SEVENTH FRAMEWORK PROGRAMME

ForgetIT

www.forgetit-project.eu

ForgetIT

Concise Preservation by Combining Managed Forgetting and Contextualized Remembering

Grant Agreement No. 600826

Deliverable D6.3

Work-package	WP6: Contextualization / Decontextualization
Deliverable	D6.3: Contextualisation Tools - Second Release: Updates to the Context Modelling Framework and Modules
Deliverable Leader	Mark A. Greenwood (USFD)
Quality Assessor	Johannes Goslar (dkd)
Dissemination level	Public
Delivery date in Annex I	31st January 2015
Actual delivery date	17th February 2015
Revisions	5
Status	Final
Keywords	context, text, images, ontologies, preservation

Disclaimer

Revision History

Version	Major changes	Authors
1	Initial Document Outline	Mark A. Greenwood
2	Initial Section Contributions	All Authors
3	Finished Section Contributions	All Authors
4	Editing Before Internal review	Mark A. Greenwood
5	Editing After Internal review	Mark A. Greenwood

List of Authors

Partner Acronym	Authors
USFD	Mark A. Greenwood, Johann Petrak, Genevieve Gorrell
L3S	Andrea Ceroni
CERTH	Vasilios Mezaris, Vassilios Solachidis, Olga Papadopoulou
DFKI	Bahaa Eldesouky, Heiko Maus

Table of Contents

Table of Contents 4

Executive Summary 7

1 Introduction 8

1.1 Target Audience 8

1.2 Structure of the Deliverable 8

2 The Big Picture 9

2.1 An Illustrative Example 9

2.2 Technical Discussion 13

2.3 How to Evaluate Contextualization 14

3 Image Contextualization 15

3.1 Problem statement 15

3.2 ForgetIT approach 15

3.2.1 Method overview 15

3.2.2 Subevent similarity 15

3.3 Experimental evaluation and comparison 17

3.4 Software implementation and integration 18

4 Text Contextualization using Seed 21

4.1 Problem statement 21

4.2 ForgetIT approach 21

4.3 Experimental evaluation 22

4.3.1 Demographics 22

4.3.2 Performance Measures Assessment 23

4.4 Software implementation and integration 24

4.5 Discussion 24

5 World Knowledge Based Contextualization of Text 26

5.1 Problem statement . . . 26

5.2 ForgetIT approach . . . 26

5.3 Experimental Evaluation and Comparison . . . 27

5.3.1 Corpora . . . 27

5.3.2 Other Systems Used for Comparison . . . 27

5.3.3 Results . . . 29

5.4 Software implementation and integration . . . 29

5.5 Discussion . . . 29

6 Re-Contextualization of Text Documents 31

6.1 User Interface . . . 31

6.2 Computational model . . . 31

6.2.1 Basic Query Formulation methods . . . 32

6.2.2 Learning to Select Hook-based Queries . . . 34

6.2.3 Context Ranking . . . 36

6.2.4 Experimental Setup . . . 38

6.2.5 Results of Query Formulation . . . 41

6.2.6 Results of Context Ranking . . . 44

7 Context Evolution 48

7.1 Document Collections . . . 49

7.2 Popularity Changes . . . 49

7.3 Semantic Changes . . . 49

7.4 Evolution Assessment . . . 50

8 Conclusions 51

8.1 Summary . . . 51

8.2 Assessment of Performance Indicators . . . 51

8.2.1 Documented Conceptualization of Information Contextualisation . . . 51

8.2.2 Contextualisation Tools and Framework 51
8.2.3 Techniques for Dealing with Information Evolution 52
8.3 Next steps 52

References **53**

A Text Contextualization Evaluation Metrics **57**

Executive summary

The goal of WP6 is to develop methods that enable the contextualization of both text and images. This is intended to support the future understanding and re-use of preserved documents by augmenting them with details of the context surrounding them at the time of creation.

In this deliverable we present the second release of the ForgetIT techniques for contextualization, which build upon those reported in D6.2 [Ceroni et al., 2014a], along with evaluation results for each component. We also present an updated approach to contextualization which focuses on a user centred approach.

A final discussion section highlights how the ForgetIT approach is mirrored by the reported components and sets out the proposed work plan for the next period of the project.

1 Introduction

This deliverable documents the second release of the ForgetIT components for contextualization. These components have been developed following a thorough state-of-the-art review [Ceroni et al., 2013] to carry out the numerous tasks defined within the ForgetIT approach to contextualization. The ForgetIT approach was initially documented via a formalism heavy description [Ceroni et al., 2014a] that, in retrospect, was difficult to follow for those not heavily immersed in the project. This deliverable includes an updated description that takes a more user centric approach with a worked example that highlights a number of the challenges inherent in contextualizing a document.

