connected. In certain approaches, such as the tag co-occurrence approach used in this research project, a high branch ratio implies that better, more complex information is available with which to make suggestions.

5.5.6 Suggestion Acceptance

The second key measure by which the SAGA and VotePlus algorithms can be compared is to determine whether one algorithm provides suggestions which are accepted more frequently than suggestions produced by the other. “Accepting” a suggestion is defined as the user pressing the add button next to a suggestion rather than typing and submitting it. This was recorded as a hidden field held by each annotation in the database.

A higher suggestion acceptance rate leads to less time spent by users manually entering concept descriptions, and would also lead to a higher reuse rate of existing concepts, which promotes a well-linked network of objects. A high suggestion acceptance rate, particularly early in the annotation process where suggestion recommendations is the most challenging, would be a significant advantage to using one algorithm over the other.

Comparing the frequency by which suggestions were accepted by a user, rather than the user manually annotating an object, forms the basis by which the SAGA and VotePlus algorithms were compared in suggestion acceptance. The acceptance rate of a sample can be found by:

Acceptance Rate = Count of Accepted Suggestions / Count of Total Annotations **(5.7)**

Then, the average acceptance rate of all samples using a particular algorithm can be found by:

Average Acceptance Rate =
Sum of Acceptance Rates / Count of Acceptance Rates **(5.8)**

Due to the behaviour of SAGE that no concepts are initially provided in new projects, an acceptance rate of 100% is impossible since no concepts exist that can be suggested by algorithms. Users will manually create a number of concepts early in the annotation process to establish the “vocabulary” available for annotation, and then accept concepts more frequently as the vocabulary becomes sufficient. This means that acceptance rates start at 0% and improve from there.

A low acceptance rate would persist if the algorithm is unable to make good suggestions to the user, forcing the user to manually enter the concept they want. This can also happen if the vocabulary rapidly grows, with little to no reuse of concepts as the user views subsequent objects in the project. This can happen when objects in a project are completely unrelated to one another and require entirely different concepts, but is very unlikely in a collection of related items such as the Saumarez heritage collection used in this experiment.

5.6 Results

In this section we present the results obtained during the user acceptance-based experiments. We begin with the user experience survey results, followed by the TAM and SUS survey results showing the explicit approval ratings for SAGE. We then present the suggestion acceptance and user productivity measurements, which are implicit approval ratings of SAGE.

5.6.1 User Experience Survey Results

Each of the three groups who completed a SAGE training session were able to finish all of the required exercises, and all groups had a high engagement rate in the training activities, with members typically spending additional time beyond what was required on the individual and groupwork tasks. All members of the acceptance testing participants group were able to contribute responses to the user experience survey.

The anonymous experience self-reporting results for these members are available in Table 5.3 and graphically represented in Figure 5.6 below.

Table 5.3 - Reported Experience

Reported Experience	
Participants	17
Average Experience Rating	2.59 / 3.00
Minimum Experience Rating	2 (7 participants)
Maximum Experience Rating	3 (10 participants)

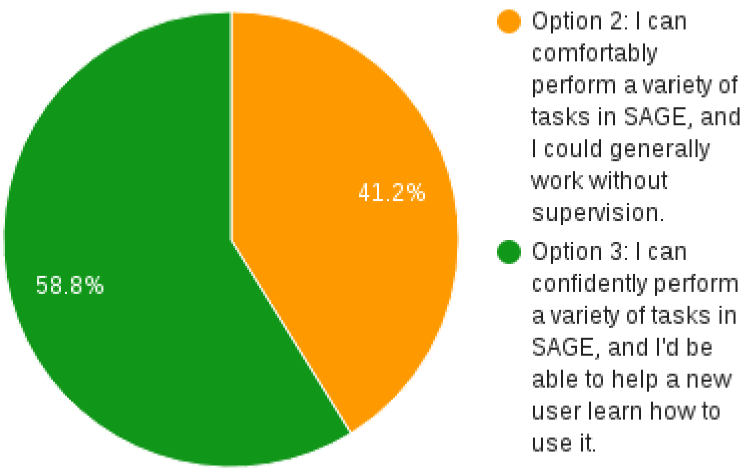


Figure 5.6 - Reported Experience Levels

A majority of the participants indicated that they felt that the experience level that most accurately described their own was a rating of 3, the highest level tracked in the self-reporting question. The remaining participants indicated that they felt they had a rating of 2, with no participants providing a rating of 1. These ratings indicate a high level of confidence with the functionality of SAGE, and support the assertion that the amount of training provided for SAGE prior to experimentation was adequate to avoid a steep initial learning curve falling during the experimentation phase.

5.6.2 TAM Survey Results

Presented here are the summarised responses of the 19 responders to the standard TAM question set. These responders are made up of the acceptance testing participants, plus those that responded from the heritage centre participants. Responders were additionally provided with the following description of a hypothetical job position:

Imagine you have been employed in a job to organise, explore and locate images within an image collection. SAGE is available for you to use in this workplace. Based on this information, provide a rating (1-5) for each statement.

Table 5.4 - TAM Survey Results

Question	Average Rating (1-5)
Using SAGE in my job would enable me to accomplish tasks more quickly.	4.05
Using SAGE would improve my job performance.	4.00
Using SAGE in my job would increase my productivity.	4.16
Using SAGE would enhance my effectiveness on the job.	4.16
Using SAGE would make it easier to do my job.	4.11
I would find SAGE useful in my job.	4.32
Learning to operate SAGE would be easy for me.	4.68
I would find it easy to get SAGE to do what I want it to do.	4.16
My interaction with SAGE would be clear and understandable.	4.11
I would find SAGE to be flexible to interact with.	4.00
It would be easy for me to become skilful at using SAGE.	4.79
I would find SAGE easy to use.	4.47

Of the results obtained (see Table 5.4), the large majority fell within a normal sample distribution in the upper ratings with few outliers (refer to Appendix C and D for a question-by-question breakdown of responses). The strongest ratings were for questions that asked users how easy SAGE would be to learn, while the weakest ratings were for questions that asked if SAGE would improve their job performance or would be flexible to interact with.

Relating these results to the TAM framework, the ideal response would be a high perceived usability and high perceived ease of use, which the TAM framework predicts will lead to a strong intention to use the software, which in turn leads to high levels of real-world usage.

On average, SAGE was perceived as being quite useful (83% on usefulness-related questions) and very easy to use (87% on ease of use-related questions). The TAM model predicts that the perceived usefulness could be further improved through increased experience or job relevance for the participants (Thong, Hong & Tam, 2002), and ease of use could be improved through increased computer literacy and enjoyability of using the software.

SAGE received an overall rating of 51 points out of 60 in the TAM survey, or 85% of the maximum possible result (see Table 5.5 and Figure 5.7 for a graphical representation).

Table 5.5 - TAM Survey Overall Result

Overall Rating	
TAM Average Rating	51 / 60 (85%)

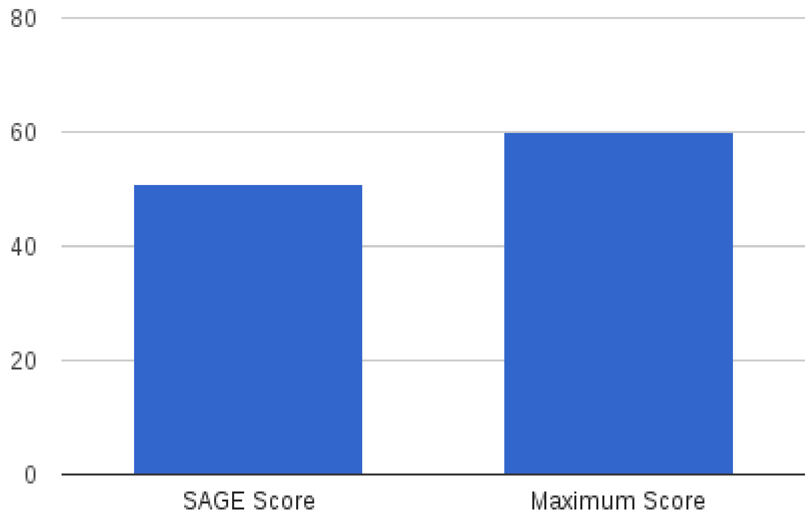


Figure 5.7 - SAGE TAM Score

5.6.3 SUS Survey Results

Presented here are the summarised responses of the 19 responders who completed the standard SUS survey questions, the same responders as above. Responders were instructed to answer based on their experience of using SAGE.

The SUS survey features alternating positive and negative statements from which it is possible to calculate a total score for the website/application being evaluated. For positive statements, a user rating of 5 gives a score of 4 towards the score total, while a rating of 1 gives a score of 0. For negative statements, the scores are reversed, and a rating of 5 gives a score of 0, while a rating of 1 gives a score of 4. As there are 10 questions, this results in a total score from 0 to 40. Optionally, the result can be multiplied by 2.5 to find a score between 0 and 100.

Average scores (not ratings), are shown in Table 5.6 below, so all values are between 0 and 4 and refer to the extent that SAGE possesses a positive, desirable quality related to that question.

Table 5.6 - SUS Survey Results

Question	Average Score (0-4)
I think that I would like to use SAGE frequently.	2.95
I found SAGE unnecessarily complex.	3.47
I thought SAGE was easy to use.	3.53
I think that I would need the support of a technical person to be able to use SAGE.	3.37
I found the various functions in SAGE were well integrated.	3.32
I thought there was too much inconsistency in SAGE.	3.26
I would imagine that most people would learn to use SAGE very quickly.	3.32
I found SAGE very cumbersome to use.	3.42
I felt very confident using SAGE.	3.37
I needed to learn a lot of things before I could get going with SAGE.	3.26

Similarly to the TAM survey, most survey responses fell within a normal sample distribution with few outliers (See Appendix C and D for a question-by-question breakdown of responses). The strongest responses indicated that users found SAGE easy to use, while the weakest response showed that participants were unsure if they would like to use SAGE frequently.

The total score awarded to SAGE was 83.2 (see Table 5.7 and Figure 5.8 for a graphical representation). This shows a high degree of consistency with the TAM survey result.

Table 5.7 - SUS Survey Overall Result

Overall Rating	
SUS Average Rating	33.26 * 2.5 = 83.2

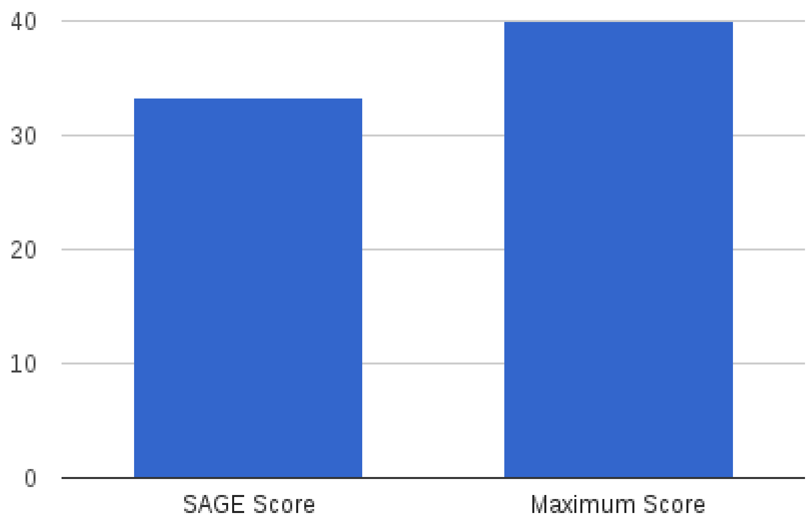


Figure 5.8 - SAGE SUS Score

Following the guide provided by the paper by Bangor, Kortum & Miller (2009), this would give SAGE an "Excellent" descriptive rating for its SUS score of 83.2 -- between the ratings “Good” and “Best Imaginable”, and much higher than the average score of the wide range of websites surveyed in their paper, a score of 68.2.

5.6.4 User Productivity Results

Analysing the 25-image samples annotated by the acceptance testing participants, we can determine a number of interesting differences and similarities between the SAGA and VotePlus algorithms when used as the driving algorithm in SAGE for suggestions. The choice of 25 images per sample provides informative samples, while controlling the amount of effort required to complete each sample.

All three groups from the user acceptance participants contributed valid results to the productivity and annotation evaluations. This means that sufficient information was provided with no significant interruptions which would impact results and raise questions about invalidating certain samples. This process resulted in 17 members annotating 36 samples with SAGA, and 36 samples with VotePlus, providing 72 samples in total.

Measures of each statistic in Table 5.8 are shown on three lines:

mean
min - max
σ = standard deviation

The definition of each measure is below:

Object Count:	The number of objects in each sample.
Concept Count:	The number of concepts in each sample.
Concept Leaf Count:	Concepts linked with one object.
Concept Branch Count:	Concepts linked with more than one object.
Concept Branch Ratio:	Proportion of branches compared to the total.
Annotation Count:	The number of annotations in each sample.
Cluster Count:	The number of annotations in clusters.
Cluster Period:	Time spent in clusters (seconds).
Cluster Rate:	Seconds per annotation in clusters.

Table 5.8 - Annotation Productivity Results

Productivity and Annotation Results (72 samples)		
	SAGA	**VotePlus**
Object Count	25.0 25.0 - 25.0 σ = 0.0	25.0 25.0 - 25.0 σ = 0.0
Concept Count	50.67 32.0 - 115.0 σ = 16.13	56.5 30.0 - 97.0 σ = 16.52
Concept Leaf Count	30.78 14.0 - 84.0 σ = 14.43	34.75 9.0 - 77.0 σ = 15.39

Concept Branch Count	19.61 11.0 - 30.0 σ = 4.61	21.42 11.0 - 35.0 σ = 5.97
Concept Branch Ratio	0.41 0.18 - 0.67 σ = 0.1	0.40 0.21 - 0.71 σ = 0.12
Annotation Count	125.31 68.0 - 228.0 σ = 34.94	130.06 81.0 - 237.0 σ = 37.91
Cluster Count	125.31 68.0 - 228.0 σ = 34.94	130.06 81.0 - 237.0 σ = 37.91
Cluster Period	916.04 seconds 356.59 - 1693.72 σ = 292.14	992.95 seconds 435.04 - 2349.09 σ = 442.55
Cluster Rate	8.82 3.58 - 15.98 σ = 2.83	8.71 3.58 - 14.34 σ = 2.61

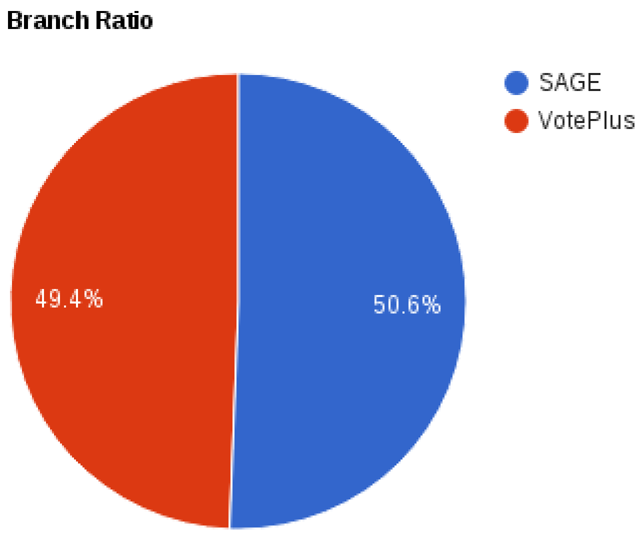


Figure 5.9 - Branch Ratio Comparison

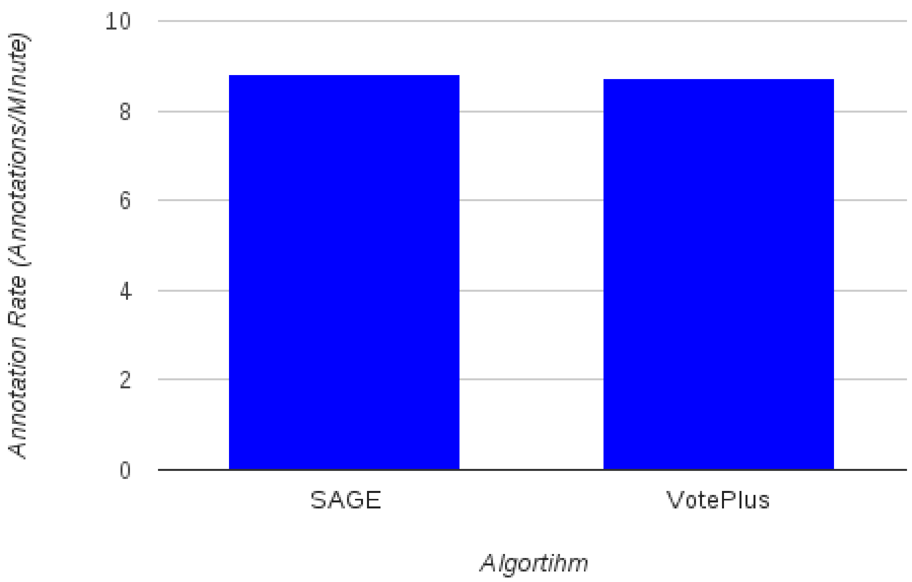


Figure 5.10 - Annotation Rate Comparison

Shown in the results in Table 5.8 and visualised in Figure 5.9 and Figure 5.10, one can see that SAGA leads to slightly better overall productivity in terms of branch ratio and annotations per minute than VotePlus, but not by a substantial amount. Both algorithms can therefore be said to allow similar levels of user productivity in the tests that were conducted.

SAGA and VotePlus both have identical object counts with no standard deviation. This simply indicates that all results correctly contain the expected 25 objects, and no objects were added to or deleted from any sample by the participants. Similarly, the number of annotations falling within clusters is 100% of the annotations added, indicating no major distractions, and high validity.

SAGA has a slightly lower average concept count than VotePlus, and a slightly lower average annotation count. However, the annotation work is performed slightly faster in SAGA, and since less concepts and annotations are being created, SAGA helps to produce an annotated sample noticeably more quickly than VotePlus.

Concepts are distributed equivalently in branches and leaves, as evidenced by the branch ratio in the SAGA and VotePlus results. This shows no major differences in the algorithms.

5.6.5 Suggestion Acceptance Results

The second measure, the suggestion acceptance results, are only available for the second and third groups from the acceptance testing participants (11 individuals, 23 samples per algorithm, 46 samples in total).

After the first group in the controlled volunteer participants completed the annotation exercise, it was discovered that a minimal difference was already apparent between the performance characteristics of the two algorithms. This illustrates the difficulty described by Shani & Gunawardana (2011) of anticipating the kind of data that needs to be collected in a user-based experiment. A new line of enquiry was devised for subsequent groups that aimed to address if a new characteristic, suggestion acceptance, differed between the algorithms.

Acceptance is a measure of how often a participant chose to accept a suggestion rather than annotate a different concept manually. It is analogous to the clickthrough rate typically used to evaluate the effectiveness of online advertising.

Acceptance is shown in quartiles (Q1 is the average acceptance for the first quarter of annotations in the samples, etc.) and QT is the overall average across all quartiles in the samples. A value of 100% would indicate that every annotation was an accepted suggestion, while a value of 0% would indicate that no suggestions were accepted and every annotation was added manually. The difference between the algorithms is shown as a percentage improvement respective to the first algorithm's (i.e. VotePlus's) results. This highlights the gain achieved in SAGA, particularly in the early quartiles where the initial percentages are small (refer to Table 5.9).

Table 5.9 - Acceptance per Quartile

Acceptance Ratios over Time (%)					
	Q1	**Q2**	**Q3**	**Q4**	**QT**
VotePlus	13%	26%	33%	36%	27%
SAGA	24%	32%	34%	47%	34%
Improvement	+85%	+23%	+3%	+31%	+26%

SAGA is consistently more frequently accepted than VotePlus in all quartiles, with almost 50% of suggestions being accepted by users in Q4 (see Table 5.9). Importantly, SAGA obtains a very high improvement in suggestion acceptance in Q1, which shows that the measures taken to address the cold start issue have had a positive effect. Both algorithms have similar acceptance in Q3 before SAGA regains a lead in Q4.

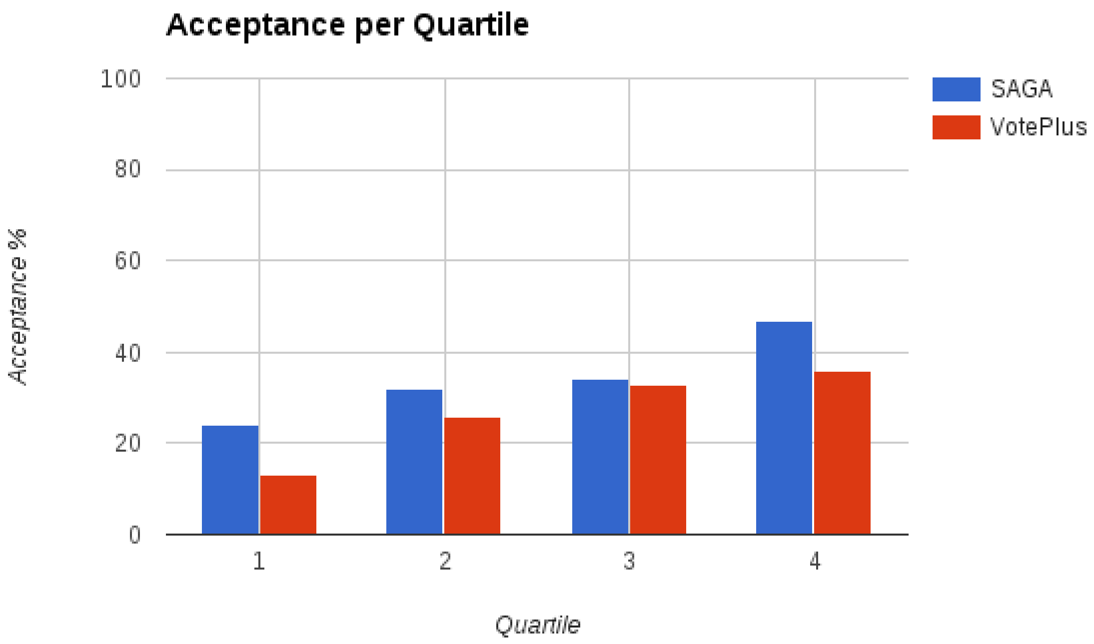


Figure 5.11 - Acceptance per Quartile Chart

Figure 5.11 shows the change in acceptance rate as more annotations are added. Neither algorithm can create new concepts without the intervention of the user, thus we begin at the very start of Q1 with a 0% acceptance rate since there are no suggestions that can initially be made. As more annotations are added, we see a sharp rise in acceptance to a grand total of 47% for the SAGA algorithm and 36% for VotePlus in Q4.

While these results cover the initial 25 objects to be annotated, we can make the general observation that as more objects are annotated, a higher proportion of annotations are accepted over this range. The degree of this effect as seen in the test data shows that a substantial amount of user support can be provided by using algorithms to provide annotation suggestions to users.

5.7 Discussion

In this chapter, we analysed how users interacted with the suggestions provided by the SAGA and VotePlus algorithms in SAGE. Surveys were conducted to elicit the participant's explicit acceptance of SAGA in the SAGE user environment, then acceptance testing was conducted to capture the participant's implicit acceptance of the suggestions provided by SAGA and VotePlus algorithms in SAGE.

The result of the surveys was very favourable overall. The surveys show that participants strongly felt that SAGE was relatively easy to learn and use, and that SAGE was useful with assisting in the annotation process. However, participants were slightly less certain that SAGE would help them to accomplish tasks more quickly. The results obtained in the surveys can be generalised, but it should be noted that the surveys did not evaluate how easily SAGE could be used in conjunction with other software and practices commonly used in digital heritage, and that no user in the survey had special user interface requirements (such as a visual disability) which may impact real-world usability.

The results of the first part of acceptance testing, user performance, showed that both the SAGA and VotePlus algorithm led to a similar level of user productivity, with a similar set of annotations being produced in each case. Sample projects that used SAGE were finished slightly faster, but with slightly fewer annotations than sample projects that used VotePlus.

The second part of acceptance testing, suggestion acceptance, showed that both algorithms gradually increased in suggestion acceptance rates as more annotations were added. However, SAGA's suggestion acceptance was consistently higher and grew more rapidly than VotePlus, showing that the enhancements the SAGA algorithm provides had a positive influence on its acceptance rate in the digital heritage sample projects.

Essentially, this means that both algorithms yield a similar number of annotations in a given time period, but SAGA requires less manual input to achieve the same result. For a suggestion mechanism like SAGA, a high rate of acceptance means that users enjoy the full benefits provided by a suggestion-driven workflow, including easier annotation of larger collections and a high degree of internal consistency in the project's annotations. It also means that SAGA can be used as part of other user-supporting services, such as searching concepts, viewing related groups of objects, or merging similar concepts.

The results obtained from these experiments indicate that the rate of acceptance of suggestions increases as more concepts are annotated to objects in a project. This means that once key concepts have been established, suggestion mechanisms such as SAGA actually become increasingly useful. While the acceptance testing projects were limited to samples of 25 images to conserve the amount of effort asked from participants to complete each sample, the suggestion acceptance of both algorithms had not plateaued by the conclusion of the sample (See Figure 5.11). This suggests that an extended sample size with additional objects may achieve higher acceptance results than those seen in these samples.

While this experiment showed a good outcome for SAGA and SAGE, the absence of a considerable user productivity improvement stemming from the use of SAGA means that both SAGA and existing suggestion methods remain limited by the amount of work that a given user can perform in a given timeframe. This means that for extremely large collections, either an automated approach or a social computing approach (which SAGE provides) must be used to handle the large number of objects in those projects.

This naturally leads to the activity of identifying large-scale collections which have significant value to a wide number of people, all of whom can contribute general observations and perceptions, and some of whom can provide a deeper level of insight by identifying specific semantic details using their knowledge of a domain to capture information that may not be readily apparent to the casual observer. Of these applications (and there are many, just consider fields such as tourism, commerce, art, history and literature as wellsprings of ideas), we have chosen to investigate digital heritage.

Chapter 6: Application to Digital Heritage

6.1 Overview

One of the major challenges encountered while a semantic network is in the process of being actively developed is the initial sparsity of connections between individual data objects and the semantic concepts which describe their significance (Bonchev & Buck, 2005). This limits the effectiveness of techniques that leverage these connections in the semantic network as a source of information upon which to base suggestions to assist the annotation process.

This is particularly important for fields such as digital heritage, where a large quantity of diverse digital objects require annotation by the limited number of experts and volunteers who are available to perform this task, and where a content-based image retrieval approach is challenging due to the semantic gap between the features of the image and the image's historical and cultural significance (Bloehdorn *et al.*, 2005).

In the previous two chapters, we identified that with the broad applicability and ease of use of social tagging systems (Morrison, 2008), researchers have the opportunity to capture crowdsourced annotations that contribute additional knowledge about the collections being explored by users of the system. Modelling these annotations as a semantic network, with objects linked by concepts, presents a way to capitalise on what has been discovered to infer additional information about the materials represented within the semantic network.

This thesis was largely motivated by the work conducted at the UNE Heritage Centre (University of New England, n.d.) and the challenges identified in effectively exploring the vast collections of heritage material stored there. This chapter aims to determine whether the approach developed and evaluated in this PhD thesis, SAGA, is effective in a case study application where volunteers at the UNE Heritage Centre use SAGE as a utility to produce annotations for digital heritage research projects.

6.2 Related Work

Key related works in this section explore the types of information systems employed in digital heritage projects, and whose motivations can be compared to the one behind this PhD project. Recent works suggest that digital heritage is increasingly open to experimenting with and adopting emerging information technologies, and would potentially stand to benefit from the technologies behind SAGE.

Digital heritage can be divided into two related subfields: Digital Archives and Digital Libraries (Lynch, 2002; Borgman, 1999). While the two employ similar technologies and experts, they are motivated by different objectives. Digital archives are motivated by collecting and preserving heritage materials, while digital libraries are concerned with making these materials accessible and of interest to the public through software and websites, acting like an electronic equivalent to libraries, galleries, museums or displays in the physical world.

6.2.1 Digital Libraries

There is an emerging trend in the field of digital library management (DLM) for information systems to be more user-centric, more interactive and supportive of active communication and collaboration. DLM systems also tend to be general-purpose rather than specialising in a particular format of data, allowing for different multimedia formats to be mixed to create vivid presentations that include text, images, and potentially even videos or interactive computer models (Agosti & Ferro, 2007).

Recent approaches that seek to provide rich experiences to visitors include interactive displays that represent physical artefacts as 3D models, which users can interact with in a virtual environment (Guarnieri, Pirotti & Vettore, 2010; Koller, Frischer & Humphreys, 2009). Some technologies allow 2D visitor photographs to be superimposed on a larger 3D model, blending photographic and virtual mediums (Stefani, Busayarat, Lombardo, Luca, & Véron, 2013), and others allow for interaction with these models in a virtual reality setting (Carrozzino & Bergamasco, 2010).

A deeper appreciation of historical locations can be fostered through electronic guidebooks for smart devices, which display information relevant to where the user is travelling (Chianese, Marulli, Piccialli & Valente, 2013). Alternatively, when the user doesn't have a destination in mind, context-aware tour guide systems can recommend that users visit historical points of interest in a city or countryside while factoring in travel conditions such as traffic or the weather that might make some destinations preferable to others (Bartolini *et al*., 2013; Bartolini *et al*., 2014).

Serious games allow for participants to learn about heritage by interacting with a game environment, which can also present an opportunity to gather new knowledge from the participant's interactions (Mortara *et al.*, 2014). In some special cases, such as when exploring the cultural heritage relating to video games, playing the game allows the participant to experience the cultural phenomenon first-hand (Suominen & Sivula, 2013).

These examples demonstrate that practitioners of DLM are enthusiastic to explore new and emerging forms of interaction as a means of increasing public engagement with heritage collections and historic sites through the use of information technologies (Cameron & Kenderdine, 2007). This is an encouraging trend for works such as this PhD project, which seek ways in which the interaction and contribution of participants can be used to enrich the information held on heritage materials.

These emerging approaches in DLM software can enhance the experience of users by capitalising on an aspect of similarity between pieces of content to enable the automatic formation of collections, which can allow effective sharing of additional metadata, and also to help with detecting seminal content in a collection. This is achieved by allowing users to explicitly mark content as similar in some regard, or in an implicit manner by analysing the corpus of the content stored in the DLM software (Aletras, Stevenson & Clough, 2012).

Public annotations provide a means of collaboration and communication between users, while private annotations provide a means for an individual user to express their workflow to the DLM software, such as how recommender systems can use items a user has previously sought out and interacted with as the basis for new recommendations of similar materials. Either of these forms of annotation allows additional information to be captured by the information system (Agosti & Ferro, 2007).

Further analysis of the relationships between collections and the documents within each collection can be performed using network analysis techniques which were popularised by a surge of interest in social networking (Rae, Sigurbjörnsson & van Zwol, 2010). For example, notable figures or events which occur in multiple heritage collections can be used as a semantic link between them, allowing visitors to pursue a common element across collections.

By having multiple levels of analysis, both document-oriented and collection-oriented, different perspectives of the heritage materials can be pursued as a source of potential discoveries (Oinas-Kukkonen, Lyytinen & Yoo, 2010). This highlights the importance of being able to detect when materials are semantically related to one another using approaches like SAGA, even if simply to offer a “See Also” section to enable different avenues of enquiry to be pursued by viewers.

6.2.2 Digital Archives

The importance of effective information retrieval has been emphasised in digital archives, and a number of important research objectives have been identified. These include engaging a community in order to capture and communicate knowledge, exploring relationships between entities in collections to infer additional information, and finding effective techniques for retrieving information that accommodates varied information needs (Lawless, Agosti, Conlan & Clough, 2013; Webb & Canberra National Library of Australia, 2003; Mudgea, Ashleyb & Schroer, 2007).

Information technology can assist with projects which must be carried out in an expeditious manner due to their sensitivity to the passage of time. In Australia, this includes the collection and preservation of Aboriginal culture and history in areas where there is a risk of losing information that would normally be passed on in the form of oral traditions. Recent projects have sought to develop regional Indigenous databases of language, culture and history (IKRMNA Project, 2006), to find and conserve artefacts such as photographs, film footage and audio recordings of native communities (Ara Irititja Project, 2011), and to record the significance of traditional travel paths and landmarks (Australian National University, 2013). These projects exhibit diverse information requirements, and require considerable contributions of time and resources to successfully carry out.

Studies in the area of community crowdsourcing have investigated quantifying the amount of effort required for volunteers to contribute annotations, as well as approaches for reducing this cost effort (Villa & Halvey, 2013; Vijayanarasimham & Grauman, 2009). The needs of experts and students in crowdsourcing differ, with experts seeking software which acts as a tool in their established workflow and preserves the structure of their meticulously-formatted metadata, while students and volunteers seek software that helps to build their understanding of a field as quickly and easily as possible (Sweetnam *et al*., 2012). One way of reducing the cost of acquiring annotations is to seek sources where they may already exist, such as linking information from Wikipedia (Fernando & Stevenson, 2012).

Topics such as social event detection (Sakaki, Okazaki & Matsuo, 2010; Yang, Pierce & Carboneli, 1998), image emotion detection (Zhao *et al*., 2014) and bridging the affective gap (Cambria, Havasi & Hussain, 2012; Baccianella, Esuli & Sebastiani, 2010) are more easily performed by people than computers, and are prime targets for applying techniques such as computer-assisted annotation to augment crowdsourcing (Kamar, Hacker & Horvitz, 2012).

An example of where bridging the affective gap affects information retrieval is in navigating art collections. In this case, both ontology-based (Isemann & Ahmad, 2014) and folksonomy-based (Semeraro, Lops, De Gemmis, Musto & Narducci, 2012) approaches for exploring collections have shown great promise for making artworks more easily accessible by interested viewers who may not be familiar with the works and artists they are investigating.

Neighbour voting algorithms utilising the relationships between images, tags and the content of images have been shown to be effective for retrieving relevant results (Li, Snoek & Worring, 2009). However, some research shows that utilising contextual information present in these relationships can actually have an adverse effect on results compared to a purely content-based approach (Truong, Sun & Bhowmick, 2012), while other research indicates that using annotations in conjunction with structured metadata leads to better information retrieval results than using either by itself (Aletras, Stevenson & Clough, 2012).

Conversely, the information needs of a digital heritage researcher might not be adequately met by retrieving a list of visually similar images as suggestions for annotation; rather, the significance of the images is of greater importance, and presenting opportunities for serendipitous connections between images and tags is a desirable characteristic for heritage information retrieval systems (Quan-Haase & Martin, 2012). This characteristic can be promoted by seeking semantically-linked entries for annotation suggestions to increase the chance of a fortunate discovery, such as finding two dissimilar images which share a subject, location or time, which would make one image of interest to a researcher who is studying the other (Quan-Haase & Martin, 2012).

6.3 Challenges

While the number of annotations produced for a specific project gives an estimate of the quantity of work performed on it, it doesn't provide an indication of how much work is needed to comprehensively annotate the project in total. Two challenges, relating to the interconnectedness and distribution of annotations, affect whether the project can be considered well-annotated from a data perspective (Bonchev & Buck, 2005; Zhu & Wu, 2009; Guy & Tonkin, 2006).

6.3.1 Annotation Subgraphs

When using an uncontrolled ontology, like in a "folksonomy" system where concepts are not required to follow any particular conventions when created by users, there are a wide range of circumstances that can result in two or more concepts being created that actually describe the same semantic idea (Synonymy), or where a concept ambiguously describes two distinct semantic ideas (Polysemy) (Marchetti, Tesconi, Ronzano, Rosella & Minutoli, 2007).

Consider the example of a concept described as "Bank". This could be a financial institution, a structural part of a river, a reserve power supply, or a collection of power points. This can result in multiple unrelated objects being closely linked via a shared concept, a polysemous concept. Polysemous concepts may eventually need to be disambiguated to distinguish their semantic meaning.

Synonym concepts can be introduced by simply using two words with similar meanings for single-word concepts (e.g. “baggage” and “luggage”) or through different word choices or ordering of multi-word concepts (e.g. “mountain images” or “images of mountains”). Synonym concepts can further be introduced by spelling errors and by using different words from the same lexeme, including pluralisation (e.g. person, people), and tenses (e.g. ran, running).

When multiple sources are used, the annotation style or focus of one source can be different to the style or focus of another, such as mixing an ontologically-structured set of annotations with a folksonomy of annotations. Differences in language or use of language (e.g. slang terms) can also lead to more synonym concepts being created.

Different users may also have different perceptions of the concepts which are important in an image. This can include different priorities for which concepts are recorded, such as when one participant is interested in the people present in images and another is interested in locations, or a potential disagreement of which concept should be annotated, such as when a participant misidentifies a person in an image.

Finally, when multiple users work together on a heritage project which contains a large number of complex concepts, additional mental effort is required on the part of new users to read the list of existing concepts and recognise when one of the concepts created by another annotator is relevant to the image that they are viewing. For particularly large projects where work is being conducted in parallel, users may not even be aware that new concepts have been created which they have not yet seen, but are relevant to the images they are viewing. This can lead to users creating synonym concepts.

In each of the above cases, the issue of synonym concepts can lead to an individual annotating a disconnected subgraph of objects, where none of the concepts provided by the annotator link with existing work in the project, and thus provide no means for a purely graph-based suggestion mechanism to traverse from the main subgraph of annotations to the new subgraph.

The annotation subgraph problem is illustrated in Figure 6.1. While the network formed between objects and concepts allows both the “flower” and “fabric” objects to be visited through tag co-occurrence, the objects annotated with the “trees” concept are completely separate, and exist in a secondary, smaller subgraph. This either requires a new concept such as “nature” to be created that would link the objects either directly or indirectly, or for suggestion algorithms to use a subgraph-spanning technique of finding suggestions that can traverse the divide between subgraphs.

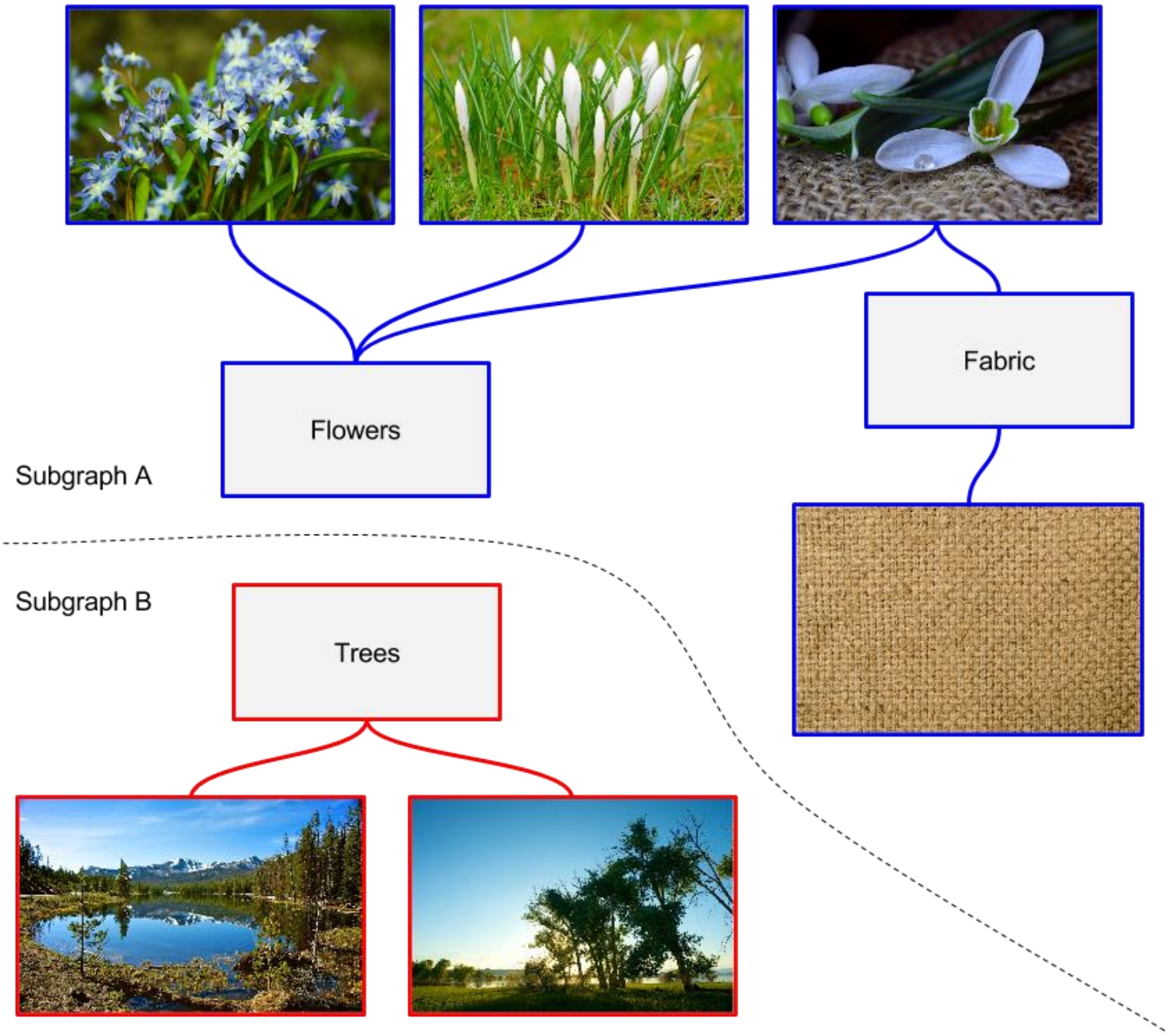


Figure 6.1 - Illustration of Annotation Subgraphs

The annotation subgraph problem makes it highly desirable to allow users to predominantly annotate a project using suggestions provided by a computer algorithm, which is able to maintain an overall perspective of the available concepts. This leads to greater consistency in annotations, and also to make it easy for users to create their own concepts ad hoc and later reconcile synonyms with the assistance of the computer system.

6.3.2 Superficial Annotations

When examining the set of annotations produced in the annotation process, we need to determine whether these annotation sets are comprehensive or superficial. A comprehensive set of annotations will provide plenty of data for an annotation-based suggestion algorithm, while a superficial set of annotations might lead to poor suggestion results.

A superficial set of annotations might result from a small set of concepts being linked with the majority of objects in a collection, providing little to distinguish one object from another, or alternatively, only a small number of annotations might be provided for each object, which again only superficially describes those objects. Both of these situations would limit the usefulness of that annotation set for producing suggestions, and would also impact any other annotation-based service, such as searching.

Another situation that could arise is that a diverse set of concepts could be contributed to a project, with each object having little or no overlap in concepts. This could easily occur if every object was annotated a unique, detailed description and nothing else. While these descriptive concepts would capture the object's' significance perfectly, and probably wouldn't bring to mind the idea of the annotations being "superficial", the annotation set would be less suitable for algorithmic use than a well-linked network of concepts.

Either of these scenarios is potentially likely to occur in a heritage collection, and will have an impact on the usefulness of the annotations as training material for computer software that seeks to automate aspects of the annotation process.

The first scenario can be caused by the creation of a large number of generic concepts such as "landscape" or "people", which while useful to narrow down the collection into broad categories, does not provide the level of insight desired from the annotation process. When applied to a large collection of images where little time is available to consider each image, this is likely to result in a number of generic annotations assigned to each image.

In the second scenario, consider a hypothetical spreadsheet file being used to store annotations, with each object having one row and a "Description" field as part of that row. This leads to an individual description being provided for every single image with little chance of overlap.

The ideal set of annotations would instead contain a modest number of annotations for each image, with some of these annotations being broad in scope and linking together many images into similar groupings, and others being quite specific in scope to accurately describe the image being considered. This would be ideal both for suggestion mechanisms to provide good recommendations to users, and would also make the annotations informative for viewers.

6.4 Suggestion-Acceptance Approach

The issues relating to the annotation subgraph problem have been addressed in this PhD research via a prevention and control strategy that was designed and implemented in SAGE. The first seeks to prevent the annotation subgraph problem from occurring in the first place by promoting the use of already established concepts when annotating each image. The second seeks to lessen the impact of discrete subgraphs by providing two methods in the similarity mechanism for bridging subgraphs, without requiring any user intervention.

6.4.1 Prevention

By promoting suggestions as a quick and easy way to annotate images, SAGE encourages users to enrich the existing semantic network by expanding it with new objects, rather than establishing a separate semantic network within a project that has a local selection of concepts and objects.

For example, if a user adds a new object to a project, they will be immediately presented with suggestions for existing concepts that may be relevant to this new object. While they may add many concepts of their own, accepting even a single suggestion for a concept that has already been established is enough to bridge the separate subgraphs and allow object-concept links to be followed by suggestion algorithms to access the complete set of objects.

Establishing links with the main network further encourages additional bridging annotations to be suggested, as concepts from that network will be judged as more relevant when more connections are made with it. These bridge-forming suggestions work both ways; participants contributing to the established parts of the semantic network will begin to see the new objects and concepts appear as suggestions as well, allowing these bridging annotations to be made by either party.

For instance, suppose the established parts of the semantic network deal with historic buildings in a city, with concepts like “Building”. A user introduces a new concept, “Restoration Building”, to annotate the restored heritage buildings they are interested in, and have been adding to the project. This user will see “Building” appear as a suggestion to their objects, while users in the established parts of the semantic network will see “Restoration Building” as a high-ranking suggestion in theirs. If either of the users accept the concept that has been suggested, the separate work will become connected, and the main group of users will see the restored heritage building objects added by the solitary user, while that solitary user will see the historic building objects.

6.4.2 Control

SAGA establishes "bridging" suggestions using two methods that avoid the constraints of a purely tag co-occurrence approach for finding suggestions. These approaches are using popularity and using text similarity.

The first method used, popularity, examines what the most popular concepts in the project are and suggests these. As all objects in a single project are intrinsically linked in some way (as collections are often from the same source, photographer, locality or some other factor), it is likely that the most popular concepts could be relevant to most, or all, of the objects being annotated.

The second method, text similarity, is achieved by examining the text descriptions provided for each concept. As concepts in SAGE can range from phrases to full descriptions, there is a chance that certain descriptive keywords will appear in multiple concepts, meaning the concepts are lexically similar to one another. When searching for suggestions, the influence in neighbour voting is allowed to propagate through lexical similarity wherever it occurs in the same way that influence propagates through tag co-occurrence. This allows for speculative bridging to occur between concepts, which means that concepts which aren't usually accessible through tag co-occurrence can be found and presented as suggestions.

The acceptance of suggestions provided by a suggestion mechanism helps to link projects consisting of isolated subgraphs, and promotes each object to have a multitude of concept annotations. Taken to an extreme, however, heedless acceptance of suggestions would result in a more or less homogenous set of annotations given to each object -- the other issue that impacts on annotation complexity.

Fortunately, as human participants are part of the suggestion-acceptance feedback cycle, there is no risk of the suggestions being automatically accepted until this scenario eventuates. When a suggestion does not match the image, the user simply ignores those suggestions until it becomes difficult to find relevant suggestions, then moves on to the next object. As human participants are also able to perceive and comprehend very specific details about the image or the image's context, they are also able to contribute highly specific concepts which are unique to a small set of images.

6.5 Methodology

This section describes the evaluations performed on a group of heritage projects to determine if they possess a number of desirable characteristics for SAGE datasets. We examine metrics which can be used to evaluate both crowdsourced and heritage datasets, contrasting annotations performed in an experimental context with those from a real-world context.

6.5.1 Saumarez Combined Dataset

The Saumarez Collection is a collection of several thousand heritage images from the New England, NSW and beyond, taken by members and friends of the White family.

The Saumarez Combined dataset is the aggregated concepts and annotations provided by participants in the user acceptance experiment in this PhD project, which used images from the Saumarez Collection (see Figure 6.2 for some examples).

Figure 6.2 - Saumarez Collection Example Images

The summarised statistics of this project can be seen in Table 6.1. This is a very large collection with a relatively high number of annotations per object, courtesy of approximately 60 cumulative hours of focused annotation work performed by the acceptance testing participants.