1.1 Target Audience

As this is a prototype deliverable it is, by it's very nature, quite technical in places. Each section is, however, self-contained and aimed at a specific audience and as such relevant parts can be read by the appropriate audience. Also each section starts with a broad overview which should allow a basic understanding of the component being described even if the technical details are outside the area of expertise of the reader.

1.2 Structure of the Deliverable

The remainder of this deliverable is structured as follows. Firstly we present an updated overview of the ForgetIT approach to contextualization that is based around a worked example as seen from a users point of view. Following on from this high level discussion there are five sections describing components (or work that will lead to a component). These cover image contextualization, user driven textual contextualization, automatic contextualization against world knowledge, the re-contextualization of textual documents, and work towards context evolution. The deliverable ends with a discussion section.

2 The Big Picture

To date within the ForgetIT project we have, in the main, developed approaches to contextualization that focus on linking a document (text or image) to some form of external knowledge [Ceroni et al., 2014a]. This has included, for example, linking text documents to DBpedia[1] instances, expanding text documents with relevant sentences from Wikipedia[2], and expanding image collections by adding other representative images from public collections. While these approaches are useful they focus on using general world knowledge as the context for a document.

While general world knowledge clearly forms a large part of the context of a document, we would argue that it is significantly less important than personal world knowledge. Even in the far distant future today's world knowledge should, barring a large scale *digital dark age* [Kuny, 1998], be accessible but its use is limited as it only extends to well known or *famous* entities and events. Everyday personal details, events and relationships form a rich context for understanding documents and this must be captured as part of any successful approach to contextualization.

In this section we look at how both general and personal world knowledge can be used to contextualize documents (both text and images) and how this additional information can be preserved and utilized. The rest of this document is split into three main parts. Firstly there is a worked example showing the different sources of information available and how these link together to form context. This is followed by a more technical discussion that suggests how such a contextualization approach would work within the confines of the ForgetIT project. We then conclude with a short section which discusses some of the issues surrounding evaluation.

2.1 An Illustrative Example

Rather than focusing on the technical details we shall start by looking at an example which we hope will illustrate the main concepts and pave the way for the development of prototype components and a thorough evaluation plan. This example revolves around a diary entry and a photo covering a single event that occurred during a ForgetIT consortium meeting held in Luleå, Sweden.

If we assume that the diary entry (shown on page 10) is the first document (text or image) that we are preserving then we will have no existing personal store of world knowledge to call upon. The first stage of contextualization will therefore be to link the document to a source of general world knowledge and for this example we will assume this is Wikipedia[3].

[1] `http://dbpedia.org`

[2] `http://www.wikipedia.org/`

[3] Implementations of this idea are more likely to use DBpedia but for a *hand worked* example Wikipedia makes more sense. See the technical discussion on page 13 for more details.

Jörgen had offered to take Elaine, Maria, and Robert out to Gammelstad this morning before the meeting started as they are looking at running a memory study there over the summer. When Jörgen arrived to pick them up I decided to be cheeky and ask if there was room in the car for one more. There was so I got to do the touristy thing of looking around while everyone else did some actual work. A large snow pile made an excellent back drop for Robert to take a group photo which hopefully I'll get a copy of at some point. It was certainly an interesting place to look around and you can understand why it is a UNESCO world heritage site.

Diary Entry 2.1: File metadata lists the author as 'Mark A. Greenwood' and the creation data as the 12th of February 2014

For this example diary entry links would be generated to the pages for Gammelstad[4] and UNESCO[5] as they are the two *famous* entities within the text. This leaves five entity mentions which, if not well known, must fall within the users personal world knowledge: Jörgen, Elaine, Maria, Robert, and I.

With no pre-existing personal world knowledge the only information we have is the file metadata which allows us to map the first person pronoun *I* to the documents author *Mark A. Greenwood*. This leaves us with four unknown people for which the system would need to prompt the user for additional data. At a minimum this additional data should probably consist of a persons full name and their relationship to the user (i.e. Jörgen Nilsson is a collaborator on the ForgetIT project and works at the Luleå University of Technology). Having gathered this information this first step would then be to store the current personal world knowledge in the archive. The diary entry could then be contextualized by storing not only the entry itself but links to both the general and personal world knowledge within a submission information package (SIP); the links to the personal world knowledge being with reference to the archived information package (AIP) containing the most recent archived version.

Not only has this process allowed us to store extra context information alongside the diary entry, but it has started the process of building up a repository of personal world knowledge which can in turn be used to contextualize new documents more accurately and with less user intervention. It would be beneficial if this linking process was part of an interactive feedback loop [Goetz, 2011] so that users could see the benefit of existing data being used to enhance their output (i.e links to Gammelstad and UNESCO appearing *magically*) as this would encourage them to provide data personal data when prompted. Having fully contextualized the diary entry, let us turn our attention to the task of contextualizing and preserving Photo 2.1.

The first thing to note is that the photo and the diary entry clearly act as context for one another. If both were archived at the same time as part of the same submission information package (SIP) this context would be clear but as we are assuming that the photo is being preserved at a later date than the diary entry then this link needs to be

[4] http://en.wikipedia.org/wiki/Gammelstad_Church_Town
[5] http://en.wikipedia.org/wiki/UNESCO

Photo 2.1: Metadata associated with this photo show that it was taken at 7:48:33 UTC on the 12th of February 2014 at N$65°38'42.75''$ **E**$22°1'38.573''$ **at a magnetic bearing of** $171.5°$**.**

made clear. This highlights the fact that one important source of contextual information are the documents you have already archived. Previously archived documents can act as context in two ways:

- A SIP can explicitly reference archived information packages (AIP) as context. In this case the SIP containing Photo 2.1 would explicitly reference the AIP containing Diary Entry 2.1 to provide contextual information that explains both the occasion of the photo and its content (i.e. the people and location).
- The processing of each new SIP adds to what we now about the individual users (where user could be a company not just a person) personal world knowledge. In this example contextualizing the diary entry will have led to four previously unknown people being added to the users personal world knowledge.

It is likely that the explicitly linking of the diary and photo would be a manual action taken by the user. It may, however, be possible to suggest the relationship to the user based on the associated metadata and context information generated when the diary entry was

Photo 2.2: A *wider* context?

preserved. Firstly the photo and the diary were both created, according to the metadata, on the same day, the 12th of February 2014, which at least suggests a common context. Furthermore the GPS information can be used to search for Wikipedia pages describing nearby places which would link the photo linked to the same page describing Gammelstad[6] as used to provide general world knowledge for the diary entry.

The GPS metadata associated with the photo could also be used to select other photos which could act as context. In this example, Photo 2.2 has almost identical GPS metadata but clearly shows a wider view than the original photo and would help to provide a larger visual context. Furthermore, GPS metadata can be used (e.g. employing Google places API) in order to list the places that are located close to the image GPS coordinates. Also, combined with magnetic bearing (if available) and the Focal length of the shot, the subset of the places that may be visible places can be extracted. An example of place info for Photo 1 (based on the GPS info of the caption) is given below:

Visible	Name	Type(s)	Distance
No	Luleå V	sublocality level 1 sublocality political	13626m
No	Äldreboende Ingridshem	establishment	290m
No	Snickare Anders Viklund i Luleå AB	establishment	225m
No	Öhemsvgen	bus station transit station establishment	225m

As a result, nearby places can be used (or proposed to the user) in order to link a photo with a diary in the case that a name of a public place (restaurant, bus stop, hotel) is included in the diary text.

In a similar way to textual documents, images can be contextualized based on their content as well as their associated metadata. In this example, face detection would highlight the four people in the photo and could be used to prompt the user to identify the people; if the link to the diary entry had already been formed then the names of the people

[6] https://en.wikipedia.org/w/api.php?action=query&list=geosearch&gsradius=500&gscoord=65.6452|22.0274&format=xml

associated with it could even be suggested. Also, face clustering can detect similar (already named) faces in existing archived images and suggest them to the user. As all the personal world knowledge needed for contextualizing the photo was already gathered and stored for the diary entry the AIP containing the previously preserved personal world knowledge can be referenced to avoid duplication of information within the preservation system.

2.2 Technical Discussion

While the example discussed in the previous section helps to highlight many of the issues around contextualization, it should be remembered that the use cases within the ForgetIT project cover a much wider and varied range of tasks involving both personal and organizational preservation [Maus et al., 2013a, Maus and Schwarz, 2014, Damhuis et al., 2014]. Fortunately, regardless of the use case scenario the issues discussed in the previous example actually cover most situations we expect to encounter. Contextualization essentially boils down to the following sequence of actions:

- Process the document to extract *contextual hooks*. For all document types metadata will act as a contextual hook, providing temporal context, authorship information, location information etc. For textual documents any sequence of characters could act as a hook, although usually these will be limited to sequences annotated using techniques such as named entity recognition or term extraction, while images and video may be subject to face detection and clustering, near duplicate detection, object recognition or scene classification.

- An attempt will then be made to link each hook to the users existing personal world knowledge as in general we assume that most references will not be to *famous* entities. For example, people mentioned in documents are more likely to be friends, relatives, or colleagues than they are to be movie stars.
  - if a link is found then it is added to the context for this document processing moves to the next hook.