Table 6.1 - Saumarez Combined Project Statistics

Saumarez Combined Statistics	
Objects	1411
Concepts	1028
Annotations	8527
Average Object Annotations	6.04 (0-22, σ: 3.12)
Average Concept Annotations	8.29 (0-611, σ: 33.38)

To obtain a set of annotations for the images in this collection, a group of volunteers were assigned randomised image sets containing 25 unseen images per set to annotate independently of one another using concepts of their own choosing. This was conducted using the annotator role in SAGE. The number of sets assigned to each volunteer varied according to the time the volunteer was free to contribute, with most volunteers contributing between two and eight sets through repeatedly redeeming the annotator key to receive additional sets.

The image selection strategy ensured that images were never assigned to the same user twice, and the independent work maximised the diversity of concepts annotated to the images by the volunteers. Since they were not influenced by the contributions of others, volunteers were encouraged to create new concepts that suited the images well, rather than re-using something that already existed, but was less suitable for those images.

The independent sets were pooled together into a single collection, resulting in links between sets forming when the concept was independently identified by two or more volunteers, was written the same way, and thus resulted in a concept collision. SAGE automatically flattens these collisions into a single concept while preserving all unique annotations, allowing traversal through this concept to objects in both sets.

This provided 1,411 unique images from the original collection of 3,168 images with annotations using 1,800 annotation requests across 72 sets (i.e. 25 images per set in 72 sets mean that the participants as a group were asked to annotate 1,800 images). These project statistics are shown in Table 6.1.

In the combined collection, 16 objects and 14 concepts do not have annotations. The concepts are spelling errors that were removed rather than edited to correct the mistake, and the objects were missed by the annotators. Neither of these are significant errors, and are a realistic mistake that can occur in SAGE collections.

Some images were also assigned more than once to separate annotators. The annotator feature of SAGE does not seek to prevent this, as allowing multiple users to annotate the same image promotes a well-linked network while providing multiple perspectives on an image's significance.

Tag co-occurrence occurs independently of whitespace, capitalisation and punctuation, but is sensitive to human error, synonyms, variances in expression, spelling mistakes and variances in lexemes, such as different word tenses and pluralisation. Rather than performing an additional step of data cleaning and normalisation to artificially remove these inconsistencies, they were preserved, as they are better representative of ontology-free annotations in the application domain than an artificially cleaned and modified dataset.

6.5.2 Heritage Project Datasets

The UNE Heritage Centre are conducting ongoing digitisation projects, where old photographs, glass plates and slides are being scanned or digitally photographed and stored digitally. This digitisation effort has produced collections which, often for the first time, can be processed using digital tools and technologies.

Over the course of the last year, a number of digital photo projects have been added to SAGE by members of the UNE Heritage Centre, and annotation work has been conducted on each. Each project has a unique theme and historic significance.

The annotation work was conducted using SAGE via a browser from the UNE Heritage Centre or from the participant's homes. This process primarily used object views when contributing information, and concept views when retrieving information.

For instance, in Figure 6.3, we see an example object which features an aerial photograph of Armidale's Central Park. The participants have identified the location and photographic style of the photograph, and have been presented with further concept suggestions including "Building", the names of a number of notable buildings from Armidale, as well as a couple of street names. This allows an object to be annotated comprehensively and in depth, depending on the participant's insight into what the object relates to.

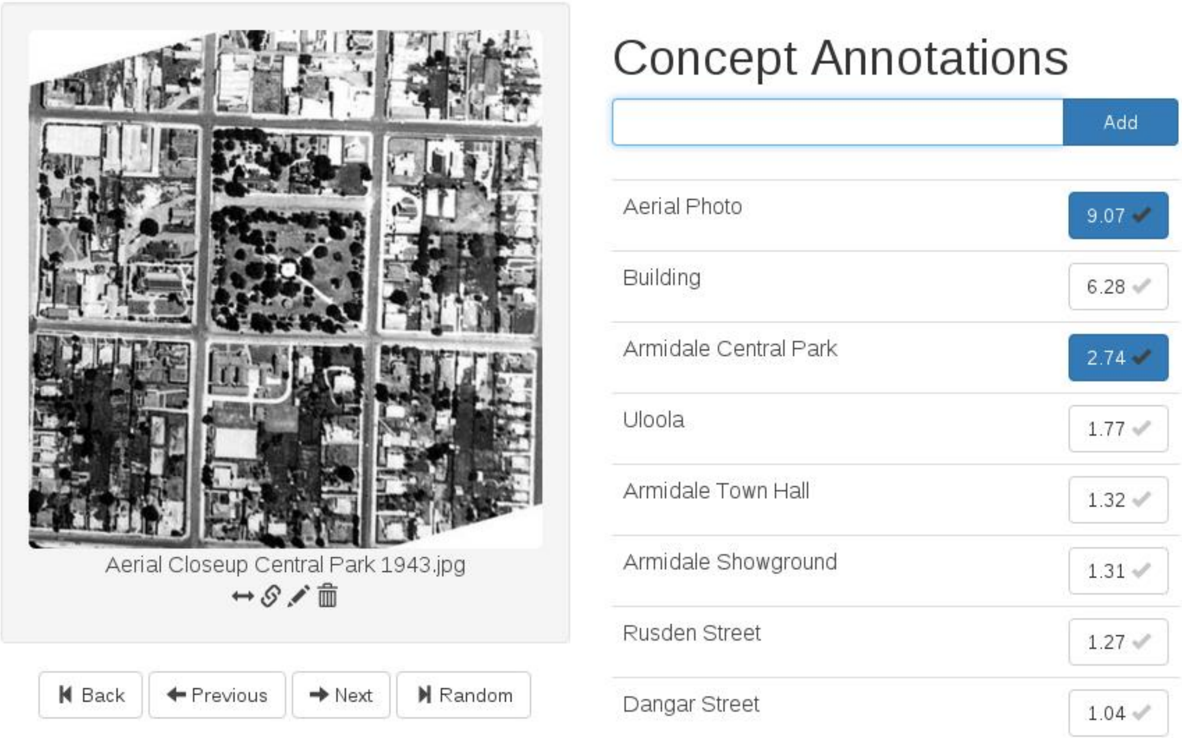


Figure 6.3 - Example Object featuring Armidale Central Park

Following the link provided to the concept "Armidale Central Park", as depicted in Figure 6.4, we are shown the two identified photographs of Central Park as well as additional aerial photographs of buildings that SAGE has identified as high-similarity matches to the aerial photographs of Central Park, which might also feature this location. This provides a view where confirmed and similar objects can be collected together when seeking to retrieve information from the application when searching or browsing by a specific concept.

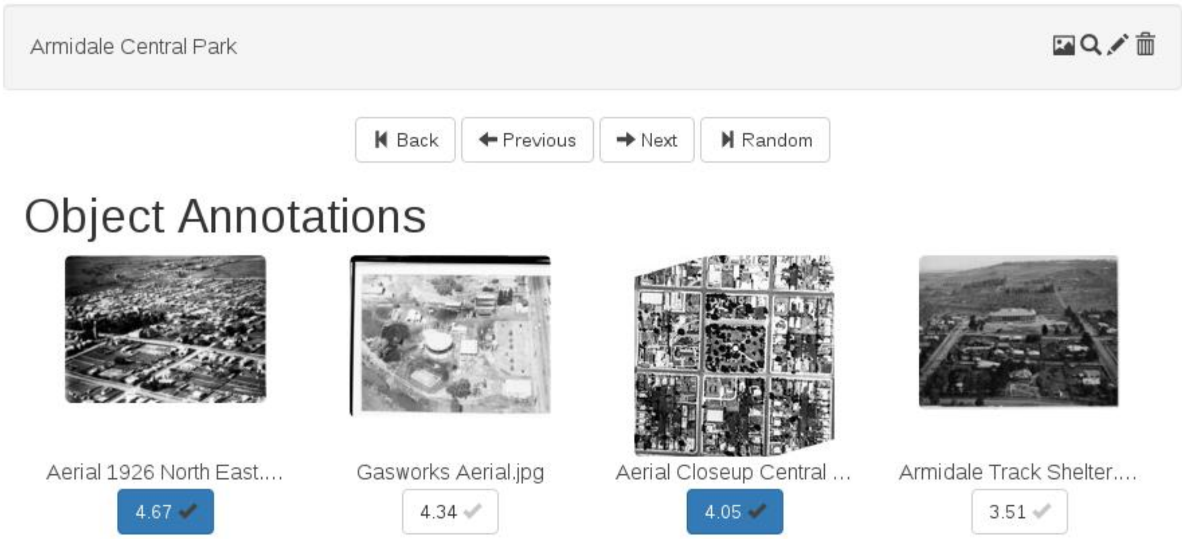


Figure 6.4 - Example Concept featuring Armidale Central Park

With this usage pattern in mind, the following are short descriptions of the individual heritage projects that have been conducted, or are in process of being conducted, using SAGE.

Armidale in Photos (85 Photos)

Armidale in Photos features a number of photographs of historic Armidale taken by land and air. These images include historic sites around the city that have been modified, destroyed or demolished since the photographs were taken, and provide a valuable insight into the architectural history of Armidale (see Figure 6.5 for some examples).

Figure 6.5 - Armidale in Photos Example Images

The summarised statistics of this project can be seen in Table 6.2. This is a relatively small collection in the early phases of annotation.

Table 6.2 - Armidale in Photos Project Statistics

Armidale in Photos Statistics	
Objects	85
Concepts	40
Annotations	101
Average Object Annotations	1.19 (0.0-7.0, σ: 1.48)
Average Concept Annotations	2.53 (0.0-22.0, σ: 3.95)

Nursing VAD Identification (70 Photos)

The Nursing VAD Identification collection features a series of photos of Volunteer Aid Detachment members assisting the recovery of wounded soldiers in the New England. This effort was based around Booloominbah, a large family home on the outskirts of Armidale in New South Wales.

Photos show Booloominbah and surrounds while it was being used as a convalescent home during the First World War. This collection includes photographs of the wounded soldiers and nurses who tended to them, along with some of the sporting and recreational activities they passed the time with (see Figure 6.6 for some examples).

Figure 6.6 - Nursing VAD Identification Example Images

The summarised statistics of this project can be seen in Table 6.3. This is a relatively small project in the early stages of annotation.

Table 6.3 - Nursing VAD Identification Project Statistics

Nursing VAD Identification Statistics	
Objects	70
Concepts	42
Annotations	122
Average Object Annotations	1.74 (0.0-19.0, σ: 2.53)
Average Concept Annotations	2.9 (0.0-35.0, σ: 5.48)

Nickson AIF Photos (415 Photos)

The Nickson AIF Photos are a large collection of World War II photographs taken by Wilfrid Lievesley Nickson, a war veteran and photographer. Along with the Nursing VAD Identification photographs, this collection has particular significance at the moment in the historic community due to the current and upcoming centennial anniversaries of the World Wars (see Figure 6.7 for some examples).

Figure 6.7 - Nickson AIF Example Images

The summarised statistics of this project can be seen in Table 6.4. This is a modestly large collection with a fair number of annotations, and was the featured collection as part of a crowdsourcing exercise in the user acceptance evaluation training.

Table 6.4 - Nickson AIF Project Statistics

Nickson AIF Statistics	
Objects	415
Concepts	210
Annotations	876
Average Object Annotations	2.11 (0.0-23.0, σ: 3.74)
Average Concept Annotations	4.17 (0.0-81.0, σ: 8.41)

Saumarez Gardens (521 Photos)

Saumarez House is a house and estates donated by the White family to the National Trust of Australia (National Trust, n.d.). The house was the centerpiece of a pastoral run dating back to the earliest European settlement in the New England.

The gardens and immediate landscape of Saumarez are the subject of ongoing preservation work, given the high level of maintenance required. A photographic history of the gardens is available in the photos taken by the White family during the establishment of Saumarez, referred to as the Saumarez Gardens collection (see Figure 6.8 for some examples).

Figure 6.8 - Saumarez Gardens Example Images

The summarised statistics of this project can be seen in Table 6.5. This is a large collection with a fair number of annotations.

Table 6.5 - Saumarez Gardens Project Statistics

Saumarez Garden Statistics	
Objects	521
Concepts	64
Annotations	1362
Average Object Annotations	2.61 (0.0-7.0, σ: 1.71)
Average Concept Annotations	21.28 (0.0-268.0, σ: 45.63)

6.5.3 Complexity Evaluations

Up until now, we have worked with datasets that have been produced outside of the processes normally employed in digital heritage research. The evaluations conducted in this chapter intend to determine if heritage projects carried out in SAGE develop characteristics which are favourable to the approach taken, or whether projects have been annotated in a manner which is incompatible with a tag co-occurrence based approach.

To evaluate the annotations produced within each of these heritage projects, we begin by identifying a suitable test statistic upon which to base this evaluation. As this evaluation is not a common, standardised type of assessment in digital heritage literature, best-practice methods are not available from other research papers which could be used in this project.

The critical aspect being studied are the sets of annotations which have been produced for the projects. These annotations are the information that would be used during further heritage work in the projects, as the basis for additional suggestions from suggestion mechanisms, and they would also be the ground truth with which an automated content-based information retrieval algorithm could be trained, so a graph-based analysis of their quality is desirable.

We can perform two types of graph-based analysis on the annotations (Bonchev & Buck, 2005):

1. Analysis of the graph as a whole, measuring completeness.
2. Analysis of local areas of the graph, measuring connectedness.

We first analyse each project to determine the total number of objects that have been annotated, and the mean number of annotations per annotated object. This provides a global indication of the completeness of annotation work that has been performed in a project, and so casts further analysis in the proper context.

$$\textit{Annotated Objects} = \textit{Objects with} >= 1\ \textit{Annotation} \quad \textbf{(6.1)}$$

$$\textit{Mean Annotations} = \textit{Count of Annotations} / \textit{Count of Annotated Objects} \quad \textbf{(6.2)}$$

Providing examples, a project with 80% of objects being annotated with an average of 4-8 annotations per annotated object could be subjectively considered reasonably well annotated. A project with 80% of objects being annotated with an average of 1-2 annotations might be considered poorly annotated due to the low depth of annotation. Finally, a project with 20% of objects being annotated with 4-8 annotations might be considered poorly annotated due to poor breadth of annotations.

Note that these descriptions or being “well annotated” or “poorly annotated” are subjective, and these measurements are intended to be descriptive of the annotation work that has been performed. The following evaluations, which focus on the connectedness of the annotations, measure concrete, desirable characteristics for tag co-occurrence based suggestion measures.

We can measure the connectedness of the graph using branch ratios, which indicates the proportion of objects that are well-connected in the graph versus the number of objects which have only a single connection to the graph. Graphs with high branch ratios provide more information for tag co-occurrence based suggestion mechanisms to traverse, and are therefore desirable.

$$\textit{Branch} = \textit{Object with} > 1\ \textit{Annotation} \quad \textbf{(6.3)}$$

$$\textit{Branch Ratio} = \textit{Branches} / \textit{Count of Annotated Objects} \quad \textbf{(6.4)}$$

We can also measure the connectedness of the graph using subgraph counts, which analyse the characteristics of the graph that emerge from annotations between objects and concepts. A subgraph is every object that is accessible by following object-to-concept and concept-to-object links from a starting object. Each discrete subgraph poses a challenge for tag co-occurrence based similarity measures, as a different technique (such as text similarity and popularity, used in SAGA) is then required to reach concepts and objects in another subgraph.

By calculating the number of discrete subgraphs formed by annotated objects, we can determine whether the project is completely connected. This process utilises the property that a set contains only unique items, and ignores duplicates added to it. For example:

$$set(a,b,c) + set(c,d,e) = set(a,b,c,d,e)$$

The subgraph count can be found using the following pseudocode, with the function (1) Find_Subgraph_Count acting as the main driver and accumulator of subgraph counts, and the function (2) Find_Subgraph(Object) being the function responsible for finding the subgraph that the given Object falls within.

(1) Find_Subgraph_Count

Define All_Objects as the set of all objects in the project.

Define Subgraph_Count as 0.

While All_Objects is not empty:

  Define First as the first object in All_Objects.

  Define Subgraph as the set of all objects connected to First (2).

  Redefine All_Objects to be All_Objects - Subgraph.

  Increment Subgraph_Count.

End While.

Return Subgraph_Count.

(2) Find_Subgraph(Object)

Define Found as a set containing the object passed to this method.

Define Count as 0.

While Count < Found's size:

  Define Focus as the object at Found[Count].

  Define Connected as all objects that share a concept with Focus.

  Redefine Found as Found + Connected.

  Increment Count.

End While.

Return Found.

Measuring the completeness and connectedness of the annotation graphs of heritage projects will demonstrate if using SAGE at the UNE Heritage Centre naturally tends to produce results which would be applicable to graph-based information retrieval techniques. Ideally, the heritage projects would have a high proportion of concepts which function as branches, and a single, completely-connected semantic network. If this is found to be the case, then tag co-occurrence based suggestion mechanisms will work effectively, providing support for using SAGE and SAGA in this application domain.

However, some heritage projects may be partially incomplete, with many objects having no annotations, or a relatively small number of annotations. These projects are more likely to be disconnected in several subgraphs, and would be a challenging dataset for tag co-occurrence based suggestion mechanisms. While this wouldn't rule out the use of SAGE and SAGA, it would present a case for needing a different approach to be designed to effectively support annotation in digital heritage projects.

6.6 Results

This section presents the results collected for the individual case study projects, analysing each in terms of completeness and connectedness. We begin with the Saumarez Combined collection produced as part of the previous evaluations in this thesis before investigating the projects conducted at the UNE Heritage Centre.

6.6.1 Saumarez Combined Results

The process used to aggregate the individual samples that make up this collection has resulted in the creation of a large, redundancy-laden set of annotations. Redundancies are introduced by semantic synonyms; concepts with the same meaning, but changed ordering of words, use of synonymous terms, or differences in pluralisations, tenses and spellings. Figure 6.9 (below) provides an example, where the challenging spelling of a historic building has resulted in four separate concepts being created.

Figure 6.9 - Example Synonyms

The primary goal of the Saumarez Combined collection is to determine whether suggestion algorithms are likely to be capable of bridging the divide between one individual's set of concepts and others without requiring a data cleaning, lemmatisation (Manning, Raghaven & Schütze, 2008) or suffix-stripping process (Porter, 1980). This resilience would allow data from multiple data sources to be mixed together, and allow annotations to be collected without any formal requirements for structure or syntax, decreasing the learning curve needed by volunteers who wish to contribute annotations.

To see whether this is the case, we present the summarised measurements calculated for the collection. Table 6.6 collects these statistics.

Table 6.6 Saumarez Combined Results

Project	Saumarez Combined
Object Count	1411
Annotation Count	8527
Annotated Object Ratio	0.99
Object Leaves	10
Object Branches	1385
Object Branch Ratio	0.99
Annotated Object Subgraphs	4

Both the object and annotation counts for this project are relatively large, stemming from the origin of this collection as a combination of sample projects. This demonstrates the effectiveness of the annotator role when used for crowdsourcing, as it allows a substantial amount of information to be collected in a relatively short period of time. Figure 6.10 shows the degree of completeness by which this collection was annotated.

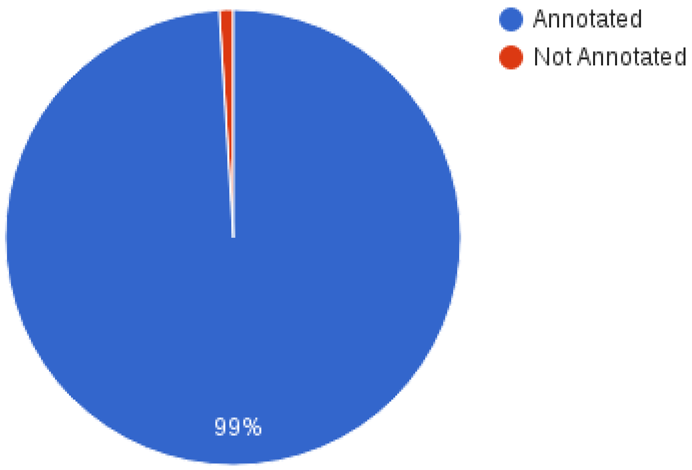


Figure 6.10 - Saumarez Combined Annotation Progress

The extremely high (but not perfectly complete) annotation level is due to a handful of objects in the samples being missed by the annotators. Nevertheless, this level of annotation has provided excellent coverage for this collection.

The large majority of objects in the combined samples have more than one concept, allowing them to act as branches which semantically links concepts together (see Table 6.6). In fact, over 99% of the concepts are branches, showing that even though the individual samples had a branch ratio averaging between 40%-41% when they were evaluated in the previous chapter, through concept collisions (where identical concepts were flattened together) this percentage has been significantly boosted. Figure 6.11 illustrates the distribution.

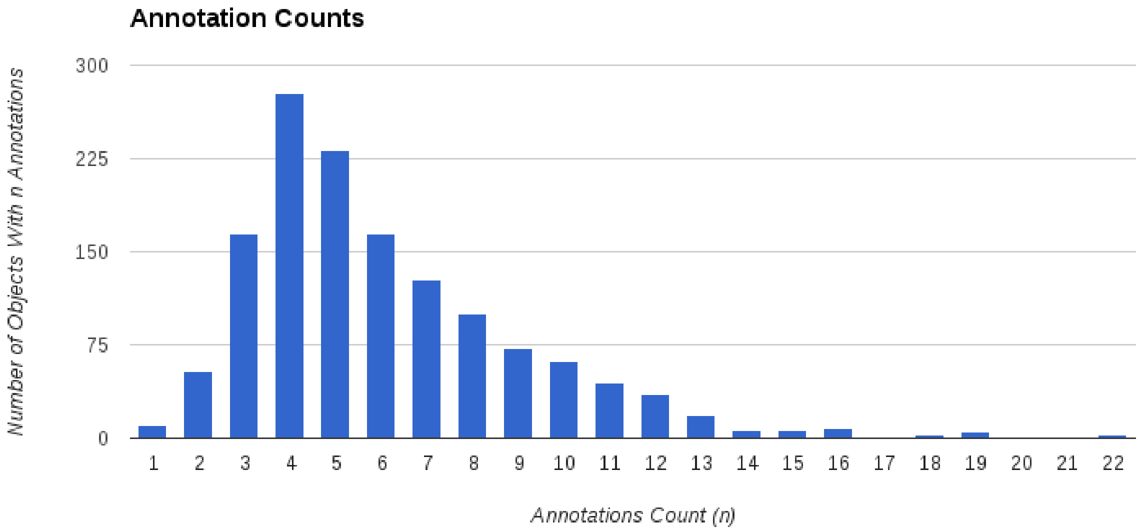


Figure 6.11 - Saumarez Combined Annotation Distribution

Concept distributions show that most objects had between 3 and 8 concepts, with some objects having considerably more (9-22) and a few having less (1-2). This shows that the majority of branches in the project were better connected than the minimum required (i.e. two annotations) to be defined as a branch. Again, this shows the high level of emergent complexity achieved through combining these samples without the need for any laborious integration process, such as stemming or lemmatisation to ensure that the concepts created by the different users could be merged in the combined collection. Complete object and concept annotation levels for all projects are available in Appendix E and Appendix F respectively.

Overall, the combined annotations form four distinct subgraphs (see Table 6.7). Subgraph counts show that almost every object falls into the main subgraph, with only three objects being unreachable through tag co-occurrence (refer to Appendix G).

Table 6.7 - Subgraphs Overview

Subgraphs Overview	
Subgraphs	4
Objects in Main Subgraph	> 99%

This shows that a tag co-occurrence similarity mechanism is able to reach almost every object in a project by following object-concept links even without any bridging method, such as text similarity or popularity. These results indicate that simply using tag collisions (where a concept is independently used in two or more samples produced by annotators, and flattens into the same concept when imported into a combined collection) is sufficient to link separate samples together into a cohesive whole.

6.6.2 Heritage Project Results

The results in the Heritage Project section represent snapshots of the ongoing work conducted by the volunteers at the UNE Heritage Centre. As such, we are able to see projects at various levels of completion, which allows us to compare how the desirable characteristics sought in these projects (the connectedness metrics) develop in projects that are actively being annotated (measured in completeness).

Armidale in Photos

Armidale in Photos was an individual's project, with one user. Roughly half the objects in this collection were annotated at time of writing. Table 6.8 provides the project statistics.

Table 6.8 - Armidale in Photos Results

Project	Armidale in Photos
Object Count	85
Annotation Count	101
Annotated Object Ratio	0.49
Object Leaves	9
Object Branches	33
Object Branch Ratio	0.79
Annotated Object Subgraphs	2

While the annotation count is not much higher than the object count, the annotated object ratio (illustrated in Figure 6.12) means that these annotations are concentrated in half of the objects.

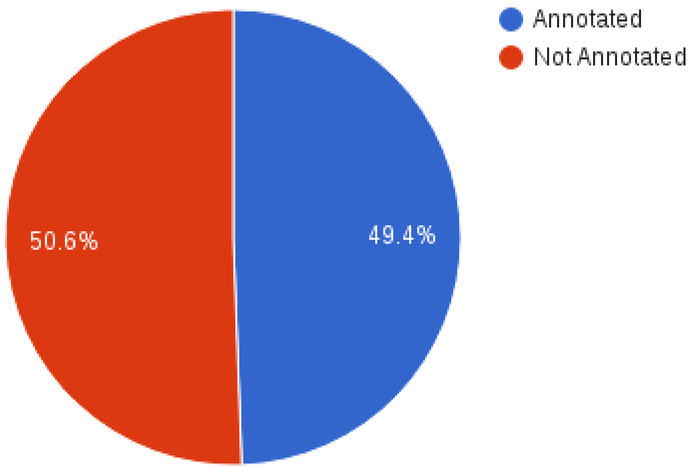


Figure 6.12 - Armidale in Photos Annotation Progress

With a branch ratio of 79%, the majority of objects are branches, but mostly only meet the minimum number of concepts to be considered as such (refer to Figure 6.13).

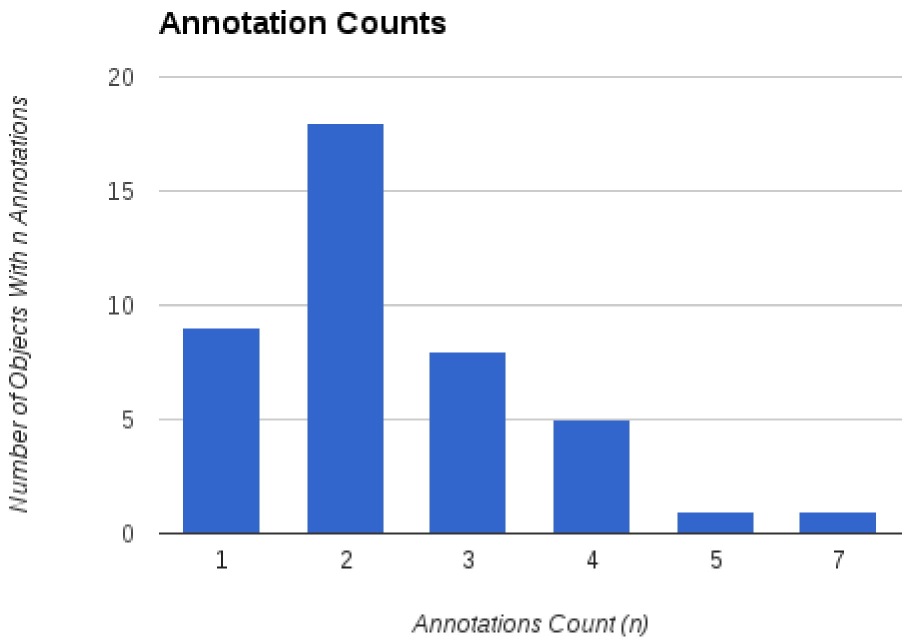


Figure 6.13 Armidale in Photos Annotation Distribution

Interestingly, the distribution of annotations is still sufficient to produce only two discrete subgraphs, with the smaller subgraph consisting of a single object (refer to Appendix G). This is a very promising result, with almost every object being discoverable using a tag co-occurrence strategy -- assuming one is not unlucky enough to begin with the single object in the separate subgraph, which would necessitate a different method for finding suggestions.

Nickson AIF Photos

The Nickson AIF Photos project was conducted as a crowdsourcing experiment within SAGE; the groupwork conducted as part of the SAGE training sessions for the acceptance testing participants used this project. In total, 18 users contributed to this project: the acceptance testing participants as well as one of the members of the UNE Heritage Centre (refer to Table 6.9 for statistics).

Table 6.9 - Nickson AIF Results

Project	Nickson AIF
Object Count	415
Annotation Count	876
Annotated Object Ratio	0.33
Object Leaves	12
Object Branches	126
Object Branch Ratio	0.91
Annotated Object Subgraphs	8

The moderately high number of annotations concentrated in 33% of the objects (illustrated in Figure 6.14) yields a relatively high number of annotations in these objects. Nevertheless, the completeness in this collection is the lowest of the heritage projects.

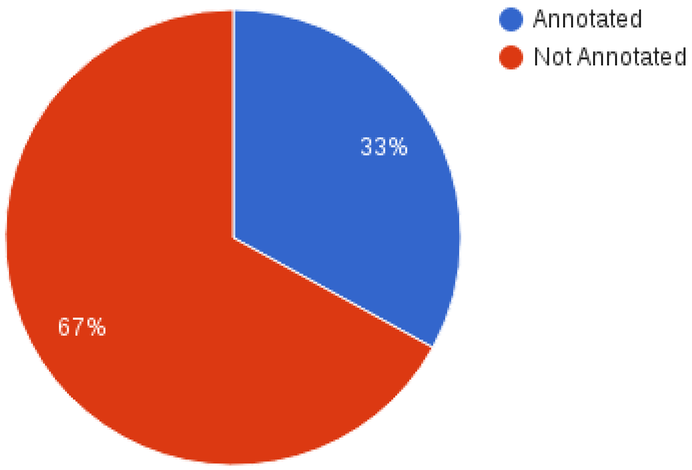


Figure 6.14 - Nickson AIF Photos Annotation Progress

Despite the low completeness, 91% of the objects in this project are branches. This presents an opportunity to see how a high branch ratio but low completeness project is expressed in annotation distributions and subgraphs.

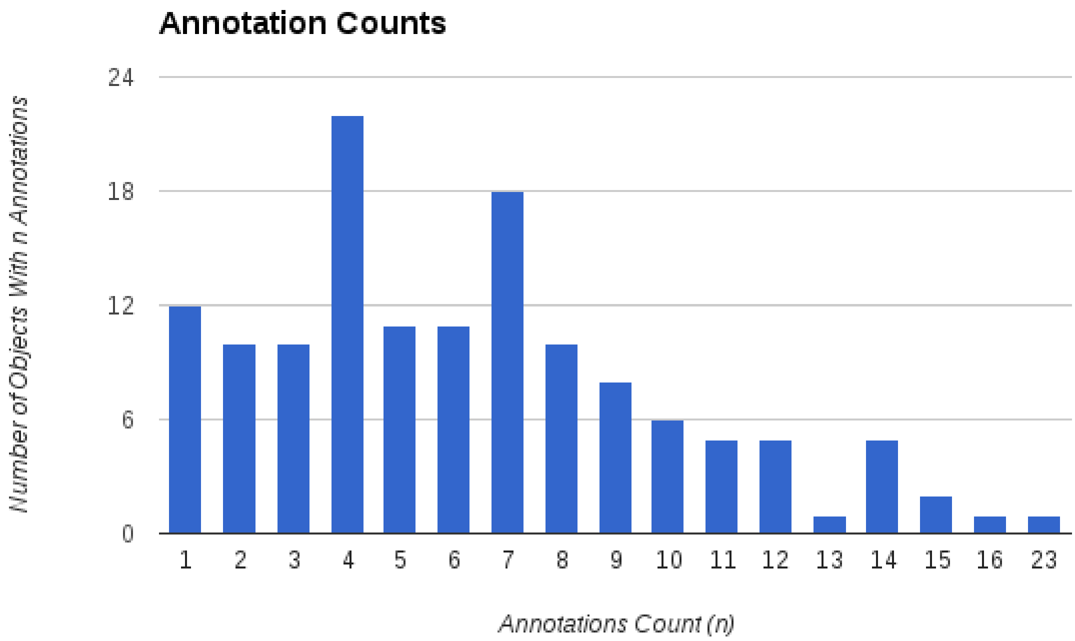


Figure 6.15 Nickson AIF Photos Annotation Distribution

The annotation distributions (refer to Figure 6.15) are unusual in that the objects with 4 and 7 concepts are much higher than the surrounding objects, and objects with 13 concepts are also lower than surrounding objects. The average number of concepts per object sits well above the minimum, showing that those objects that have been annotated are fairly well-annotated.

This project has the highest number of subgraphs in the heritage projects, with 8 distinct subgraphs. 91% of objects fall under the main subgraph (refer to Appendix G), but this still means that 12 objects aren't accessible from the main subgraph using a tag co-occurrence approach.

Nursing VAD Identification

The Nursing VAD Identification was conducted by a group in the UNE Heritage Centre. In total, four users contributed to this project (refer to Table 6.10 for statistics).

Table 6.10 - Nursing VAD Identification Results

Project	Nursing VAD Identification
Object Count	70
Annotation Count	122
Annotated Object Ratio	0.71
Object Leaves	23
Object Branches	27
Object Branch Ratio	0.54
Annotated Object Subgraphs	2

In total, 71% of the objects in this project were annotated (illustrated in Figure 6.16). Considering the relatively similar number of objects and annotations (70 and 122 respectively), this means that objects will have a relatively low average number of annotations.

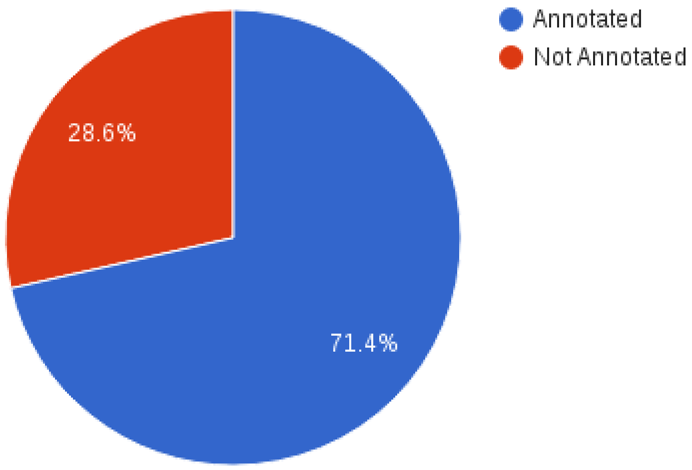


Figure 6.16 - Nursing VAD Identification Annotation Progress

The number of object leaves and object branches are almost the same, with 23 leaves to 27 branches and a branch ratio of 54%. This is reflected in the annotation distributions (shown in Figure 6.17) which shows a large number of objects with a single concept (the leaves) and the branches distributed mostly in the range of having 2-5 concepts. The object with 19 concepts is a highly unusual outlier, and is a group photo where many of the nurses have been identified thanks to their names being in the caption.

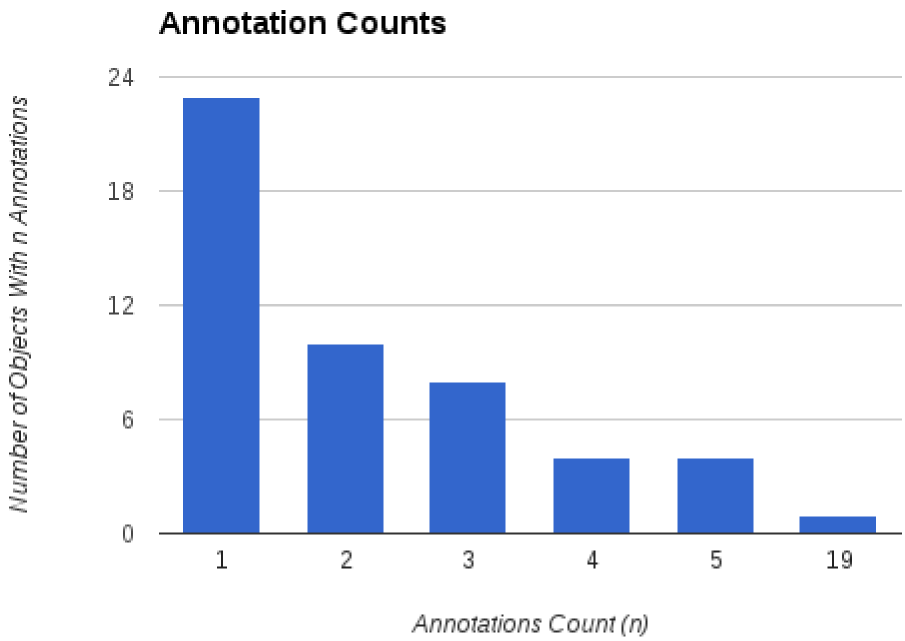


Figure 6.17 Nursing VAD Identification Annotation Distribution

Despite the low branch ratio, the Nursing VAD Identification has only two subgraphs, with the second subgraph containing a single object (refer to Appendix G). This means that almost all of the objects were accessible using a tag co-occurrence based suggestion mechanism.

Saumarez Gardens

Saumarez Gardens was an individual's project, with one user. Despite this, the project has a high number of objects and annotations. Refer to Table 6.11 for statistics.

Table 6.11 - Saumarez Gardens Results

Project	Saumarez Gardens
Object Count	521
Annotation Count	1362
Annotated Object Ratio	0.86
Object Leaves	91
Object Branches	356
Object Branch Ratio	0.80
Annotated Object Subgraphs	2

With an annotated object ratio of 86% (refer to Figure 6.18), this project has a relatively high level of completeness courtesy of the single volunteer working on its annotations, a total of 448 objects having been annotated.

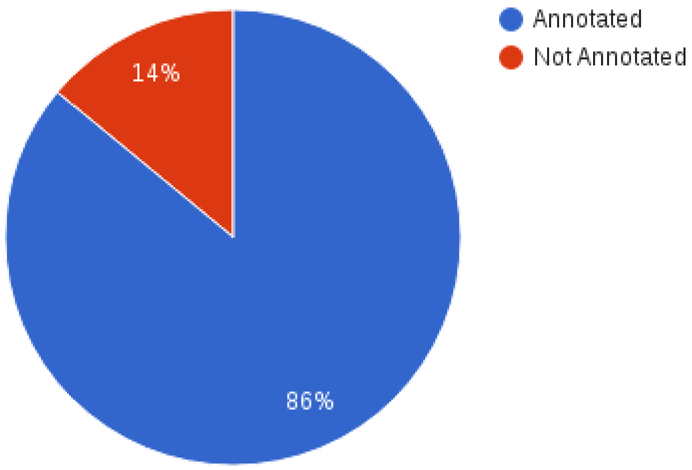


Figure 6.18 - Saumarez Gardens Annotation Progress

The object branch ratio is 80%, roughly average among the heritage projects. Considering the distributions shown in Figure 6.19, we can see that most objects have around 3-4 concepts, a modest average.

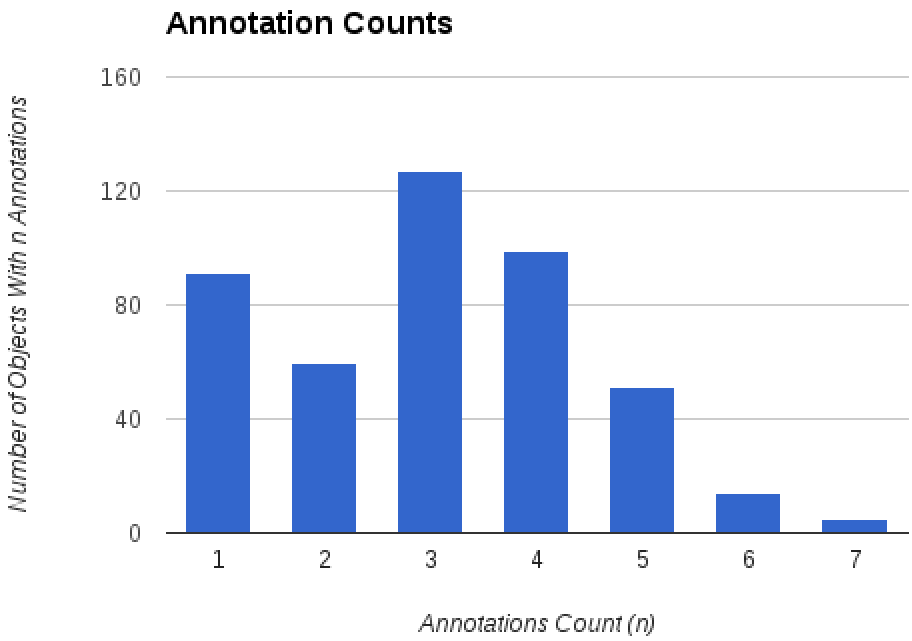


Figure 6.19 Saumarez Gardens Annotation Distribution

The project has two subgraphs (refer to Appendix G), with the second subgraph containing just a single object. This shows, as in all the other heritage projects, that virtually all objects in these SAGE digital heritage projects are accessible using a tag co-occurrence based algorithm such as SAGA or VotePlus.

6.7 Discussion

Both the annotation subgraph problem and the annotation complexity problems were shown to be only a minor issue in heritage projects conducted in SAGE, with the number of discrete subgraphs and the level of annotation connectedness typically falling in very acceptable ranges. The number of subgraphs and the number of objects outside the main subgraph was shown to be small, and all projects were shown to contain on average a high proportion of branch concepts, showing good interconnection of annotations. Both of these factors are very encouraging for the application of graph analysis techniques, such as those based on tag co-occurrence.

There was also a reasonable breakdown between popular concepts which helped to link together multiple images, and specific concepts that were used to provide a more specialised insight into certain images. No project had less than a 50% branch ratio, with the most frequent ratio being approximately 80%. The average number of concept annotations in the branches varied significantly by project.

All projects conducted in SAGE have a small number of subgraphs, with almost all objects falling under the primary subgraph. This observation even holds true for the Saumarez Combined project, where the chance of producing a number of distinct subgraphs was considerably higher than on average. This suggests that users will naturally identify and annotate concepts which are present in many images that allow distinct sets of work to be connected.

The high degree of concept reuse and the connectedness of annotations indicates that SAGE has been effective in encouraging users to identify and document the common elements in the heritage datasets. Interestingly, this practice may have resulted in a qualitative shift in the way that these datasets are documented. Firstly, this approach favours the style of multiple short descriptors which each address a specific idea presented in the material, contrasted with annotation using a single descriptive passage. Secondly, this approach promotes annotating every piece of material, rather than selectively annotating the most notable and interesting images. This behaviour is highly beneficial when using SAGE as an instrument to explore and retrieve heritage information, and shows that the users of SAGE are willing to adapt their workflow to accommodate the use of this technology.

Powered by SAGA, SAGE has demonstrated the viability of user participation to create a rich and informative network of annotations, and that while the effort of work scales proportionally to the number of objects being annotated, the number of concepts that need to be created does not. These findings support the use of SAGE as the source of training material for automated classifier or annotator systems, which are able to analyse vast numbers of objects with lesser semantic thoroughness compared to human annotators, but can allow this large-scale data to be made searchable and accessible to users due to its high scalability.

The relatively uncontrolled nature of a case study shows that these results are likely to be generalisable to a wide range of projects, though the specific practices used in different institutions (e.g. museums compared to archives, or government sector compared to private sector) may affect the way that SAGE is utilised. Another factor to consider is the other types of software employed by institutions, which may stipulate that certain types or formats of information should be prioritised for cross-software compatibility.

Chapter 7: Conclusion

7.1 Overview

In this chapter, we review the results obtained from the three major assessments conducted in this PhD project and identify how the design choices made in SAGA and SAGE have achieved a measurable impact in the key performance indicators in each assessment. This allows us to evaluate the extent to which these results affirm the project's research objectives, as well as how the results compare to the best in the field.

We then discuss the challenges that were identified in this project and how they were overcome through the software design of SAGA and SAGE. Reflecting on these issues provides justification for the approaches that were adopted in the algorithm and user environment, and presents a reference model which other information systems in digital heritage can refer to as a way of avoiding potential pitfalls.

Finally, we examine the potential of future work to extend the project's objectives, and the areas of research within which this project's findings could be useful. This lays out a prospective sequence of developments which proceeds from this work, which would advance the state of the art in digital heritage. We then present the closing remarks of this project.

7.2 Review of Results

Three major assessments were conducted in this research project: *performance evaluations*, *user acceptance evaluations* and *case study evaluations*. SAGA and SAGE have obtained promising results in all three areas, and demonstrates higher performance and user acceptance when compared to the results obtained by the popular and widely-referenced VotePlus algorithm.

7.2.1 Algorithmic Performance

When compared to the Vote, VotePlus, Sum and SumPlus algorithms on the MIRFlickr25K photo collection, SAGA achieved superior overall results in both precision and recall, which also led to a higher overall phi coefficient, the primary measure of algorithmic performance (refer to Figure 7.1). This demonstrates that SAGA is highly capable of finding relevant suggestions at all four levels of training data.

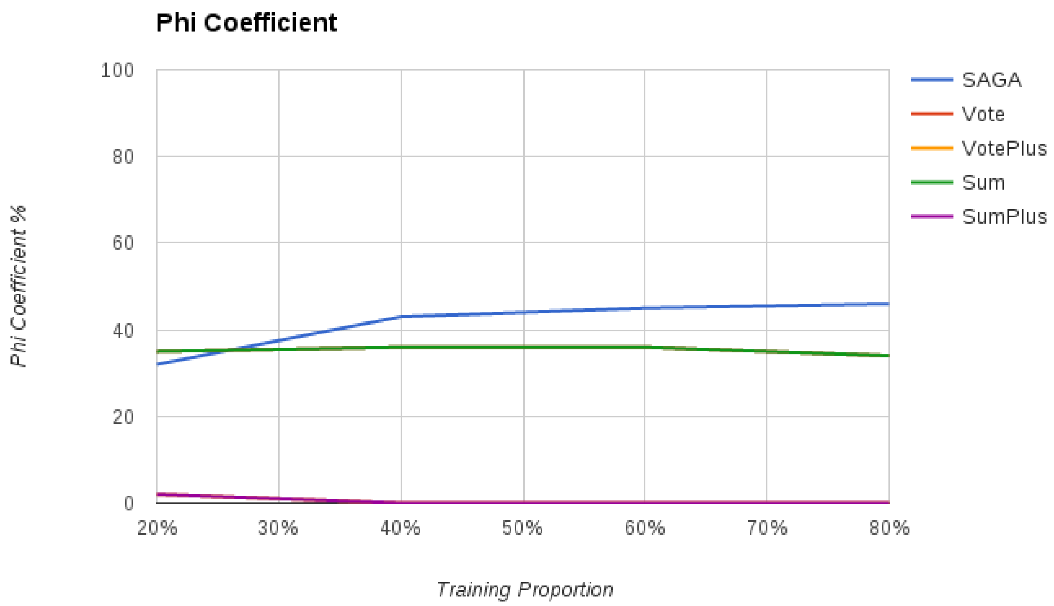


Figure 7.1 - Phi Coefficient vs Training Proportion

While the closest competitors, the Sum and Vote algorithms (occupying the same line in the graph), obtained slightly better precision at very low levels of data, their effectiveness dropped off as more training material was added, whereas SAGA's performance steadily improved. The SumPlus and VotePlus algorithms (occupying the same, lower line in the graph) obtained lower results at all levels of training material.

Even when artificially constraining results to the top suggestions in the precision@5 and MRR measurements, SAGA remained competitive, with both the Sum and SumPlus algorithms performing equally well to SAGA in MRR, and performing slightly better than SAGA in the precision@5 measurement (refer to Figure 7.2). This shows that while the overall performance of these competing algorithms was lower, they are still quite capable of finding relevant results among their top-scoring suggestions.