- An attempt will next be made to link the hook to general world knowledge to help place the document into a wider context.
  - if a link is found then it is added to the context for this document processing moves to the next hook.

- If the hook has not been linked to either the personal or general sources of world knowledge then we assume that it is personal knowledge that we have not encountered before.
  - the user is prompted to provide information about the hook
  - the new information is added to the personal world knowledge and a copy is preserved

- the link between the hook and the new information is added to the context for this document

While this approach to contextualization (focusing on personal context over general world knowledge) may differ from that described previously in the project [Ceroni et al., 2014a] the techniques developed so far within the ForgetIT project all still have a role to play. In fact most of the steps outlined above can be completed using tools and techniques which have already been developed within the project. For example, there are semantic writing component under development within the two use cases that act as the bridge between the technology and the users and where prompts for more information could be inserted. Techniques for the extraction of contextual hooks have been reported for both text and images, and the ability to store contextual information within a SIP or to reference existing AIPs has been discussed and shown to be possible.

2.3 How to Evaluate Contextualization

As we saw in the previous section, quite a few of the individual technical components have already been developed, although there is clearly still work to be done to both improve the components and to integrate them into a cohesive and easy to use system. The main challenge, however, will be evaluation. While the individual components can be evaluated in isolation (i.e. is this entity link correct, is that really a face in a photo...) it is as yet unclear how the end to end system of contextualization can be evaluated.

The main issue with performing an end-to-end evaluation of contextualization is that the context required to understand a document is specific to the person consuming the information. For example, it is unlikely that when looking at a photo of a close relative, baring a degenerative neurological condition, you would need to be reminded of who is in the photo. If, however, in fifty or a hundred years time one of your descendants was viewing the photo it is more likely that a reminder of who the photo depicts would be useful; this is after all why so many old family photos have writing on the back. Similar issues arise when two people consume the same document contemporaneously as their different knowledge and expertise will provide an initial context in which the document will be interpretted; an economist and a geologist are likely to require different context information when reading an article from the Financial Times.

The simplest solution would be to present all users with all known context, but this is likely to lead to user dissatisfaction as they will be constantly overwhelmed with unnecessary information. A more appropriate solution would be to limit the returned context based upon knowledge of the user and the task they are currently undertaking. Such an evaluation will require careful thought and collaboration with WP9 and W10 (as well as WP2) to devise a sensible scenario in which to determine the relevant context to display. One possible approach would be to adopt an evaluation method similar to that proposed by [Bizzozero et al., 2004] which investigated memories associated with media-mediated events, i.e. with well known events rather than personal memories.

3 Image Contextualization

3.1 Problem statement

Image contextualization enriches a seed image collection with images of an archive collection taking into account semantic content and metadata information. It can be applied both in public (world) and personal events. For example, in public events (concerts, sports), the user can import his seed collection and retrieve other user images related with the event, such as images from different viewpoints. The same procedure can be also applied in personal events such as a ceremony, a personal or a business trip. In this case, the user's image collection can be enriched from the friends' or colleagues' collections, similarly to what is done in the public event case.

3.2 ForgetIT approach

3.2.1 Method overview

Our image contextualization approach is illustrated in Figure 3.1. The proposed method initially looks for subevents in the seed collection using visual, time and GPS information. This is particularly important in the case of long events or multi-location events (e.g. trips, Olympic games). Subevent detection is performed using multimedia analysis methods developed in WP4 (Multi-user synchronization, Concept detection, Near duplicate detection), which are presented in detail in D4.3 [D4.3, 2015]. Then, similar and different sub-events of similar events are detected within the archive collection, and a selected subset of their images is used for enriching and contextualizing the seed collection.

It should be noted that the archive collection is already analysed and organized in subevents since subevent clustering and similarity assessment is performed iteratively as each new set of images is added in the archive collection. Thus, as depicted in Figure 3.1, subevent extraction is executed only for the seed collection.

Finally, we assume that all the collections of the archive belong to the same event, or each collection is tagged with an event label by the users when it is added into the archive (e.g. using the PIMO interface).

3.2.2 Subevent similarity

Let S be the seed collection which consists of I_N images which are clustered in S_{SE_N} subevents $SE_i^S, i = 1, ...S_{SE_N}$. Similarly, $SE_i^A, i = 1, ...S_{SE_A}$ be the sub-events of the archive collection which have created after applying the clustering algorithm of WP4 [D4.3, 2015].