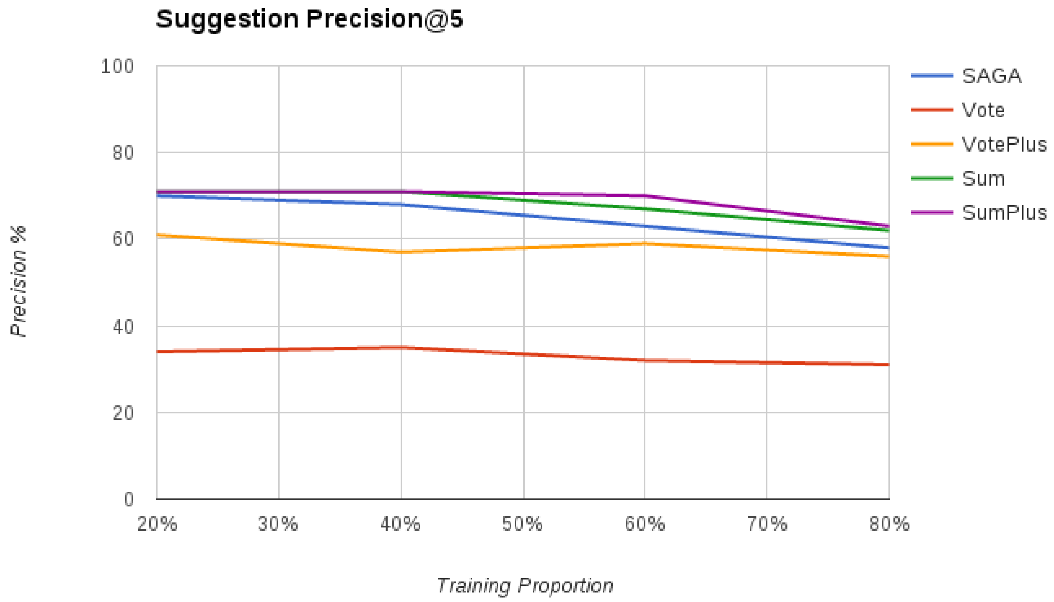


Figure 7.2 - Suggestion Precision @ 5 vs Training Proportion

The algorithm that the authors of the tag recommendation paper (Sigurbjörnsson & Van Zwol, 2008) have recommended is VotePlus, a modestly-performing algorithm in our MIRFlickr25K-based evaluations. SAGA outperforms the VotePlus algorithm in each of the tests conducted on this dataset, including those that use the metrics applied in the original paper. However, with the author's backing, strong results in their original assessments, and wide usage of this algorithm in other work, it was selected as the most appropriate algorithm for comparison in the user acceptance assessment.

7.2.2 User Acceptance

User acceptance testing was conducted using surveys, annotation rate measurements and suggestion acceptance measurements collected while the acceptance testing participants were working on sample projects created from the Saumarez Collection. Survey responses from responders among the heritage centre participants were also collected.

When users were surveyed using the standard questions from the widely-used TAM and SUS surveys, participants responded favourably to the usability and learnability aspects of SAGE, and indicated perceived potential for SAGE to be used within heritage research projects for annotation work. High usability ratings for SAGE supports and affirms the other results obtained in the user acceptance evaluations, as it ameliorates concerns regarding software usability as a factor that could affect these results.

Both SAGA and VotePlus obtained similar results when usage testing was conducted to determine if the algorithms led to different user productivity levels. Work was conducted slightly more quickly in SAGA, while slightly higher contributions of concepts and annotations was achieved in VotePlus. The difference between the two in terms of annotations per minute was relatively small, and both demonstrate a progressively increasing capability to support annotations as more information becomes available in a project (refer to Figure 7.3).

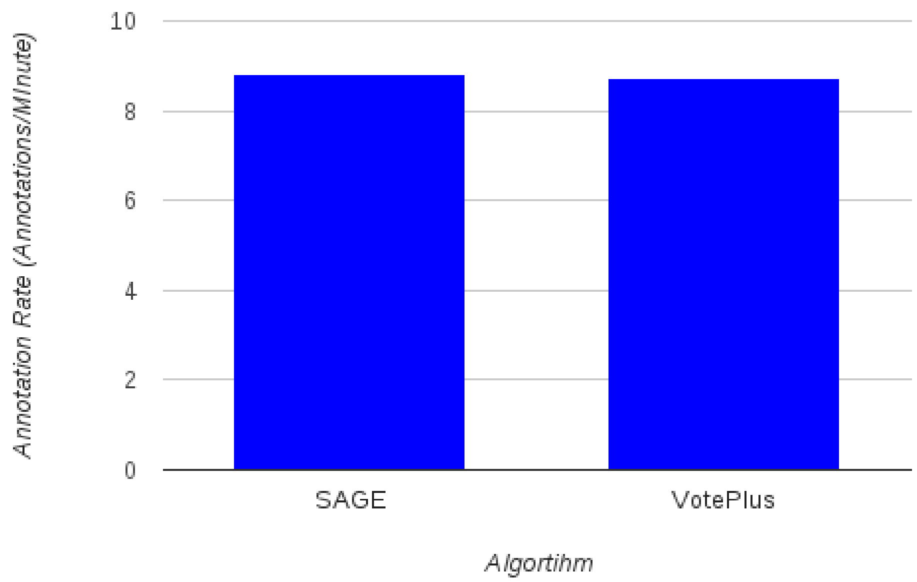


Figure 7.3 - Annotation Rate Comparison

In usage testing scenarios, SAGA was consistently accepted more frequently than the competing VotePlus algorithm, showing that the enhancements SAGA uses for finding relevant suggestions has had a positive effect on its agreeability with users. As more annotations were added, SAGA and VotePlus both grew in suggestion acceptance, with SAGA growing at a faster rate than VotePlus (refer to Figure 7.4).

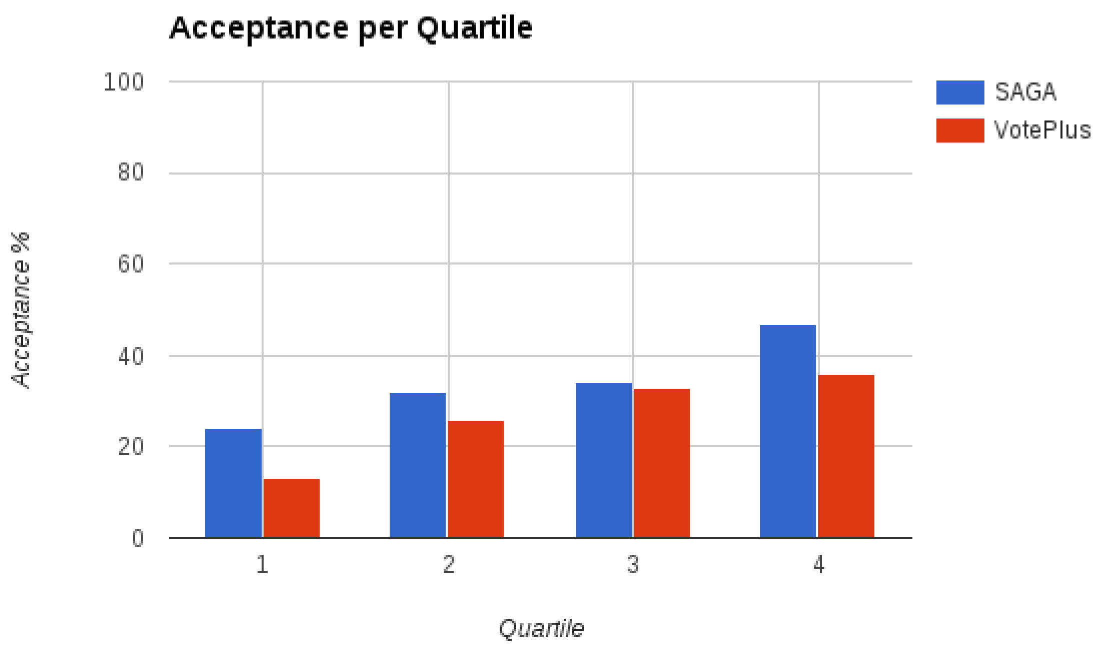


Figure 7.4 - Acceptance per Quartile Chart

7.2.3 Digital Heritage Case Study

A number of heritage annotation projects were conducted in SAGE over the course of a year. These projects were examined based on the quantity and quality of information captured within them, with metrics being calculated for the completeness and connectedness of the annotations.

It was discovered that despite the ongoing annotation efforts for the projects, with objects having yet to be considered for annotation with the limited resources available, the level of connectedness in the graph formed between objects and concepts was quite high, providing an information-rich resource.

In an experiment conducted using a crowdsourced digital heritage project designed to produce a high level of redundancy in annotations, the Saumarez Combined project, it was shown that the use of computer-supported work led to an almost completely connected graph with a very high level of branch rate. This largely mitigates the anticipated issue of having discrete subgraphs, which are problematic for purely tag co-occurrence based suggestion mechanisms (refer to Figure 7.5).

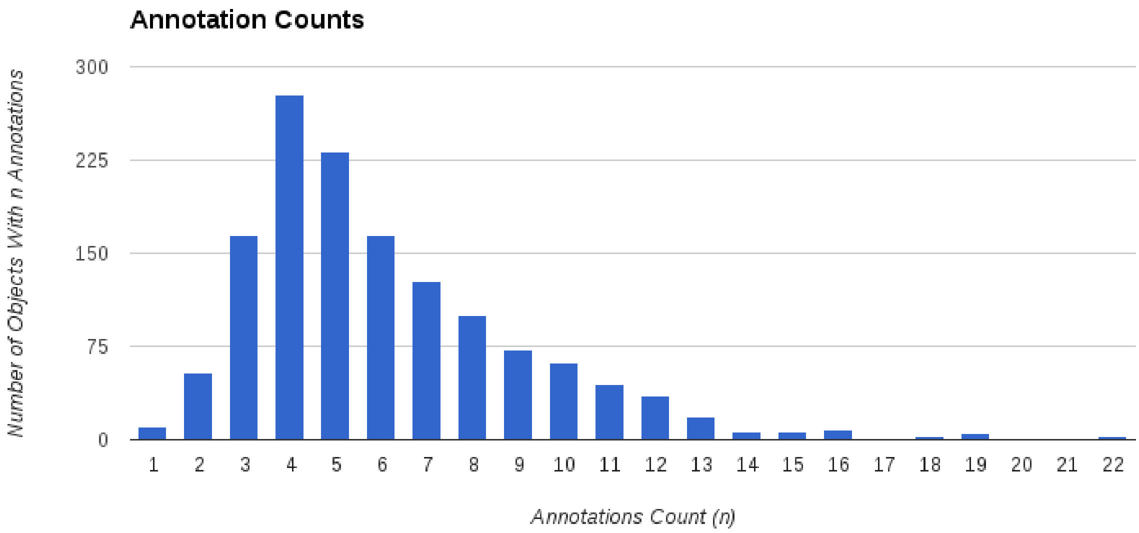


Figure 7.5 - Saumarez Combined Annotation Distribution

This implies that tag co-occurrence based suggestion mechanisms are likely to be effective when using the annotator role provided in SAGE to securely crowdsource annotations from a diverse range of participants, as no additional steps to artificially promote connectedness in the overall annotation graph are necessary to prepare the data before it is merged.

7.3 Contributions

This PhD research identifies, analyses and proposes solutions to a range of issues affecting the application of semantic search technologies to the area of digital cultural heritage. This process seeks to provide a foundation on which further research can be conducted, encouraging the adoption of these technologies to help augment the efforts of experts and volunteers in a diverse range of heritage projects, with particular regard to those that seek to utilise communities as a source of historical or cultural knowledge.

7.3.1 Research Aims

This PhD project has sought to support the early phases of the information lifecycle of heritage collections. With collections of heritage materials being steadily digitised by heritage organisations around the world, there is a growing opportunity for novel software to be developed which can support the annotation process for these materials.

By producing these annotations, we provide documentation for the digitised heritage materials which is informative to viewers, can prompt users with specialised knowledge to contribute what they know through community memory building, and can also help with user-supporting algorithms that use annotations to direct users to informative and interesting parts of the collection, such as through recommendations or allowing users to search for topics they are interested in.

Most importantly, comprehensive annotations for publicly-available digital heritage collections would provide the ideal information for developing and testing artificial intelligence algorithms which can examine the content of heritage materials and correlate them with the annotations that have been provided to be able to autonomously annotate heritage materials. This would primarily involve content-based information being annotated, rather than context-based information, which may still be easier for humans to contribute for the foreseeable future.

To accomplish this, this research project has sought to determine a method by which the diverse and semantically complex collections of heritage data can be explored and searched effectively, while encouraging users to contribute new information as participants in a community memory building process.

The project also seeks to identify and analyse the challenges encountered by multimedia technology when specifically applied to digital heritage, and provide strategies that account for these critical challenges. SAGE and the SAGA algorithm have contributed valuable new tools for exploring and contributing to digital cultural heritage; through performance testing, acceptance testing and a case study application, SAGA and SAGE have demonstrated their efficacy as a support system for community memory building.

Finally, SAGE seeks to provide a facility whereby annotations can be exported, providing the necessary priming data for other information retrieval technologies used in digital heritage research. SAGE can export a project's worth of annotations through the inbuilt export tool. These exported annotations are made available in the widely-used CSV data format for cross-program compatibility. SAGE provides extensible algorithm and evaluation code, allowing algorithms to be compared in the same environment using the same evaluations.

7.3.2 Challenges

Eleven issues faced by information systems specialising in digital heritage were identified during the course of this PhD project. These challenges led to or necessitated certain decisions being made regarding SAGA's algorithm and SAGE's architecture, and are relevant to other projects being conducted in this area. We review these challenges here, with the challenges being:

1. Sparse Existing Resources
2. Digital Heritage Images
3. Sparsity of Experts
4. Cold Start
5. Long Tail
6. Expertise
7. Certainty
8. Trust
9. User Engagement
10. Annotation Subgraphs
11. Superficial Annotations

One of the key challenges that motivated the development of the SAGE environment was the sparse existing resources in terms of digital heritage annotations. Unlike image collections (such as MIRFlickr25K) which are openly available with relevant annotations, digital heritage image collections often lack a comprehensive mapping of images to relevant annotations which would allow them to be analysed by computational methods utilising techniques such as content-based image retrieval.

Computational methods are also made challenging by the nature of the heritage materials they analyse. Digital heritage images are missing much of the metadata which would be available in normal digital photography, such as GPS location and time of capture. Digital heritage images can also suffer from degradation and visual artefacts due to their age. Much of the work being conducted on these images is also semantically complex in nature, such as identifying events, inferring dates and recognising locations. This strongly encourages a human computation-based approach to handle this added semantic complexity.

When working with a specialised heritage collection, experts must be found who can identify the locations, people and events present in those materials. These experts may be few in number, separated by large distances, and have varied levels of technological acceptance, so relying on software that these experts are expected to install, configure, network and maintain raises a hurdle that must be overcome in order to begin collaborative work. Accessing a centralised server requiring minimal training and installation on the user's part, such as a web application accessed by web browsers, is an elegant solution to this issue.

Without a comprehensive set of annotations as training material, automated methods can be difficult to apply effectively, and without the use of simple, networked software to work with, annotations are difficult to collect. This necessitates a human-centric web application for collecting annotations (such as SAGE) that provides a shared environment for heritage experts around the country to collaborate on producing an initial set of annotations for high-value heritage materials.

A purely manual application for collecting annotations means that the amount of effort required to comprehensively annotate a collection of heritage materials will always be proportional to its size. Due to the value of the limited amount of time experts and volunteers have to work on the annotation process, a software-supported method for collecting annotations is preferable.

When these support algorithms are faced with limited existing annotations to analyse and to base suggestions upon, these algorithms will encounter difficulty in providing relevant suggestions. This is the cold start issue. Conversely, when many annotations are present, a great many suggestions may be detected as relevant to different degrees, and a means of establishing the best results to give to users is needed. This is a long tail in results.

SAGA provides suggestions for annotations as a means of reducing the effort required to annotate a collection. To address the cold start problem, SAGA places high emphasis on project-wide means of making suggestions while annotations are scarce (popularity), then shifts emphasis to local means of suggestions when annotations become plentiful (tag co-occurrence). While this still means that the cold start issue is encountered at the very beginning of each project, when no annotations or concepts exist, it allows suggestions to be made immediately once the first concept has been added.

To address the long tail in results, SAGA uses a neighbour voting strategy that allocates a limited quantity of influence to suggestions, which propagates through tag co-occurrence and popularity. The amount of influence available is limited by the maximum number of suggestions that could theoretically be made (i.e. for an object, the amount of influence will equal the number of concepts in the project) and suggestions compete to acquire influence in proportion to their relevance (i.e. more relevant suggestions get a greater share of the influence). SAGA filters weaker, low-influence suggestions before returning the remaining suggestions as results to the user.

Addressing challenges relating to production of suggestions at different levels of information has influenced the design of SAGA significantly. Similarly, addressing the issues of user expertise, certainty, trust and user engagement has a significant impact on the design of SAGE.

Some users of SAGE will have higher levels of expertise regarding a collection, some concept annotations will be more certain than others, and known volunteers will be more trustable than strangers. Considering these individual and social factors, SAGE has been designed so that small groups of collaborators can work in private projects, with the option of allowing larger groups of viewers or annotators to participate through a secure, key-based user role system.

Contributors can provide semantic concepts at different levels of complexity depending on their knowledge, from simple perceptive observations such as listing visible store or street signs through to utilising expertise to identify significant events and contextual information about images. These semantic concepts can intrinsically represent a degree of uncertainty through the concept description. Concepts that start with "May be an image of…" rather than "An image of..." are a perfect example of this.

User engagement is a very difficult challenge to address as it relies on factors regarding an individual's own motivations and experience, and as such it is the only challenge that was only modestly addressed in this research. SAGE seeks to minimise the barriers to these motivations, rather than attempting to engineer the motivation in the first place. SAGE has been verified by user surveys to have a high degree of ease of use and learnability, which are usually two major impediments to accomplishing a task. One of the ways that this is accomplished is through deploying SAGE as a web application, which allows any user with a web browser to quickly register for an account and begin collaborating or creating projects without the need for installation, updating or configuration.

The final two challenges relate to how SAGA is used within SAGE to produce annotations, which relate to one another through tag co-occurrence. Subgraphs can form when one set of annotations are created separately from the main set of annotations and pose a substantial issue for suggestion mechanisms that rely solely on tag co-occurrence, as suggestions from the secondary graph can't be accessed and therefore made from the main graph. On the other hand, if annotations are applied too infrequently or too generally, then there can be little to distinguish one object from another.

SAGE addresses the annotation subgraph problem by providing global-scale suggestions such as through popularity, and also through text similarity. These techniques create longitudinal connections that can reach across a dataset to provide annotation suggestions that can connect separate subgraphs. SAGE has addressed issues of superficial annotations through having users act as a key part of the annotation feedback cycle, a human-in-the-loop design that uses human relevance judgements to apply annotation suggestions only to those objects to which they are relevant (avoiding concepts being uniformly distributed) and the suggestion annotations which are made to users aim to continually improve the level of annotation in a project (addressing objects with few or no annotations).

7.3.3 Applications

It is intended that the SAGA algorithm, along with the contribution of human insights from the users of SAGE, will provide an effective tool for exploring and navigating semantically-related information in digital heritage collections. In particular, SAGE could be useful within several collection management practices employed by heritage institutions, including documentation, significance assessment and investigative processes.

The annotations produced in SAGE represent unstructured but rich information on both the content and context of digitised heritage artefacts. These annotations can be collected in conjunction with the structured metadata already documented by institutions as a means of improving the ability for information to be selectively retrieved from large collections, which is a significant challenge when faced with a large quantity of unvaryingly- or sparsely-annotated materials.

SAGE provides a facility for crowdsourcing annotations from volunteers, which frees experts from performing this work themselves, if they choose. This is particularly desirable as volunteers are increasingly “digital natives”, who were brought up with computers and the Internet, and find web-based applications like SAGE natural and easy to use. These volunteers are able to identify and record observations or common themes in collections that may be of interest to historians when determining the significance of the collection. The unstructured nature of annotations and the usability of the SAGE interface presents a distinct advantage compared to using formal ontologies or documentation practices, as a lengthy volunteer training process can be avoided.

Finally, SAGE assists with investigative processes by improving information retrieval capabilities within a collection, by allowing common themes (recorded as concepts) to be used as the basis for navigation, and most critically, allowing new observations about materials examined during an investigation to be contributed to the information stored in SAGE, making it easier to conduct a similar or related investigation in future.

SAGE can be applied to a wide range of collections, assuming that several key requirements are met. Firstly, SAGE requires materials to be represented in a simple digital format, such as a JPEG or PNG image which can be viewed in the interface. Secondly, SAGE requires that suitable expertise is available from experts and volunteers to annotate the collection, with larger collections requiring a larger number of people (or a longer period of time) to complete. Finally, SAGE does not (yet) support extremely large-scale collections numbering in the tens to hundreds of thousands of items, though having this capacity is a distinct possibility in the future.

7.4 Future Work

While the results obtained by SAGE and SAGA are very encouraging, it is important to note that the experiments designed to measure their effectiveness were conducted in controlled conditions with a limited group of users. It is likely that there are more challenges to be discovered when applying SAGE to large, real world heritage projects, particularly those that involve the general public, high quantities of concurrent web traffic, or primarily utilise representations of heritage materials other than digital imagery.

Beyond these implementational challenges, there are two main ways in which future work could further the outcomes of this research project. Enhancement of the SAGA algorithm could provide a much higher degree of scalability, allowing for larger projects with more contributors. Application to larger projects could then produce the comprehensive set of annotations and openly-available digital heritage images that would enable broad scale, autonomous techniques to be designed and tested in digital heritage research.

7.4.1 Scalability

Throughout the course of the evaluations performed on SAGA and SAGE, SAGE has been shown to be able to collect hundreds to thousands of annotations for small to medium-sized heritage collections. The number of concepts did not grow proportionally to the number of objects in any of the heritage collections, and in fact stabilised at a much smaller percentage, showing that collections of similar material have a limited number of people, events, themes or locations, which can be articulated adequately by a small number of concepts.

As the SAGA algorithm becomes more costly in projects with many annotations, SAGE experienced a decrease in performance on larger collections with many objects, concepts and annotations. This limits the practical application of SAGE on very large collections, numbering in the tens to hundreds of thousands of objects (10,000-100,000+ objects). For extremely large data collections, the time taken to produce a set of suggestions becomes particularly noticeable by users. This can make it laborious to work with larger data collections, but is an issue that could be addressed by techniques to improve the scalability of SAGE, such as multi-core processing or algorithmic optimisations.

It isn’t always possible to readily access motivated volunteers for annotating certain photo collections in SAGE. While every effort has been made to allow small groups and individuals to work effectively within this environment, large quantities of data in a collection can still pose a challenge without a commensurately motivated group of participants. Integration with a commercial human computation service such as

Amazon's Mechanical Turk (Amazon, n.d.a), where human workers are paid a small amount of money for each predefined, simple task they complete, could provide human processing power on demand in these instances (Amazon, n.d.a).

7.4.2 Heritage Projects

The best way to advance the outcomes laid out in this research project is to actively apply SAGE and SAGA to new heritage projects, particularly newly digitised heritage collections where annotations are not yet present, and where the addition of semantically-rich documentation created using SAGE would be likely to have a long-term positive benefit due to expected demand for investigative or exploratory collection management processes.

Ideally, this would be accomplished with networking between heritage experts who are interested in digital technologies and share similar historic interests. Establishing a small group of motivated researchers as contributors to a project in SAGE would allow larger collections containing many thousands of images to be tackled, and the group would gain the benefits of diverse experience brought by having multiple experts and institutions actively collaborating on the collection.

A project such as this presents an opportunity for SAGE and SAGA to be exposed to sustained real-world conditions, where the software and algorithm would need to perform effectively while conforming to institutional needs and requirements. This would help to uncover and explore the limitations that the design, architecture and multiagent approach will encounter, and serve as an informative account that successive digital heritage software projects can refer to.

7.4.3 Open Annotations

After large heritage collections are comprehensively annotated using SAGE, the collections can be made available to others along with their set of exported annotations. By sharing the annotations produced in SAGE, SAGE acts as part of a toolchain that incorporates other specialised software, or as part of a process that seeks to bring new resources to the public. This casts SAGE not as an isolated piece of software, where annotations are retained for select group indefinitely, but as part of a wider context of investigation and preservation.