The method gets two input parameters a and b. Parameter a, $a > 0$, controls the number of

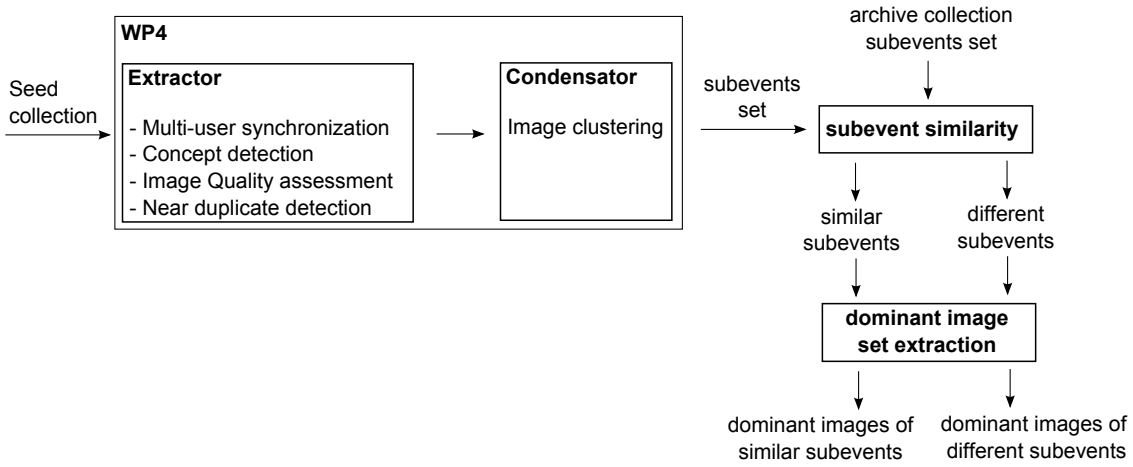

Figure 3.1: Method overview

the archive collection that will be used for contextualizing the seed collection; our method enriches the seed collection with $a \cdot I_N$ images from the archive one. Parameter b, for which $0 \leq b \leq 1$ specifies what percentage of the latter images should belong to sub-events that are originally represented in the seed collection; the rest of the archive images that will be used for contextualizing the seed collection will be chosen so as to belong to the same event but different sub-events. Both these parameters are user-controlled.

Initially, the distance between the seed and archive sub-events is calculated. Sub-event distance is defined as the minimum distance between the sub-events images:

$$D(SE_i^S, SE_j^A) = \min_{k,l}\{\ d(I_k^S, I_l^A)\},$$

where $I_k^S \in SE_i^S$ and $I_k^A \in SE_i^A$. Image distances are given by [Mezaris et al., 2010]

$$d(I_i, I_j) = \sqrt{\sum_{k=1}^{J} \frac{(C(I_i) - C(I_j))^2}{C(I_i) + C(I_j)}}$$

where $C(I)$ is the image model vectors [D4.3, 2015] of image I. Finally, we calculate the distance between the seed collection and the archive sub-events, which is given by

$$D(S, SE_j^A) = \min_{i}\{\ D(SE_i^S, SE_j^A)\}.$$

We select the $a \cdot b \cdot S_{SE_N}$ archive sub-events that are most similar to the seed collection subevents based on $D(S, SE_j^A)$, and for each archive sub-event we pick one image which is the most dissimilar to the images already contained in this subevent of the seed collection. The set of the $a \cdot b \cdot S_{SE_N}$ images is the set of images that contextualizes the subevents of the event that were originally represented in the seed collection.

The remaining $a \cdot (1-b) \cdot S_{SE_N}$ images that will enrich the seed collection will be collected from the most dissimilar sub-events for the same event that can be found within the seed collection chosen (using the same distance measure as above). From each sub-event among the most dissimilar ones, the most representative image will be retrieved; this image is the one whose model vector is closest to the mean of the model vectors of the images belonging to the sub-event.

3.3 Experimental evaluation and comparison

We have tested our method on one of the datasets used in the MediaEval SEM task [Conci et al., 2014]: the London dataset, consisting of 2142 photos capturing various sub-events of the London 2012 Olympic Games.

As explained in the previous section, archive images belonging to two distinct sets (i.e. images belonging to subevents originally represented in the seed collection, and images not belonging to these subevents) are used for contextualizing the seed collection. We numerically evaluate the impact of using these images for contextualization by looking at the sub-events that they represent. Specifically, we calculate three evaluation measures:

- Percentage of similar: Out of the images of the archived collection that were selected for contextualization on the basis of representing subevents that were already included in the seed collection, we measure the percentage of them that truly belong to such subevents. The volume of this measure ranges from zero to one, one being the optimal.