The semantic richness and accessibility of open annotation sets created in SAGE makes them desirable as training and verification data for large-scale computer science data analysis techniques, including as content-based image retrieval and machine learning, which could then be applied to digital heritage as a means of developing new techniques and capabilities in this domain.

Should multimedia information retrieval algorithms be fully automated without requiring human intervention, it would allow vast heritage collections to be made searchable and accessible by users. Ultimately, this would benefit both disciplines as it would allow vast quantities of heritage materials to be analysed computationally, which makes searching them much more effective, and it also provides a challenging application area for image information retrieval techniques which can be refined to work on heritage materials.

Giving a specific example of how this might be used, SAGE could be complemented with an image analysis algorithm, such as object and facial recognition. This would allow unannotated objects to be found in collections by leveraging the annotations from objects in SAGE as training material. Users could then perform a search for that object across both the annotated and unannotated portions of a large collection, potentially then annotating the objects where the object was correctly identified as a means of refining the object recognition algorithm's training.

7.5 Conclusion

SAGE and the SAGA algorithm are an advancement in the current state of the field of multimedia information retrieval within the domain of digital heritage, and form a platform of research and services upon which other techniques can build. SAGE addresses a shortfall in existing heritage systems, whereby an initial quantity of semantic information needs to be captured in order to obtain more. In this way, SAGE provides an essential intermediate step in research progress in this area, which allows digital heritage materials to be annotated, and for these annotations to then be analysed and produced on a massive scale using other algorithms.

References

Agirre, E., Aletras, N., Clough, P. D., Fernando, S., Goodale, P., Hall, M. M., ... & Stevenson, M. (2013, August). PATHS: A System for Accessing Cultural Heritage Collections. In *ACL (Conference System Demonstrations)*(pp. 151-156).

Agosti, M., & Ferro, N. (2007). A formal model of annotations of digital content. *ACM Transactions on Information Systems (TOIS)*, 26(1), 3.

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 107-136.

Aletras, N., Stevenson, M., & Clough, P. (2012). Computing similarity between items in a digital library of cultural heritage. *Journal on Computing and Cultural Heritage (JOCCH)*, 5(4), 16.

Amazon. (n.d.a). *Amazon Mechanical Turk* - *Welcome*. Retrieved 18 February 2016 from https://www.mturk.com/mturk/welcome

Amazon. (n.d.b). *Amazon Simple Storage Service (S3)*. Retrieved on 18 February 2016 from https://aws.amazon.com/s3/

Ara Irititja Project. (2011). *The Ara Irititja Project* - *About Ara Irititja* - *Collection Management Software*. Retrieved from http://www.irititja.com/about_ara_irititja/index.html

Ardissono, L., Gena, C., Aroyo, L., Kuflik, T., Wecker, A. J., Oomen, J., & Stock, O. (2015, March). PATCH 2015: Personalized Access to Cultural Heritage. In

Proceedings of the 20th International Conference on Intelligent User Interfaces (pp. 447-449). ACM.

Ardissono, L., Kuflik, T., & Petrelli, D. (2012). Personalization in cultural heritage: the road travelled and the one ahead. *User Modeling and User-Adapted Interaction,* 22(1-2), 73-99.

Australian National University. (2013). *Deepening histories of Place*. Retrieved from http://www.deepeninghistories.anu.edu.au/

Babaoglu, O., Meling, H., & Montresor, A. (2002). Anthill: A framework for the development of agent-based peer-to-peer systems. In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on* (pp. 15-22). IEEE.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA).*

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies,* 4(3), 114-123.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007, January). Open Information Extraction from the Web. In *IJCAI* (Vol. 7, pp. 2670-2676).

Bartolini, I., Moscato, V., Pensa, R. G., Penta, A., Picariello, A., Sansone, C., & Sapino, M. L. (2013). Recommending multimedia objects in cultural heritage applications.

In *New Trends in Image Analysis and Processing–ICIAP 2013* (pp. 257-267). Springer Berlin Heidelberg.

Bartolini, I., Moscato, V., Pensa, R. G., Penta, A., Picariello, A., Sansone, C., & Sapino, M. L. (2014). Recommending multimedia visiting paths in cultural heritage applications. *Multimedia Tools and Applications*, 1-30.

Beale, R. (2007). Supporting serendipity: Using ambient intelligence to augment user exploration for data mining and web browsing. *International Journal of Human-Computer Studies*, 65(5), 421-433.

Bennett, J., & Lanning, S. (2007, August). The netflix prize. In *Proceedings of KDD Cup and Workshop* (Vol. 2007, p. 35).

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 28-37.

Bloehdorn, S., Petridis, K., Saathoff, C., Simou, N., Tzouvaras, V., Avrithis, Y., ... & Strintzis, M. G. (2005). Semantic annotation of images and videos for multimedia analysis. In *The Semantic Web: Research and Applications* (pp. 592-607). Springer Berlin Heidelberg.

Bonchev, D., & Buck, G. A. (2005). Quantitative measures of network complexity. In *Complexity in Chemistry, Biology, and Ecology* (pp. 191-235). Springer US.

Borges, J., & Levene, M. (2000). Data mining of user navigation patterns. In *Web Usage Analysis and User Profiling* (pp. 92-112). Springer Berlin Heidelberg.

Borgman, C. L. (1999). What are digital libraries? Competing visions. *Information Processing and Management*, 35(3), 227-243.

Box. (n.d.). *Box | Secure Content & Online File Sharing for Business*. Retrieved on 18 February 2016 from https://www.box.com/

Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 43-52). Morgan Kaufmann Publishers Inc.

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18), 3825-3833.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194), 4-7.

Brooke, J. (2013). SUS: a retrospective. *Journal of Usability Studies*, *8*(2), 29-40.

Budescu, D. V., & Chen, E. (2014). Identifying expertise to extract the wisdom of crowds. *Management Science*, 61(2), 267-280.

Cambria, E., Havasi, C., & Hussain, A. (2012, May). SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis. In *FLAIRS Conference* (pp. 202-207).

Cameron, F., & Kenderdine, S. (2007). *Theorizing digital cultural heritage: A critical discourse*. Mit Press.

Carmel, D., Zwerdling, N., & Yogev, S. (2012). Entity oriented search and exploration for cultural heritage collections. In *Proceedings of WWW*.

Carroll, J. M. (2010). Beyond Being Social: Prospects for Transformative Social Computing. *Communications of the Association for Information Systems*, 27(34), 641-650.

Carrozzino, M., & Bergamasco, M. (2010). Beyond virtual museums: Experiencing immersive virtual reality in real museums. *Journal of Cultural Heritage*, 11(4), 452-458.

Cedilnik, A., & Rheingans, P. (2000, October). Procedural annotation of uncertain information. In *Visualization 2000. Proceedings* (pp. 77-84). IEEE.

Celma, Ò., & Serra, X. (2008). FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4), 250-256.

Čermáková, M. (2012, August 15). “Lens into the past”. *The Prague Post*. Retrieved on February 26 2016 from http://www.praguepost.cz/tempo/14029-lens-into-the-past.html

Chang, W. Y., Abu-Amara, H., & Sanford, J. F. (2010). *Transforming Enterprise Cloud Services*. Springer Science & Business Media.

Chedzoy, O. B. (2006). Phi-coefficient. In *Encyclopedia of Statistical Sciences*.

Chen, H., Wu, Z., & Cudré-Mauroux, P. (2012). Semantic Web Meets Computational Intelligence: State of the Art and Perspectives. *Computational Intelligence Magazine, IEEE*, 7(2), 67-74.

Chianese, A., Marulli, F., Piccialli, F., & Valente, I. (2013, December). A novel challenge into multimedia cultural heritage: an integrated approach to support cultural information enrichment. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2013 International Conference on* (pp. 217-224). IEEE.

Chirita, P. A., Firan, C. S., & Nejdl, W. (2007, July). Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 7-14). ACM.

Hale, C. (2009, August 12). *codahale/bcrypt-ruby*. Retrieved from https://github.com/codahale/bcrypt-ruby/

Collins, M. J. (1996, June). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* (pp. 184-191). Association for Computational Linguistics.

CULTURA. (n.d.). *CULTURA: Cultivating Understanding and Research through Adaptivity*. Retrieved from http://www.cultura-strep.eu/

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2), 5.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 319-340.

Davison, R. M., Ou, C. X., & Martinsons, M. G. (2013). Information technology to support informal knowledge sharing. *Information Systems Journal*, 23(1), 89-109.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, 41(6), 391-407.

Deng, J., Russakovsky, O., Krause, J., Bernstein, M. S., Berg, A., & Fei-Fei, L. (2014, April). Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 3099-3102). ACM.

Doerr, M., & Iorizzo, D. (2008). The dream of a global knowledge network—A new approach. *Journal on Computing and Cultural Heritage (JOCCH)*, 1(1), 5.

Dropbox. (n.d.). *Dropbox*.Retrieved on 18 February 2016 from https://www.dropbox.com/

Egozi, O., Markovitch, S., & Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2), 8.

Europeana. (n.d.). *PATHS - Europeana Professional*. Retrieved from http://pro.europeana.eu/project/paths

Fernando, S., & Stevenson, M. (2012, April). Adapting wikification to cultural heritage. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities* (pp. 101-106). Association for Computational Linguistics.

Ferragina, P., González, R., Navarro, G., & Venturini, R. (2009). Compressed text indexes: From theory to practice. *Journal of Experimental Algorithmics (JEA)*, 13, 12.

Fielding, R. T., & Taylor, R. N. (2002). Principled design of the modern Web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2), 115-150.

Firan, C. S., Georgescu, M., Nejdl, W., & Paiu, R. (2010, October). Bringing order to your photos: event-driven classification of flickr images based on social knowledge. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (pp. 189-198). ACM.

Fischer, G. (2009, July). Cultures of participation and social computing: Rethinking and reinventing learning and education. In *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on* (pp. 1-5). IEEE.

Forman, G., & Kirshenbaum, E. (2008, October). Extremely fast text feature extraction for classification and indexing. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (pp. 1221-1230). ACM.

Fu, W. T., Kannampallil, T., Kang, R., & He, J. (2010). Semantic imitation in social tagging. *ACM Transactions on Computer-Human Interaction (TOCHI)*,17(3), 12.

Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., & Lochbaum, K. E. (1988, May). Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 465-480). ACM.

Garg, N., & Weber, I. (2008, October). Personalized, interactive tag recommendation for flickr. In *Proceedings of the 2008 ACM Conference on Recommender Systems* (pp. 67-74). ACM.

Goodall, S., Lewis, P. H., Martinez, K., Sinclair, P. A., Giorgini, F., Addis, M. J., ... & Stevenson, J. (2004). SCULPTEUR: multimedia retrieval for museums. In *Image and Video Retrieval* (pp. 638-646). Springer Berlin Heidelberg.

Google. (n.d.a). *Google Drive - Cloud Storage & File Backup for Photos, Docs & More.* Retrieved on 18 February 2016 from https://www.google.com.au/drive/

Google. (n.d.b). *Google Forms - create and analyse surveys, for free*. Retrieved on 18 February 2016 from https://www.google.com.au/forms/about/

Guarnieri, A., Pirotti, F., & Vettore, A. (2010). Cultural heritage interactive 3D models on the web: An approach using open source and free software. *Journal of Cultural Heritage*, 11(3), 350-353.

Gupta, M., Li, R., Yin, Z., & Han, J. (2010). Survey on social tagging techniques. A*CM SIGKDD Explorations Newsletter*, 12(1), 58-72.

Guy, M., & Tonkin, E. (2006). Tidying up tags. *D-lib Magazine*, 12(1), 1082-9873.

Hampson, C., Agosti, M., Orio, N., Bailey, E., Lawless, S., Conlan, O., & Wade, V. (2012a). The CULTURA project: supporting next generation interaction with digital cultural heritage collections. In *Progress in Cultural Heritage Preservation* (pp. 668-675). Springer Berlin Heidelberg.

Hampson, C., Lawless, S., Bailey, E., Yogev, S., Zwerdling, N., Carmel, D., ... & Wade, V. (2012b). CULTURA: A metadata-rich environment to support the enhanced interrogation of cultural collections. In *Metadata and Semantics Research* (pp. 227-238). Springer Berlin Heidelberg.

Hare, J. S., & Lewis, P. H. (2010, March). Automatically annotating the mir flickr dataset: Experimental protocols, openly available data and semantic spaces. In *Proceedings of the International Conference on Multimedia Information Retrieval* (pp. 547-556). ACM.

Hare, J. S., Lewis, P. H., Enser, P. G., & Sandom, C. J. (2006, January). Mind the gap: another look at the problem of the semantic gap in image retrieval. In *Electronic Imaging 2006* (pp. 607309-607309). International Society for Optics and Photonics.

Hare, J. S., Sinclair, P. A., Lewis, P. H., Martinez, K., Enser, P. G., & Sandom, C. J. (2006). Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In *Mastering the Gap: From Information Extraction to Semantic Representation / 3rd European Semantic Web Conference*.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3), 146-162.

Haslhofer, B., Momeni, E., Gay, M., & Simon, R. (2010, September). Augmenting Europeana content with linked data resources. In *Proceedings of the 6th International Conference on Semantic Systems* (p. 40). ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53.

Heroku. (n.d.). *Heroku | Deploy and run apps on today's most innovative Platform as a Service*. Retrieved on 18 February 2016 from https://www.heroku.com/platform

Hiltz, S. R., & Turoff, M. (1985). Structuring computer-mediated communication systems to avoid information overload. *Communications of the ACM*, 28(7), 680-689.

Historic Environment Scotland. (n.d.). *About the Project | ScotlandsPlaces*. Retrieved on 25 February 2016 from http://www.scotlandsplaces.gov.uk/about

Huiskes, M. J. & Lew, M. S. (2008). The MIR Flickr Retrieval Evaluation. *ACM International Conference on Multimedia Information Retrieval (MIR'08)*, Vancouver, Canada

IKRMNA Project. (2006). *IKRMNA - making collective memory with computer*s. Retrieved from http://www.cdu.edu.au/centres/ik/ikhome.html

ImageCLEF. (n.d.). *ImageCLEF - The CLEF Cross Language Image Retrieval Track*. Retrieved on 19 February 2016 from http://www.imageclef.org/

Isaac, A., & Haslhofer, B. (2013). Europeana linked open data–data. europeana. eu. *Semantic Web*, 4(3), 291-297.

Isemann, D., & Ahmad, K. (2014). Ontological access to images of fine art. *Journal on Computing and Cultural Heritage (JOCCH)*, 7(1), 3.

Johansen, 1988 in Baecker, R. M. (Ed.). (2014). *Readings in Human-Computer Interaction: Toward the Year 2000*. Morgan Kaufmann.

Johnson, P. E., Grazioli, S., Jamal, K., & Glen Berryman, R. (2001). Detecting deception: adversarial problem solving in a low base-rate world. *Cognitive Science*, 25(3), 355-392.

Kamar, E., Hacker, S., & Horvitz, E. (2012, June). Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems* Volume 1 (pp. 467-474). International Foundation for Autonomous Agents and Multiagent Systems.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1), 49-79.

Kittur, A., Chi, E., Pendleton, B. A., Suh, B., & Mytkowicz, T. (2007). Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World Wide Web*, 1(2), 19.

Kok, S., & Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Machine Learning and Knowledge Discovery in Databases* (pp. 624-639). Springer Berlin Heidelberg.

Koller, D., Frischer, B., & Humphreys, G. (2009). Research challenges for digital archives of 3D cultural heritage models. *Journal on Computing and Cultural Heritage (JOCCH)*, *2*(3), 7.

Kosorukoff, A. (2001). Human based genetic algorithm. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on* (Vol. 5, pp. 3464-3469). IEEE.

Kubicek, H., & Wagner, R. M. (2002). Community networks in a generational perspective: The change of an electronic medium within three decades. Information, *Communication & Society*, 5(3), 291-319.

Lawless, S., Agosti, M., Clough, P., & Conlan, O. (2013, July). Exploration, navigation and retrieval of information in cultural heritage: ENRICH 2013. In P*roceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1136-1136). ACM.

Lawless, S., Agosti, M., Conlan, O., & Clough, P. (2013, December). ENRICH 2013: the first workshop on the exploration, navigation and retrieval of information in cultural heritage. In *ACM SIGIR Forum* (Vol. 47, No. 2, pp. 68-73). ACM.

Leavitt, A., & Clark, J. A. (2014, April). Upvoting hurricane Sandy: event-based news production processes on a social news site. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1495-1504). ACM.

Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1), 1-19.

Li, X., Snoek, C. G., & Worring, M. (2009). Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7), 1310-1322.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*.

Ling, K., Beenen, G., Ludford, P., Wang, X., Chang, K., Li, X., ... & Resnick, P. (2005). Using social psychology to motivate contributions to online communities. *Journal of Computer‐Mediated Communication*, 10(4), 00-00.

Liu, D., Hua, X. S., Yang, L., Wang, M., & Zhang, H. J. (2009, April). Tag ranking. In *Proceedings of the 18th International Conference on World Wide Web* (pp. 351-360). ACM.

Liu, Y., Zhang, D., Lu, G., & Ma, W. Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1), 262-282.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook* (pp. 73-105). Springer US.

Lu, C., Hu, X., & Park, J. R. (2011). Exploiting the social tagging network for Web clustering. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(5), 840-852.

Lughofer, E., Smith, J. E., Tahir, M. A., Caleb-Solly, P., Eitzinger, C., Sannen, D., & Nuttin, M. (2009). Human–machine interaction issues in quality control based on online image classification. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(5), 960-971.

Lusenet, Y. D. (2007). Tending the garden or harvesting the fields: digital preservation and the UNESCO charter on the preservation of the digital heritage. *Library Trends*, 56(1), 164-182.

Lynch, C. (2002). Digital collections, digital libraries & the digitization of cultural heritage information. *Microform & Imaging Review*, 31(4), 131-145.

Lynch, C. (2005). *Where do we go from here? The next decade for digital libraries.* In D-Lib Magazine, 11(7/8).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.

Marchetti, A., Tesconi, M., Ronzano, F., Rosella, M., & Minutoli, S. (2007, May). Semkey: A semantic collaborative tagging system. In *Workshop on Tagging and Metadata for Social Information Organization at WWW* (Vol. 7, pp. 8-12).

Marshall, C. C. (2009, June). No bull, no spin: a comparison of tags with other forms of user metadata. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 241-250). ACM.

Martens, G., Lambert, P., & Van de Walle, R. (2010). *Bridging the Semantic Gap using Human Vision System Inspired Features* (pp. 261-276). In-Tech.

McParlane, P. J., Moshfeghi, Y., & Jose, J. M. (2014, January). Collections for Automatic Image Annotation and Photo Tag Recommendation. In *MultiMedia Modeling* (pp. 133-145). Springer International Publishing.

Meade, M. L., Nokes, T. J., & Morrow, D. G. (2009). Expertise promotes facilitation on a collaborative memory task. *Memory*, 17(1), 39-48.

Microsoft. (n.d.). *Microsoft OneDrive*. Retrieved on 18 February 2016 from https://onedrive.live.com/about/en-au/

Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI* (Vol. 6, pp. 775-780).

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.

Morrison, P. J. (2008). Tagging and searching: Search retrieval effectiveness of folksonomies on the World Wide Web. *Information Processing & Management*, 44(4), 1562-1579.

Mortara, M., Catalano, C. E., Bellotti, F., Fiucci, G., Houry-Panchetti, M., & Petridis, P. (2014). Learning cultural heritage by serious games. *Journal of Cultural Heritage,* 15(3), 318-325.

Mudgea, M., Ashleyb, M., & Schroer, C. (2007). *A digital future for cultural heritage*. Available in: http://culturalheritageimaging. org/What_We_Do/Publications/cipa2007/CIPA_2007. pdf.

National Trust. (n.d.). *National Trust, About Us*. Retrieved from https://www.nationaltrust.org.au/about-us/

Noll, M. G., Au Yeung, C. M., Gibbins, N., Meinel, C., & Shadbolt, N. (2009, July). Telling experts from spammers: expertise ranking in folksonomies. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 612-619). ACM.

Nov, O., & Wattal, S. (2009, April). Social computing privacy concerns: antecedents and effects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 333-336). ACM.

Nwana, H. S. (1996). Software agents: An overview. *The Knowledge Engineering Review*, 11(03), 205-244.

Odell, J., Parunak, H. V. D., Brueckner, S., & Sauter, J. A. (2003). Changing Roles: Dynamic Role Assignment. *Journal of Object Technology*, 2(5), 77-86.

Oinas-Kukkonen, H., Lyytinen, K., & Yoo, Y. (2010). Social networks and information systems: ongoing and future research streams. *Journal of the Association for Information Systems*, 11(2), 61-68.

Oomen, J., & Aroyo, L. (2011, June). Crowdsourcing in the cultural heritage domain: opportunities and challenges. In *Proceedings of the 5th International Conference on Communities and Technologies* (pp. 138-149). ACM.

O'reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & Strategies*, (1), 17.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: bringing order to the web.* Technical Report. Stanford InfoLab.

Pantic, M., Pentland, A., Nijholt, A., & Huang, T. S. (2007). Human computing and machine understanding of human behavior: a survey. In A*rtificial Intelligence for Human Computing* (pp. 47-71). Springer Berlin Heidelberg.

Paquet, E., El-Hakim, S., Beraldin, A., & Peters, S. (2001). The virtual museum: Virtualisation of real historical environments and artefacts and three-dimensional shape-based searching. In *Virtual and Augmented Architecture (VAA'01)* (pp. 183-193). Springer London.

Parameswaran, M., & Whinston, A. B. (2007). Research issues in social computing. *Journal of the Association for Information Systems*, 8(6), 336-350.

PastPerfect Software. (n.d.). *PastPerfect 5.0*. Retrieved 18 February 2016 from http://www.museumsoftware.com/pp5.html

Paşca, M., Lin, D., Bigham, J., Lifchits, A., & Jain, A. (2006, July). Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 809-816). Association for Computational Linguistics.

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004, May). WordNet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004* (pp. 38-41). Association for Computational Linguistics.

Petras, V., Bogers, T., Toms, E., Hall, M., Savoy, J., Malak, P., ... & Masiero, I. (2013). Cultural heritage in clef (chic) 2013. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization* (pp. 192-211). Springer Berlin Heidelberg.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130-137.

Powers, D. M. (2011). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.* Journal of Machine Learning Technologies, 2(1), 37-63.

Quan-Haase, A., & Martin, K. (2012, February). Digital humanities: the continuing role of serendipity in historical research. In *Proceedings of the 2012 iConference* (pp. 456-458). ACM.

Quinn, A. J., & Bederson, B. B. (2011, May). Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1403-1412). ACM.

Rae, A., Sigurbjörnsson, B., & van Zwol, R. (2010, April). Improving tag recommendation using social networks. In *Adaptivity, Personalization and Fusion of Heterogeneous Information* (pp. 92-99).

Rails Guides. (n.d.). *Rails Routing from the Outside In - Ruby on Rails Guides*. Retrieved on 18 February 2016 from http://guides.rubyonrails.org/routing.html

RMagick. (n.d.). *Ruby Bindings for ImageMagick*. Retrieved on 18 February 2016 from https://github.com/rmagick/rmagick

Rolla, P. J. (2011). User tags versus subject headings. *Library Resources & Technical Services*, 53(3), 174-184.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web* (pp. 851-860). ACM.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523.

Sangiorgi, U. B., Beuvens, F., & Vanderdonckt, J. (2012, June). User interface design by collaborative sketching. In *Proceedings of the Designing Interactive Systems Conference* (pp. 378-387). ACM.

Santos Jr, E., & Johnson Jr, G. (2004, August). Toward detecting deception in intelligent systems. In *Defense and Security* (pp. 130-141). International Society for Optics and Photonics.

Santos, E., & Li, D. (2010). On deception detection in multiagent systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 40(2), 224-235.

Schaffert, S., Bry, F., Baumeister, J., & Kiesel, M. (2008). Semantic wikis. *Software, IEEE*, 25(4), 8-11.

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002, August). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 253-260). ACM.

Schirner, G., Erdogmus, D., Chowdhury, K., & Padir, T. (2013). The future of human-in-the-loop cyber-physical systems. *Computer*, (1), 36-45.

Schuler, D. (1998). Designing across borders: The community design of community networks. In *Proceedings of the CSCW'98 Conference on Computer Supported Cooperative Work, (Workshops)* (Vol. 416).

Sculpteur. (2005, April 14). *Sculpteur Events and Publications*. Retrieved on March 2, 2016 from http://www.sculpteurweb.org/html/3Dmodels.html

Semeraro, G., Lops, P., De Gemmis, M., Musto, C., & Narducci, F. (2012). A folksonomy-based recommender system for personalized access to digital artworks. *Journal on Computing and Cultural Heritage (JOCCH),* 5(3), 11.

Shahaf, D., & Amir, E. (2007, March). Towards a Theory of AI Completeness. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning* (pp. 150-155).

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. *Recommender Systems Handbook*, 257-297.

Sheth, A., & Nagarajan, M. (2009). Semantics-empowered social computing. *Internet Computing, IEEE,* 13(1), 76-80.

Shtykh, R. Y., & Jin, Q. (2011). A human-centric integrated approach to web information search and sharing. *Human-Centric Computing and Information Sciences*, 1(1), 1-37.

Sigurbjörnsson, B., & Van Zwol, R. (2008, April). Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 327-336). ACM.

Sinclair, P., Lewis, P., Martinez, K., Addis, M., Prideaux, D., Fina, D., & Bormida, G. D. (2005, November). eCHASE: sustainable exploitation of electronic cultural heritage. In *Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the* (Ref. No. 2005/11099) (pp. 411-417). IET.

Singhal, A. (2012, May 16). Introducing the Knowledge Graph: Things, not Strings. Retrieved on 21 February 2016 from https://googleblog.blogspot.com.au/2012/05/introducing-knowledge-graph-things-not.html

Singhal, A., Buckley, C., & Mitra, M. (1996, August). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 21-29). ACM.

Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4), 35-43.

Sowa, J. F. (2006). Semantic networks. *Encyclopedia of Cognitive Science*.

Sparrow, B., Liu, J., & Wegner, D. M. (2011). Google effects on memory: Cognitive consequences of having information at our fingertips. *Science*, 333(6043), 776-778.

Stefani, C., Busayarat, C., Lombardo, J., Luca, L. D., & Véron, P. (2013). A web platform for the consultation of spatialized and semantically enriched iconographic sources on cultural heritage buildings. *Journal on Computing and Cultural Heritage (JOCCH)*, 6(3), 13.

Sun, A., Bhowmick, S. S., & Chong, J. A. (2011, November). Social image tag recommendation by concept matching. In *Proceedings of the 19th ACM International Conference on Multimedia* (pp. 1181-1184). ACM.

Suominen, J., & Sivula, A. (2013). Gaming legacy? Four approaches to the relation between cultural heritage and digital technology. *Journal on Computing and Cultural Heritage (JOCCH)*, 6(3), 12.

Sutanto, J., Kankanhalli, A., & Tan, B. C. Y. (2011). Eliciting a sense of virtual community among knowledge contributors. *ACM Transactions on Management Information Systems (TMIS)*, 2(3), 14.

Sweetnam, M. S., Agosti, M., Orio, N., Ponchia, C., Steiner, C. M., Hillemann, E. C., ... & Lawless, S. (2012). User needs for enhanced engagement with cultural heritage collections. In *Theory and Practice of Digital Libraries* (pp. 64-75). Springer Berlin Heidelberg.

Szabo, G., & Huberman, B. A. (2010). Predicting the popularity of online content. *Communications of the ACM*, 53(8), 80-88.

Thong, J. Y., Hong, W., & Tam, K. Y. (2002). Understanding user acceptance of digital libraries: what are the roles of interface characteristics, organizational context, and individual differences?. *International Journal of Human-Computer Studies*, *57*(3), 215-242.

Tran, Q. N. N., & Low, G. (2008). MOBMAS: A methodology for ontology-based multi-agent systems development. *Information and Software Technology*, 50(7), 697-722.

Truong, B. Q., Sun, A., & Bhowmick, S. S. (2012, June). Content is still king: the effect of neighbor voting schemes on tag relevance for social image retrieval. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval* (p. 9). ACM.

Tuzhilin, A. (2011). Knowledge management revisited: Old dogs, new tricks. *ACM Transactions on Management Information Systems (TMIS)*, 2(3), 13.

UNESCO. (2003, October 17). *UNESCO Charter on the Preservation of Digital Heritage*. Retrieved from http://portal.unesco.org/ci/en/files/13367/10700115911Charter_en.pdf/Charter_en.pdf

University of New England. (n.d.). A*bout the Heritage Centre - University of New England*. Retrieved on 19 February 2016 from https://www.une.edu.au/library/heritage-centre/heritage-centre

van der Hoek, W., & Wooldridge, M. (2008). Multi-agent systems. *Foundations of Artificial Intelligence*, 3, 887-928.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186-204.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 425-478.

Vijayanarasimhan, S., & Grauman, K. (2009, June). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 2262-2269). IEEE.

Villa, R., & Halvey, M. (2013, July). Is relevance hard work?: evaluating the effort of making relevant assessments. In *Proceedings of the 36th international ACM*

SIGIR Conference on Research and Development in Information Retrieval (pp. 765-768). ACM.

W3C. (2001a). *RDF 1.1 Primer*. Retrieved from on February 27 2016 from https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/

W3C. (2001b). *RDF Examples and Miscellaneous Tests*. Retrieved from https://www.w3.org/2000/10/rdf-tests/

W3C. (2012). *OWL 2 Web Ontology Language Primer (Second Edition)*. Retrieved on February 27 2016 from https://www.w3.org/TR/owl2-primer/

Waibel, A., Stiefelhagen, R., Carlson, R., Casas, J., Kleindienst, J., Lamel, L., ... & Terken, J. (2010). Computers in the human interaction loop. In *Handbook of Ambient Intelligence and Smart Environments* (pp. 1071-1116). Springer US.

Wang, C., Zhang, L., & Zhang, H. J. (2008, July). Learning to reduce the semantic gap in web image retrieval and annotation. In *Proceedings of the 31st annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 355-362). ACM.

Wang, F. Y., Carley, K. M., Zeng, D., & Mao, W. (2007). Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2), 79-83.

Wang, M., Ni, B., Hua, X. S., & Chua, T. S. (2012). Assistive tagging: A survey of multimedia tagging with human-computer joint exploration. *ACM Computing Surveys (CSUR)*, 44(4), 25.

Wang, Y., & Fesenmaier, D. R. (2003). Assessing motivation of contribution in online communities: An empirical investigation of an online travel community. *Electronic Markets*, 13(1), 33-45.

Webb, C., & Canberra National Library of Australia. (2003). *Guidelines for the Preservation of Digital Heritage.* Information Society Division, United Nations Educational, Scientific and Cultural Organization.

Wei, W., & Ram, S. (2012). Using a Network Analysis Approach for Organizing Social Bookmarking Tags and Enabling Web Content Discovery. *ACM Transactions on Management Information Systems (TMIS)*, 3(3), 15.

Wiggins, A. (2012). The Twelve Factor App. Retrieved on February 26 2016 from http://12factor.net/

Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of Information Science*, 18(1), 45-55.

Wojtinnek, P. R., Pulman, S., & Völker, J. (2012). Building semantic networks from plain text and Wikipedia with application to semantic relatedness and noun compound paraphrasing. *International Journal of Semantic Computing*,6(01), 67-91.

Wu, L., Yang, L., Yu, N., & Hua, X. S. (2009, April). Learning to tag. In *Proceedings of the 18th International Conference on World Wide Web* (pp. 361-370). ACM.

Yang, Y., Pierce, T., & Carbonell, J. (1998, August). A study of retrospective and on-line event detection. In *Proceedings of the 21st Annual International ACM SIGIR*

Conference on Research and Development in Information Retrieval(pp. 28-36). ACM.

Yin, H., Cui, B., Li, J., Yao, J., & Chen, C. (2012). Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9), 896-907.

Yu, C. H., & Hunter, J. (2013). Documenting and sharing comparative analyses of 3D digital museum artifacts through semantic web annotations. *Journal on Computing and Cultural Heritage (JOCCH)*, 6(4), 18.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.

Zhang, D., Guo, B., & Yu, Z. (2011). The emergence of social and community intelligence. *Computer*, (7), 21-28.

Zhao, R., & Grosky, W. I. (2002). Bridging the semantic gap in image retrieval. *Distributed Multimedia Databases: Techniques and Applications*, 14-36.

Zhao, S., Gao, Y., Jiang, X., Yao, H., Chua, T. S., & Sun, X. (2014, November). Exploring principles-of-art features for image emotion recognition. In *Proceedings of the ACM International Conference on Multimedia* (pp. 47-56). ACM.

Zhao, W. L., Wu, X., & Ngo, C. W. (2010). On the annotation of web videos by efficient near-duplicate search. *Multimedia, IEEE Transactions on*, 12(5), 448-461.

Zhu, H. H., & Wu, H. (2009). Sloppy Tags and Metacrap? Quality of User Contributed Tags in Collaborative Social Tagging Systems. *Quality of User Contributed Tags in Collaborative Social Tagging Systems* (May 21, 2009).

Zhu, H., & Zhou, M. (2008). Roles in information systems: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(3), 377-396.

Appendix A: Suggestion Performance Results

The following tables summarise the performance evaluation results obtained by the SAGA, Vote, VotePlus, Sum and SumPlus algorithms using a binary classifier on suggestions. Evaluations were performed at 20%, 40%, 60% and 80% of the original data being preserved as training data and the remaining % being used for verification. The source data was the MIRFlickr25K collection, consisting of popular tags applied to 25,000 images with permissive usage rights sourced from Flickr.

SAGA				
	20%	**40%**	**60%**	**80%**
Precision	0.25	0.34	0.35	0.34
Recall	0.93	0.97	0.97	1.0
F0.5	0.3	0.38	0.39	0.39
F1	0.39	0.48	0.49	0.49
F2	0.59	0.67	0.68	0.67
Phi	0.32	0.43	0.45	0.46
Precision@5	0.7	0.68	0.63	0.58
Success@1	0.93	0.97	0.97	1.0
Success@5	0.93	0.97	0.97	1.0
MRR	0.93	0.97	0.97	1.0

Vote				
	20%	**40%**	**60%**	**80%**
Precision	0.27	0.29	0.28	0.25
Recall	0.93	0.97	0.97	1.0
F0.5	0.31	0.33	0.32	0.29
F1	0.41	0.43	0.42	0.39
F2	0.61	0.63	0.62	0.59
Phi	0.35	0.36	0.36	0.34
Precision@5	0.34	0.35	0.32	0.31
Success@1	0.07	0.33	0.33	0.23
Success@5	0.87	0.77	0.8	0.7
MRR	0.34	0.49	0.49	0.44

VotePlus				
	20%	**40%**	**60%**	**80%**
Precision	0.16	0.17	0.16	0.15
Recall	0.93	0.97	0.97	1.0
F0.5	0.19	0.2	0.19	0.18
F1	0.27	0.28	0.27	0.25
F2	0.46	0.48	0.46	0.44
Phi	0.02	0.0	0.0	0.0
Precision@5	0.61	0.57	0.59	0.56
Success@1	0.93	0.93	0.93	0.93
Success@5	0.93	0.97	0.97	1.0
MRR	0.93	0.95	0.95	0.96

Sum				
	20%	**40%**	**60%**	**80%**
Precision	0.27	0.29	0.28	0.25
Recall	0.93	0.97	0.97	1.0
F0.5	0.31	0.33	0.32	0.29
F1	0.41	0.43	0.42	0.39
F2	0.61	0.63	0.62	0.59
Phi	0.35	0.36	0.36	0.34
Precision@5	0.71	0.71	0.67	0.62
Success@1	0.93	0.97	0.97	1.0
Success@5	0.93	0.97	0.97	1.0
MRR	0.93	0.97	0.97	1.0

SumPlus				
	20%	**40%**	**60%**	**80%**
Precision	0.16	0.17	0.16	0.15
Recall	0.93	0.97	0.97	1.0
F0.5	0.19	0.2	0.19	0.18
F1	0.27	0.28	0.27	0.25
F2	0.46	0.48	0.46	0.44
Phi	0.02	0.0	0.0	0.0
Precision@5	0.71	0.71	0.7	0.63
Success@1	0.93	0.97	0.97	1.0
Success@5	0.93	0.97	0.97	1.0
MRR	0.93	0.97	0.97	1.0

Appendix B: Participant Usability Survey Form

SAGE Participant Usability Survey

This usability survey aims to determine the usability and usefulness of the SAGE web application.

*Required

1. **Select the statement about your experience level in using the SAGE application which you feel most accurately represents your ability. ***
   Mark only one oval.

   - I can perform simple tasks in SAGE, but I'd like to have someone available who can help me if I get stuck.
   - I can comfortably perform a variety of tasks in SAGE, and I could generally work without supervision.
   - I can confidently perform a variety of tasks in SAGE, and I'd be able to help a new user learn how to use it.

Part 1. Technology Acceptance

Imagine you have been employed in a job to organise, explore and locate images within an image collection. SAGE is available for you to use in this workplace. Based on this information, provide a rating (1-5) for each statement.

2. **Using SAGE in my job would enable me to accomplish tasks more quickly. ***
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

3. **Using SAGE would improve my job performance. ***
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

4. **Using SAGE in my job would increase my productivity. ***
   Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

5. **Using SAGE would enhance my effectiveness on the job.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

6. **Using SAGE would make it easier to do my job.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

7. **I would find SAGE useful in my job.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. **Learning to operate SAGE would be easy for me.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

9. **I would find it easy to get SAGE to do what I want it to do.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

10. **My interaction with SAGE would be clear and understandable.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

11. **I would find SAGE to be flexible to interact with.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

12. **It would be easy for me to become skillful at using SAGE.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

13. **I would find SAGE easy to use.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

Part 2. Software Usability

Provide a rating (1-5) for each statement based on your experience of using SAGE.

14. **I think that I would like to use SAGE frequently.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

15. **I found SAGE unnecessarily complex.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

16. **I thought SAGE was easy to use.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

17. **I think that I would need the support of a technical person to be able to use SAGE.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

18. **I found the various functions in SAGE were well integrated.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

19. **I thought there was too much inconsistency in SAGE.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

20. **I would imagine that most people would learn to use SAGE very quickly.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

21. **I found SAGE very cumbersome to use.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

22. **I felt very confident using SAGE.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

23. **I needed to learn a lot of things before I could get going with SAGE.** *
Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	⬭	⬭	⬭	⬭	⬭	Strongly Agree

24. **What did you feel were the most positive aspects of using SAGE?**

25. **What did you feel were the aspects of SAGE that need improvement?**

....

....

....

....

....

26. **Do you have comments about your responses to any of the above questions?**

....

....

....

....

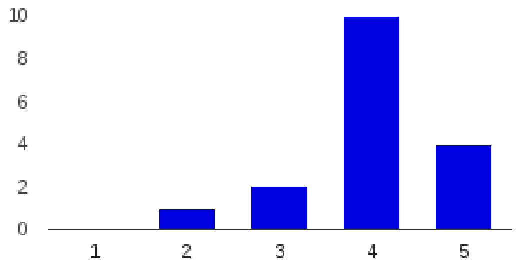
....

Powered by
Google Forms

Appendix C: Acceptance Testing Group Survey Responses

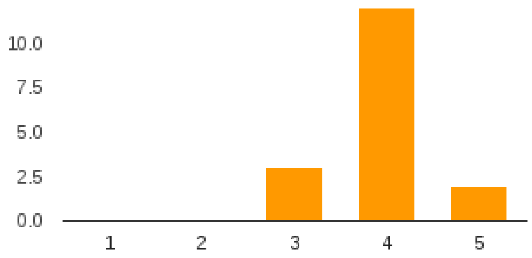
Part 1. Technology Acceptance

Using SAGE in my job would enable me to accomplish tasks more quickly.



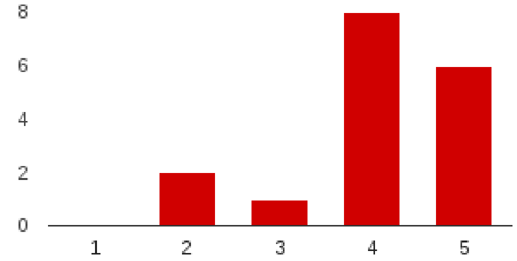
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**2**	11.8%
4	**10**	58.8%
Strongly Agree: 5	**4**	23.5%

Using SAGE would improve my job performance.



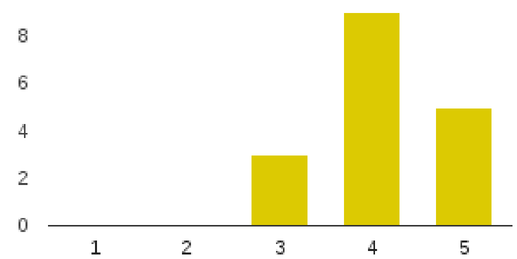
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**3**	17.6%
4	**12**	70.6%
Strongly Agree: 5	**2**	11.8%

Using SAGE in my job would increase my productivity.



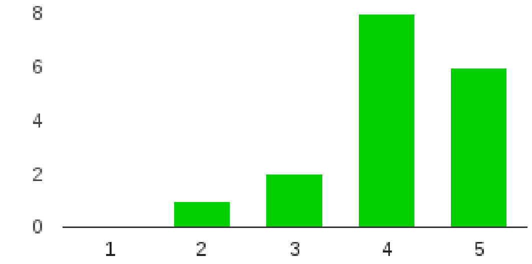
Strongly Disagree: 1	**0**	0%
2	**2**	11.8%
3	**1**	5.9%
4	**8**	47.1%
Strongly Agree: 5	**6**	35.3%

Using SAGE would enhance my effectiveness on the job.



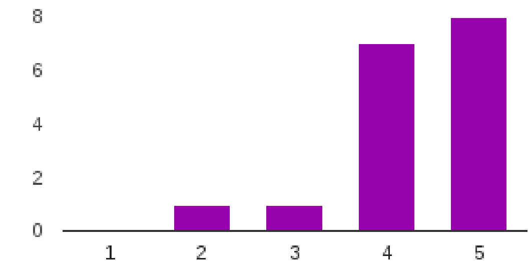
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**3**	17.6%
4	**9**	52.9%
Strongly Agree: 5	**5**	29.4%

Using SAGE would make it easier to do my job.



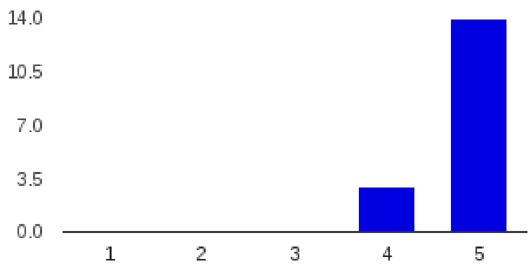
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**2**	11.8%
4	**8**	47.1%
Strongly Agree: 5	**6**	35.3%

I would find SAGE useful in my job.



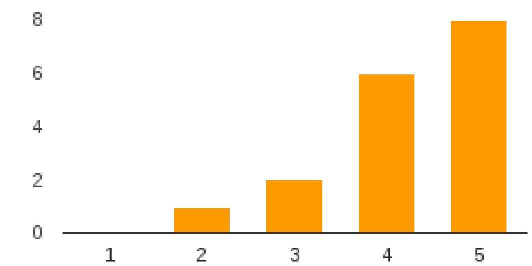
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**1**	5.9%
4	**7**	41.2%
Strongly Agree: 5	**8**	47.1%

Learning to operate SAGE would be easy for me.



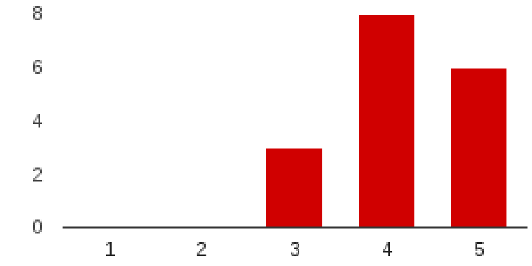
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**3**	17.6%
Strongly Agree: 5	**14**	82.4%

I would find it easy to get SAGE to do what I want it to do.



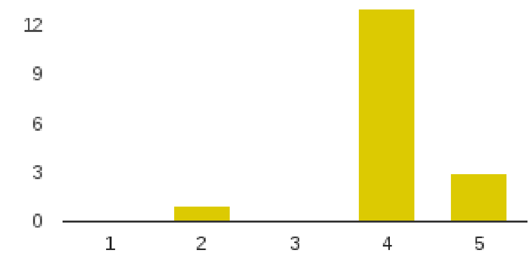
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**2**	11.8%
4	**6**	35.3%
Strongly Agree: 5	**8**	47.1%

My interaction with SAGE would be clear and understandable.



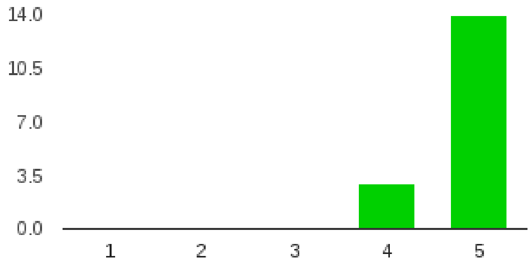
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**3**	17.6%
4	**8**	47.1%
Strongly Agree: 5	**6**	35.3%

I would find SAGE to be flexible to interact with.



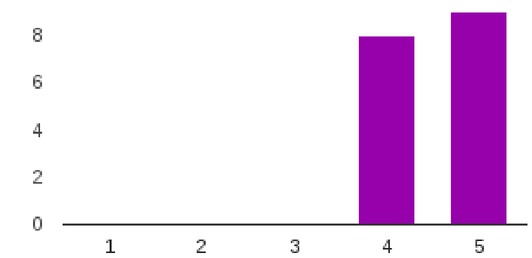
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**0**	0%
4	**13**	76.5%
Strongly Agree: 5	**3**	17.6%

It would be easy for me to become skillful at using SAGE.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**3**	17.6%
Strongly Agree: 5	**14**	82.4%

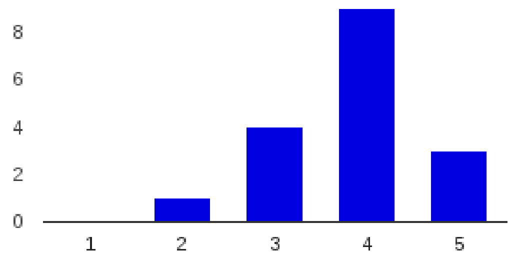
I would find SAGE easy to use.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**8**	47.1%
Strongly Agree: 5	**9**	52.9%

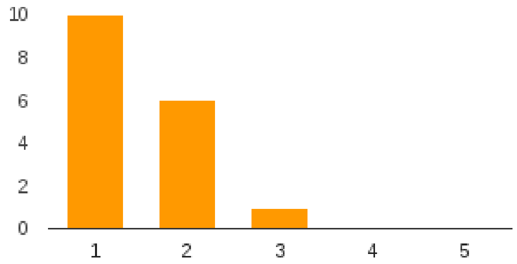
Part 2. Software Usability

I think that I would like to use SAGE frequently.



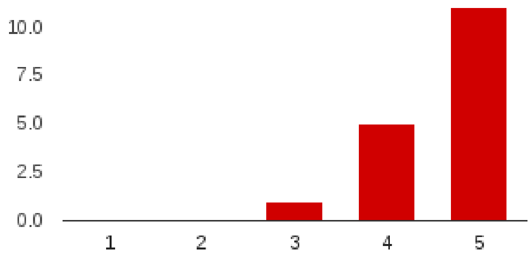
Strongly Disagree: 1	**0**	0%
2	**1**	5.9%
3	**4**	23.5%
4	**9**	52.9%
Strongly Agree: 5	**3**	17.6%

I found SAGE unnecessarily complex.



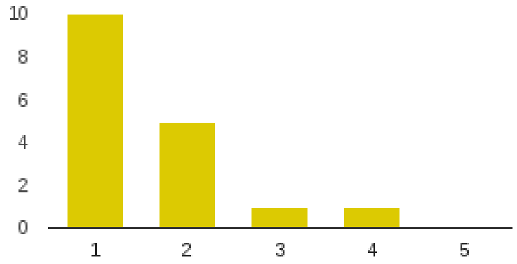
Strongly Disagree: 1	**10**	58.8%
2	**6**	35.3%
3	**1**	5.9%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I thought SAGE was easy to use.