- Percentage of dissimilar: Out of the images of the archived collection that were selected for contextualization on the basis of representing subevents that were not included in the seed collection, we measure the percentage of them that truly belong to such subevents. The volume of this measure ranges from zero to one, one being the optimal.

- Cluster recall: we measure the coverage increase after contextualization. Namely, the initial coverage of the seed collection is calculated based on the number of different sub-events of the total number of sub-events contained in the overall event. By contextualizing the collection we attempt to include more sub-events into the seed collection and increase the coverage.

The London dataset consists of 37 user collections. We consider as seed collection the $user1$ collection; the remaining 36 form the archive collection (thus are treated here as having already been processed and archived). As aforementioned, we assume that we know that the seed collection and the 37 archived collections are from the same event, London Olympics.

Figure 3.2 illustrates the values of these measures when varying the values of parameters a and b, while indicative results for selected values of a and b are shown in Table

3.1. As far as the cluster recall measure is concerned, the $user1$ seed collection consists of 46 out of the 238 total sub-events forming the London Olympics event before contextualization, thus being equal to 0.1932. As parameter a increases, which means that we increase the number of images that we use for contextualizing our seed collection, cluster recall also increases and reaches almost 0.3. This indicates that the contextualized seed collection offers a broader coverage of the event, in comparison to what it did before contextualization.

Table 3.1: Percentage of the three evaluation measures for different a and b parameters

	a=0.5			a=1		
	b=0.2	b=0.5	b=0.8	b=0.2	b=0.5	b=0.8
Percentage of similar	0.8	0.7619	0.5758	0.8333	0.5833	0.4615
Percentage of dissimilar	0.8261	0.8421	0.9	0.7297	0.8148	0.8889
Cluster recall	0.2269	0.2605	0.2731	0.2563	0.2815	0.2941

A contextualization example is presented in Figure 3.3 allowing for visual inspection of the results. This example shows how the seed collection is contextualized with other images (from similar and dissimilar sub-events with the user's ones) from the same event (2012 London Olympics). In Figure 3.3a the seed collection images are illustrated, grouped in sub-events using the clustering method presented in D4.3 [D4.3, 2015]. It seems that this collection contains images from a part of the opening and award ceremonies, and the rowing (coxless pair, eight, single scull, quad scull), weightlifting, soccer, track (including. marathon, long jump, races), wrestling, tennis, beach volley and judo competitions. Figure 3.3b shows the images of similar subevents that were chosen from the archive for contextualization. Finally, Figure 3.3c shows images from other sub-events that are also chosen for contextualization. These include images from the taekwondo, cycling, badminton, fencing, sailing and horse riding competitions, as well as different parts of the opening and award ceremonies.

3.4 Software implementation and integration

The method above is implemented in MATLAB and gets as input the output of the Extractor service.

The method returns two sets of images to be used for contextualizing the seed collection (images from similar and dissimilar subevents respectively). The prototype component described in this section can be downloaded from `http://www.forgetit-project.eu/en/downloads/workpackage-6/`. Please contact the project for access to this protected section of the website.

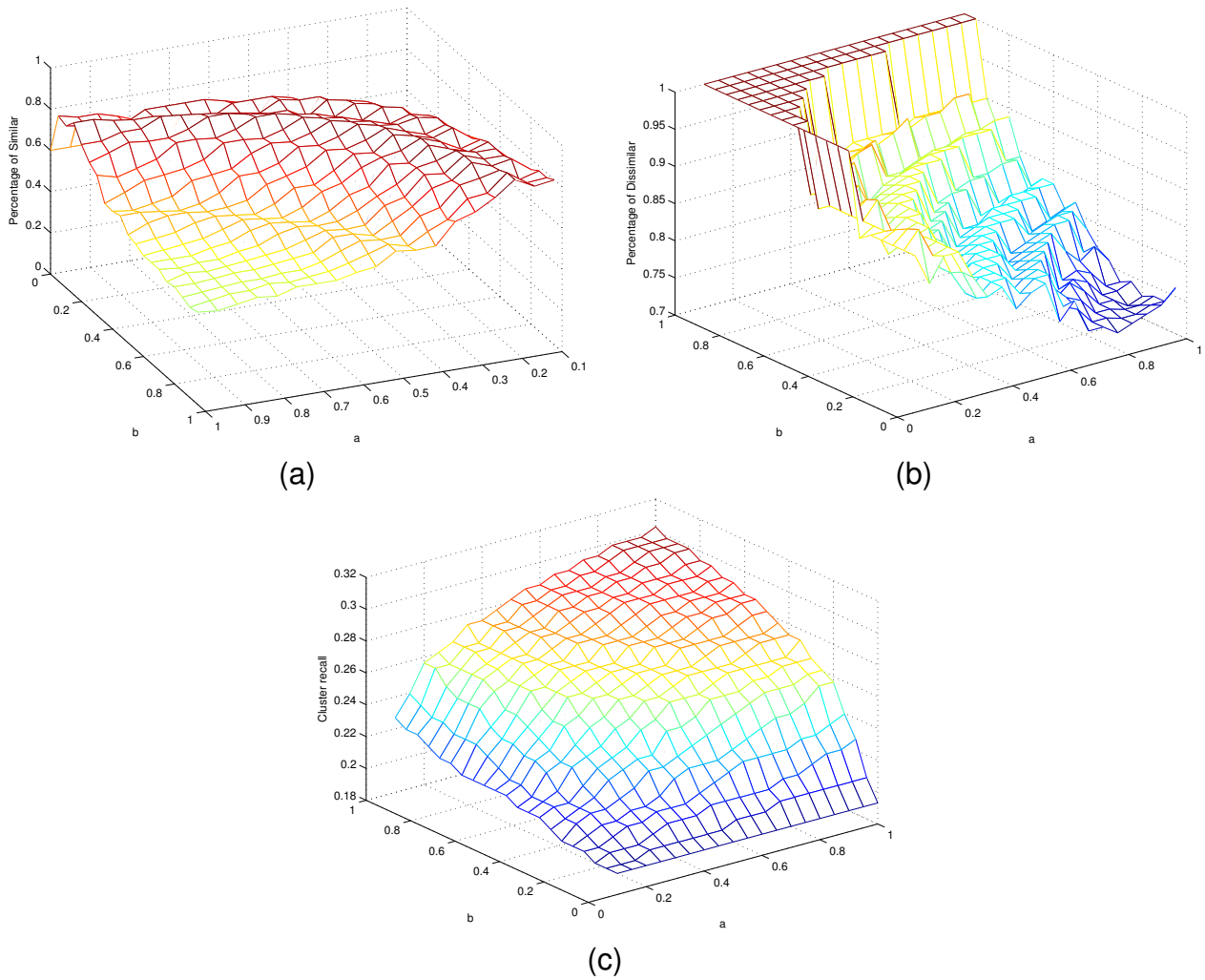

Figure 3.2: Evaluation results using the three defined evaluation measures, for different a and b parameters (a) Percentage of Similar, (b) Percentage of Dissimilar and (c) Cluster recall

Functional description	Image collection contextualization
Input	Seed and archive collections
Output	A directory containing the selected images
Contextualization Level	Personal and World
Limitations/Scalability	None
Language/technologies	MATLAB
Hardware Requirements	N/A
OS Requirements	Windows
Other Requirements	Tools developed in WP4

Table 3.2: Image contextualization software summary

(a)

(b)

(c)

Figure 3.3: (a) Seed collection sub-events, (b) Images added through contextualization that belong to subevents already represented in the seed collection, (c) Images added through contextualization that belong to other subevents of the same event

4 Text Contextualization using Seed

4.1 Problem statement

Unstructured data such as free text, emails, notes, meeting minutes ... etc. are pervasive in personal computing. This pervasive nature brings the challenge of managing and properly benefiting from its content. This challenge is even more evident in the context of personal information management on the Semantic Desktop [Maus et al., 2013b]. As shown in Section 2 of D9.3, users of the Semantic Desktop perform tasks that frequently involve writing such as taking notes, writing descriptive texts for PIMO concepts, creating tasks, writing meeting minutes, ... etc. As a result, a considerable percentage of the data stored in a user's PIMO consists of unstructured natural language text.

In order to better support contextualized remembering and consequently concise preservation, we tackle challenge of contextualizing text in the Semantic Desktop. We focus in the first place on contextualisation at the personal level, but we also support contextualisation on the world level.

4.2 ForgetIT approach

The interface for dealing with textual content in the Semantic Desktop is Seed, an extensible knowledge-supported portable natural language text composition tool. We embed it in multiple GUIs of the Semantic Desktop, which require interaction with text (e.g. note taking, task descriptions ... etc). Our approach to textual content contextualisation in ForgetIT by using Seed has the following main characteristics:

- **Immediate contextualisation** Users can annotate their texts with PIMO entities as early as during the composition process.

- **Proactivity** Seed automatically as well as semi-automatically annotates entity mentions during text composition, thus reducing the effort required by the user for adding contextual information to the text being composed.

- **Reliability** In contrast to batch text annotation, using Seed for contextualizing text while interacting with it, provides a chance to review, modify or completely reject annotations suggested by Seed. This in turn results in more reliability of the annotation process.