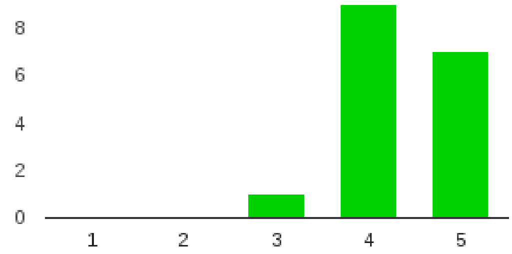
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	5.9%
4	**5**	29.4%
Strongly Agree: 5	**11**	64.7%

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



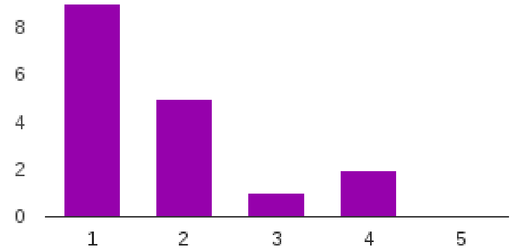
Strongly Disagree: 1	**10**	58.8%
2	**5**	29.4%
3	**1**	5.9%
4	**1**	5.9%
Strongly Agree: 5	**0**	0%

I found the various functions in SAGE were well integrated.



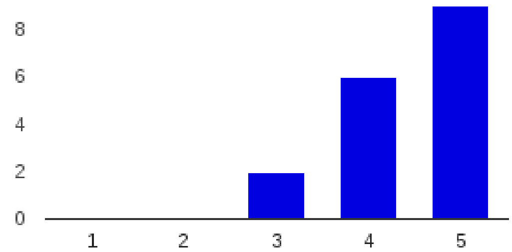
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	5.9%
4	**9**	52.9%
Strongly Agree: 5	**7**	41.2%

I thought there was too much inconsistency in SAGE.



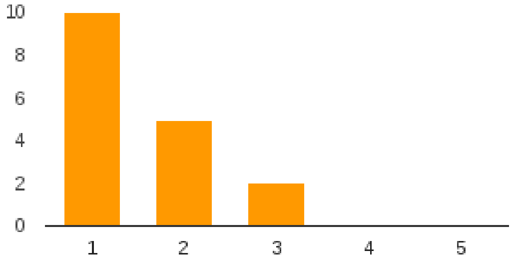
Strongly Disagree: 1	**9**	52.9%
2	**5**	29.4%
3	**1**	5.9%
4	**2**	11.8%
Strongly Agree: 5	**0**	0%

I would imagine that most people would learn to use SAGE very quickly.



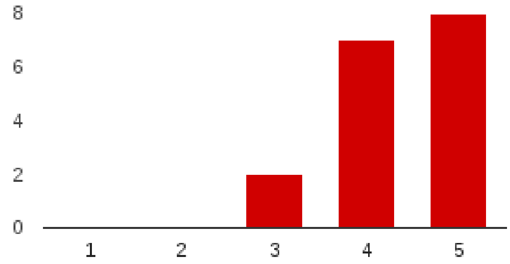
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**2**	11.8%
4	**6**	35.3%
Strongly Agree: 5	**9**	52.9%

I found SAGE very cumbersome to use.



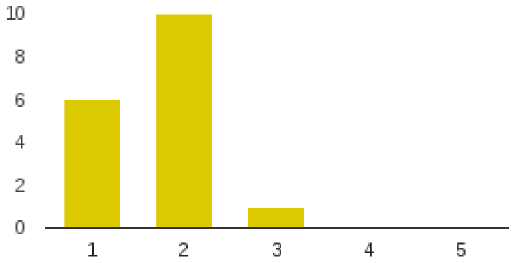
Strongly Disagree: 1	**10**	58.8%
2	**5**	29.4%
3	**2**	11.8%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I felt very confident using SAGE.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**2**	11.8%
4	**7**	41.2%
Strongly Agree: 5	**8**	47.1%

I needed to learn a lot of things before I could get going with SAGE.

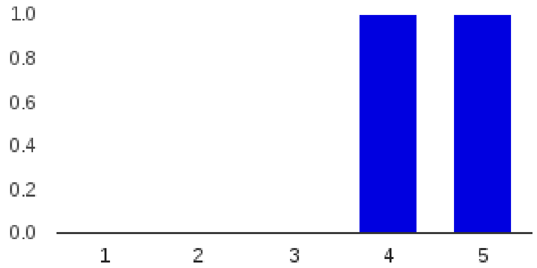


Strongly Disagree: 1	**6**	35.3%
2	**10**	58.8%
3	**1**	5.9%
4	**0**	0%
Strongly Agree: 5	**0**	0%

Appendix D: Heritage Centre Group Survey Responses

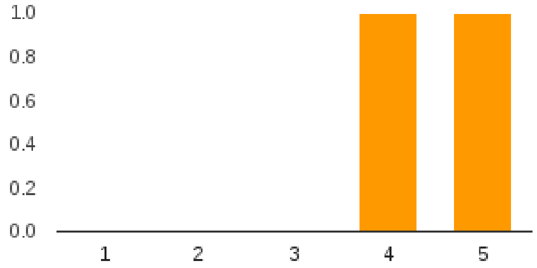
Part 1. Technology Acceptance

Using SAGE in my job would enable me to accomplish tasks more quickly.



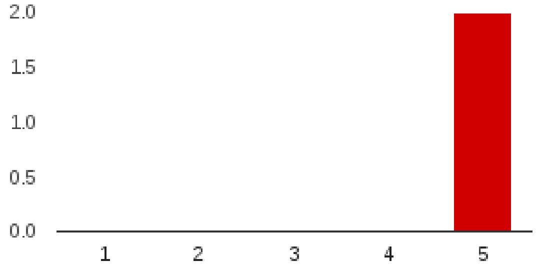
Strongly Disagree:	1	**0**	0%
	2	**0**	0%
	3	**0**	0%
	4	**1**	50%
Strongly Agree:	5	**1**	50%

Using SAGE would improve my job performance.



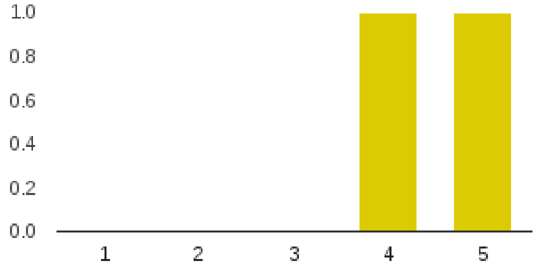
Strongly Disagree:	1	**0**	0%
	2	**0**	0%
	3	**0**	0%
	4	**1**	50%
Strongly Agree:	5	**1**	50%

Using SAGE in my job would increase my productivity.



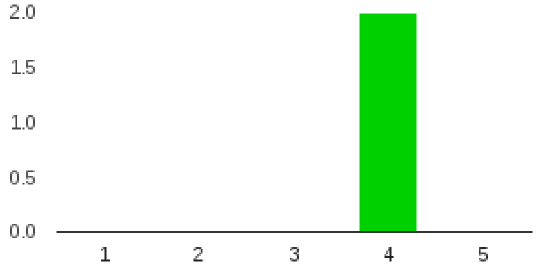
Strongly Disagree:	1	**0**	0%
	2	**0**	0%
	3	**0**	0%
	4	**0**	0%
Strongly Agree:	5	**2**	100%

Using SAGE would enhance my effectiveness on the job.



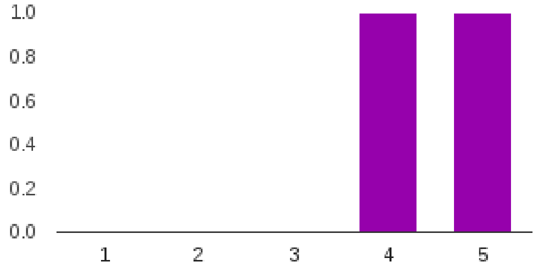
Strongly Disagree:	1	**0**	0%
	2	**0**	0%
	3	**0**	0%
	4	**1**	50%
Strongly Agree:	5	**1**	50%

Using SAGE would make it easier to do my job.



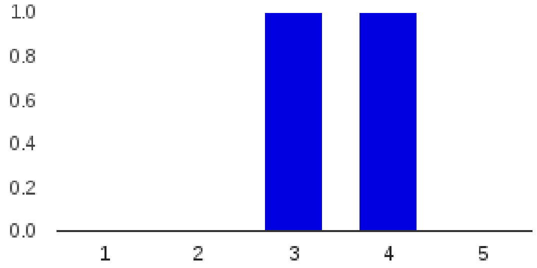
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**2**	100%
Strongly Agree: 5	**0**	0%

I would find SAGE useful in my job.



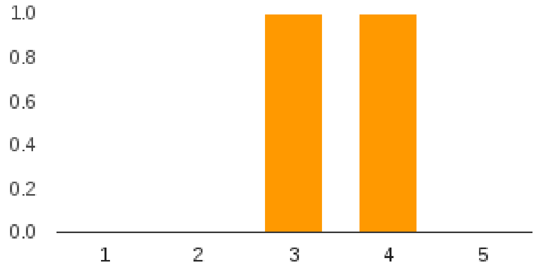
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**1**	50%
Strongly Agree: 5	**1**	50%

Learning to operate SAGE would be easy for me.



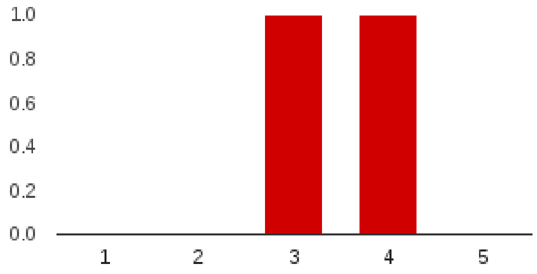
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**1**	50%
Strongly Agree: 5	**0**	0%

I would find it easy to get SAGE to do what I want it to do.



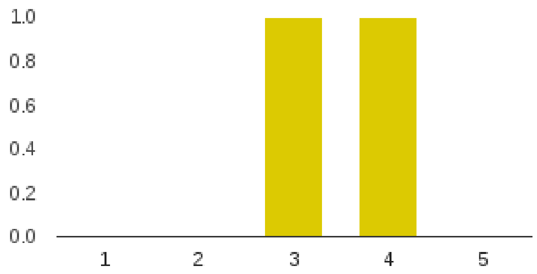
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**1**	50%
Strongly Agree: 5	**0**	0%

My interaction with SAGE would be clear and understandable.



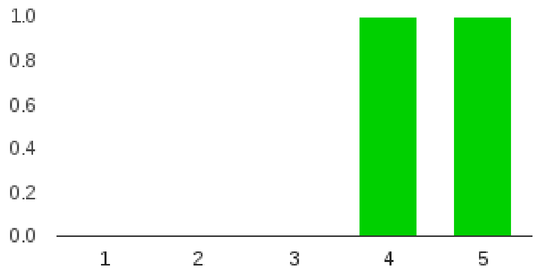
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**1**	50%
Strongly Agree: 5	**0**	0%

I would find SAGE to be flexible to interact with.



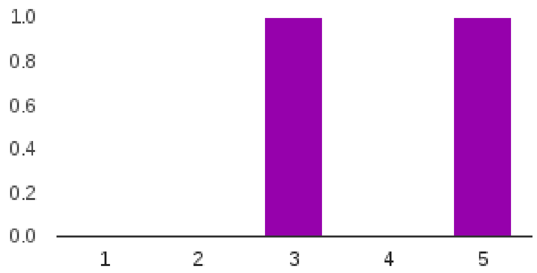
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**1**	50%
Strongly Agree: 5	**0**	0%

It would be easy for me to become skillful at using SAGE.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**1**	50%
Strongly Agree: 5	**1**	50%

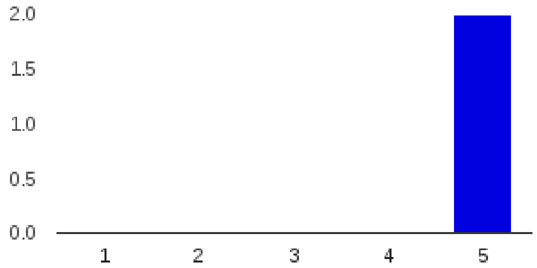
I would find SAGE easy to use.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**0**	0%
Strongly Agree: 5	**1**	50%

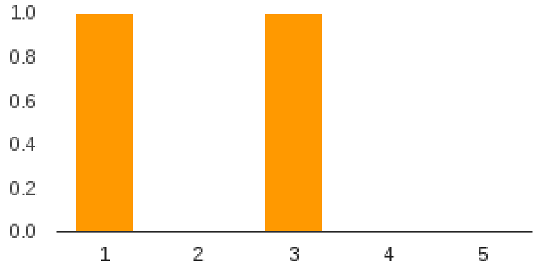
Part 2. Software Usability

I think that I would like to use SAGE frequently.



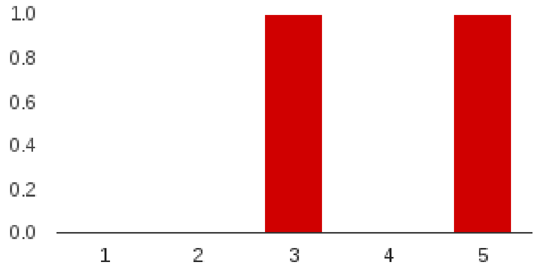
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**0**	0%
Strongly Agree: 5	**2**	100%

I found SAGE unnecessarily complex.



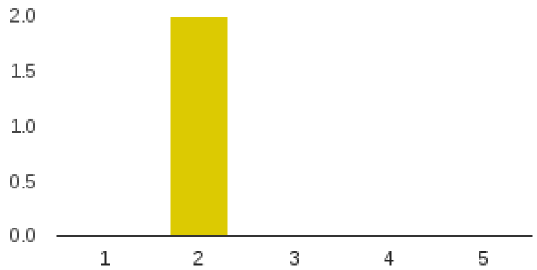
Strongly Disagree: 1	**1**	50%
2	**0**	0%
3	**1**	50%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I thought SAGE was easy to use.



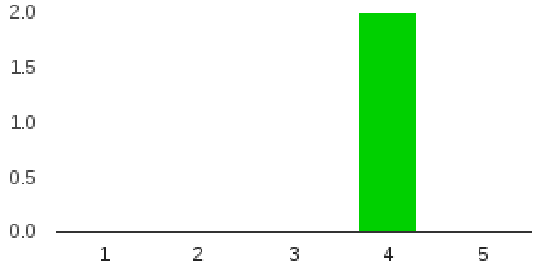
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**0**	0%
Strongly Agree: 5	**1**	50%

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



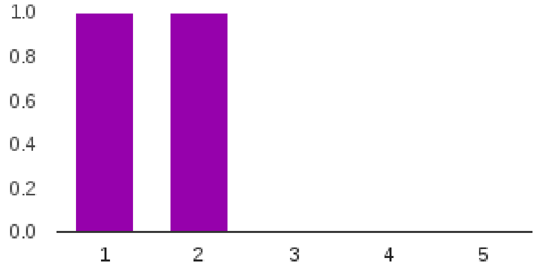
Strongly Disagree: 1	**0**	0%
2	**2**	100%
3	**0**	0%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I found the various functions in SAGE were well integrated.



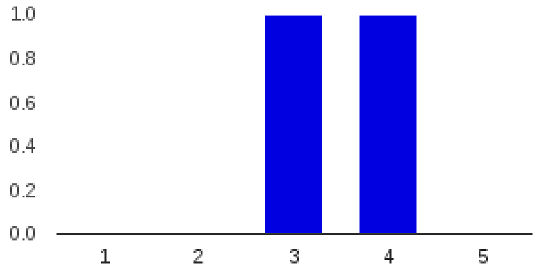
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**2**	100%
Strongly Agree: 5	**0**	0%

I thought there was too much inconsistency in SAGE.



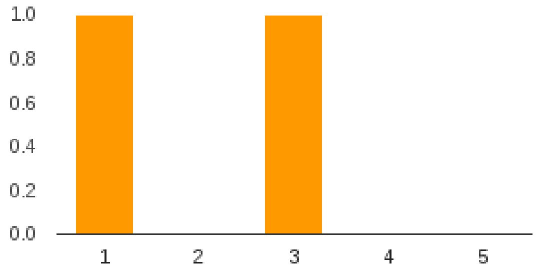
Strongly Disagree: 1	**1**	50%
2	**1**	50%
3	**0**	0%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I would imagine that most people would learn to use SAGE very quickly.



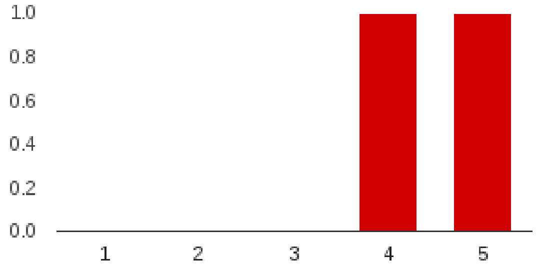
Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**1**	50%
4	**1**	50%
Strongly Agree: 5	**0**	0%

I found SAGE very cumbersome to use.



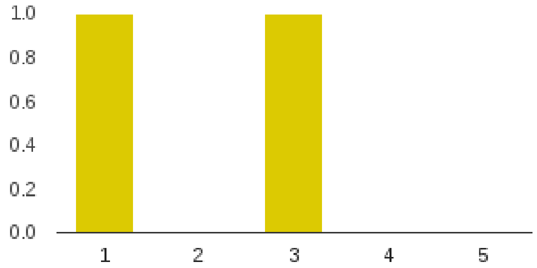
Strongly Disagree: 1	**1**	50%
2	**0**	0%
3	**1**	50%
4	**0**	0%
Strongly Agree: 5	**0**	0%

I felt very confident using SAGE.



Strongly Disagree: 1	**0**	0%
2	**0**	0%
3	**0**	0%
4	**1**	50%
Strongly Agree: 5	**1**	50%

I needed to learn a lot of things before I could get going with SAGE.



Strongly Disagree: 1	**1**	50%
2	**0**	0%
3	**1**	50%
4	**0**	0%
Strongly Agree: 5	**0**	0%

Appendix E: Object Annotation Levels in Heritage Projects

The following distribution table shows the number of objects that have n concepts as annotations, with 1 <= n <= n_max. This shows the extent to which objects in a project have been annotated.

Saumarez Combined

Number of Annotations (n)	Objects With n Annotations
1 concept	10
2 concepts	54
3 concepts	164
4 concepts	278
5 concepts	231
6 concepts	164
7 concepts	128
8 concepts	100
9 concepts	72
10 concepts	62
11 concepts	45
12 concepts	36
13 concepts	19
14 concepts	7
15 concepts	6
16 concepts	8
17 concepts	1
18 concepts	2
19 concepts	5

20 concepts	1
...	
22 concepts	2

Armidale in Photos

Number of Annotations (n)	**Objects With n Annotations**
1 concept	9
2 concepts	18
3 concepts	8
4 concepts	5
5 concepts	1
...	
7 concepts	1

Nickson AIF Photos

Number of Annotations (n)	**Objects With n Annotations**
1 concept	12
2 concepts	10
3 concepts	10
4 concepts	22
5 concepts	11
6 concepts	11
7 concepts	18
8 concepts	10
9 concepts	8
10 concepts	6
11 concepts	5
12 concepts	5
13 concepts	1
14 concepts	5
15 concepts	2
16 concepts	1
...	
23 concepts	1

Nursing VAD Identification

Number of Annotations (n)	**Objects With n Annotations**
1 concept	23
2 concepts	10
3 concepts	8
4 concepts	4
5 concepts	4
...	
19 concepts	1

Saumarez Gardens

Number of Annotations (n)	Objects With n Annotations
1 concept	91
2 concepts	60
3 concepts	127
4 concepts	99
5 concepts	51
6 concepts	14
7 concepts	5

Appendix F: Concept Annotation Levels in Heritage Projects

The following distribution table shows the number of concepts that have been applied to n objects as annotations, with 1 <= n <= n_max. This shows the extent to which concepts in a project have been used for annotation.

Saumarez Combined

Number of Annotations (n)	Concepts With n Annotations
1 object	556
2 objects	135
3 objects	82
4 objects	39
5 objects	18
6 objects	17
7 objects	14
8 objects	17
9 objects	11
10 objects	11
11 objects	6
12 objects	7
13 objects	2
14 objects	7
15 objects	6
...	
17 objects	9
18 objects	6
19 objects	3

20 objects	3
21 objects	5
22 objects	1
23 objects	1
24 objects	2
25 objects	2
26 objects	1
27 objects	1
...	
29 objects	1
30 objects	1
31 objects	1
32 objects	3
33 objects	1
34 objects	2
35 objects	2
...	
38 objects	2
39 objects	2
...	

41 objects	1
...	
44 objects	1
...	
46 objects	1
...	
53 objects	2
...	
55 objects	2
56 objects	2
...	
61 objects	1
...	
67 objects	1
...	
71 objects	1
72 objects	1
...	
74 objects	1
...	

76 objects	1
...	
78 objects	1
...	
93 objects	1
...	
96 objects	1
...	
98 objects	1
...	
103 objects	1
...	
112 objects	1
113 objects	1
...	
118 objects	1
119 objects	1
...	
156 objects	2
157 objects	1

...	
163 objects	1
...	
170 objects	1
...	
223 objects	1
224 objects	1
...	
243 objects	1
...	
263 objects	1
...	
297 objects	1
...	
309 objects	1
...	
328 objects	1
...	
611 objects	1

Armidale in Photos

Number of Annotations (n)	Concepts With n Annotations
1 object	14
2 objects	8
3 objects	2
4 objects	3
5 objects	2
...	
8 objects	1
...	
13 objects	1
...	
22 objects	1

Nickson AIF Photos

Number of Annotations (n)	Concepts With n Annotations
1 object	14
2 objects	8
3 objects	2
4 objects	3
5 objects	2
...	
8 objects	1
...	
13 objects	1
...	
22 objects	1

Nursing VAD Identification

Number of Annotations (n)	**Concepts With n Annotations**
1 object	23
2 objects	4
3 objects	5
4 objects	1
5 objects	1
6 objects	1
7 objects	1
...	
9 objects	1
10 objects	1
...	
35 objects	1

Saumarez Gardens

Number of Annotations (n)	Concepts With n Annotations
1 object	15
2 objects	2
3 objects	6
4 objects	3
5 objects	2
6 objects	2
7 objects	1
8 objects	4
9 objects	2
10 objects	1
11 objects	1
12 objects	2
13 objects	1
14 objects	2
...	
18 objects	1
19 objects	2
20 objects	1
...	

22 objects	1
...	
25 objects	1
...	
28 objects	1
29 objects	1
...	
32 objects	1
...	
34 objects	1
...	
45 objects	1
...	
50 objects	1
...	
64 objects	1
...	
72 objects	1
...	
83 objects	1

...	
88 objects	1
...	
232 objects	1
...	
268 objects	1

Appendix G: Subgraphs in Heritage Projects

This shows the number of subgraphs in each project as well as the size of each subgraph in terms of the objects that comprise them. This shows how well the annotated objects in each project are internally connected.

Saumarez Combined

Subgraph Sizes	
Subgraph 1	1392 objects
Subgraph 2	1 object
Subgraph 3	1 object
Subgraph 4	1 object

Armidale in Photos

Subgraph Sizes	
Subgraph 1	41 objects
Subgraph 2	1 object

Nickson AIF Photos

Subgraph Sizes	
Subgraph 1	126 objects
Subgraph 2	2 objects
Subgraph 3	2 objects
Subgraph 4	3 objects
Subgraph 5	1 object
Subgraph 6	2 objects
Subgraph 7	1 object
Subgraph 8	1 object

Nursing VAD Identification

Subgraph Sizes	
Subgraph 1	49 objects
Subgraph 2	1 object

Saumarez Gardens

Subgraph Sizes	
Subgraph 1	446 objects
Subgraph 2	1 objects