- **Personalisation** This is emphasized by giving priority to personal knowledge over world knowledge. Users of Seed can annotate private concepts

- **User-friendliness** Seed has to be user-friendly in order for users to be encouraged to use it.

Our goal is to capture knowledge in the text and use it to add contextual information to it. Seed does so by analyzing the text being authored and annotating mentioned entities from user's PIMO or from Linked Open Data (LOD) sources such as DBPedia [Auer et al., 2007] and Freebase [Bollacker et al., 2008].

4.3 Experimental evaluation

In D4.2 [Papadopoulou et al., 2014] and D4.3 [D4.3, 2015], we showed how we perform and improve upon the named entity recognition done in Seed. In D6.3 we wanted to evaluate the contribution to text contextualization Seed represents. So, we carried out an online evaluation experiment on a sizable group of test subjects [Eldesouky et al., 2015] with the following setup:

- Participants watched a video approximately 3 minutes long about Seed, which explained in an non-technical way its functionalities.
- Using *Seed*, participants were required to read and annotate multiple text passages with entities from Linked Open Data (LOD) sources. All passages were annotated by 3 different human annotators to produce a ground truth sample. Annotations agreed upon by 2 or more were included in the final sample. At the end, participants' annotations were compared with the ground truth sample for performance measures assessment.
- Every participant started with a given text. The user then reviewed automatic annotations by Seed as well as annotation suggestions that (s)he can confirm, modify or reject. Figure 4.1 shows a sample screenshot of the user interface with which test subjects interacted.

The evaluation assessed Seed's value for world knowledge contextualization. The assessment of its value for personal knowledge contextualization on a large scale was not possible in the same evaluation, because of the lack of a large population of users that have used PIMO for a period long enough to have usable personal knowledge models.

4.3.1 Demographics

Number of participants
The evaluation was performed by 115 participants.

Age

- (15-25) years: 47.3%
- (25-35) years: 41.9%
- (35-45) years: 8.6%

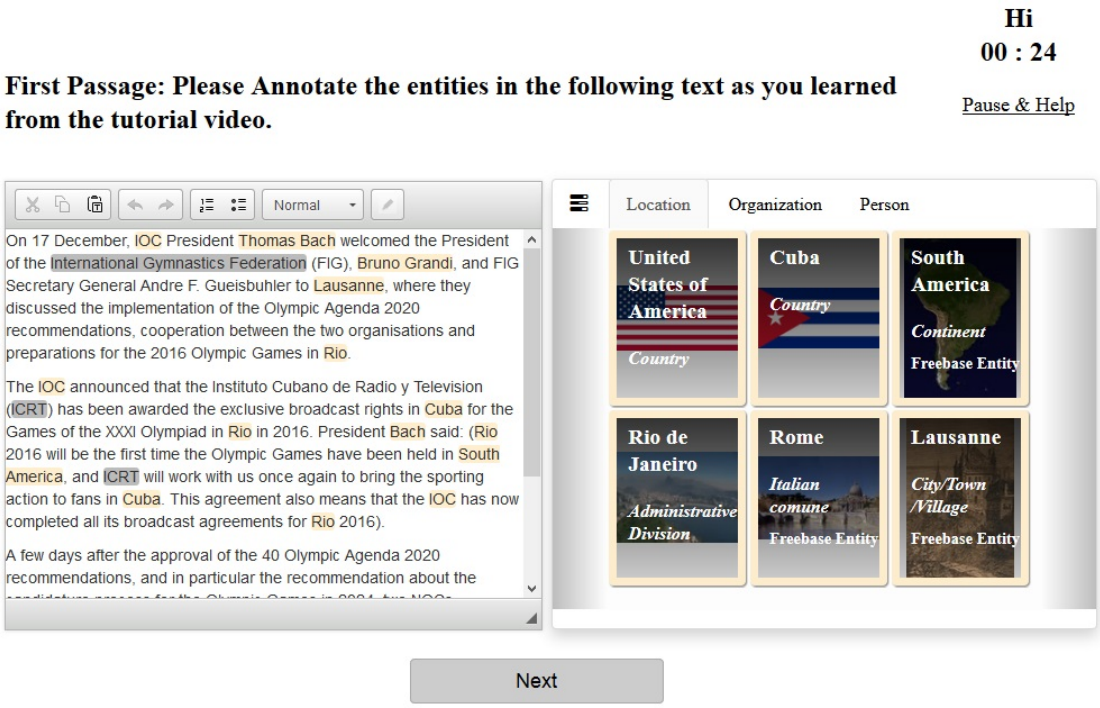

Figure 4.1: Screenshot of the evaluation interface. Orange highlights signal automatically disambiguated entities while grey ones require user intervention to disambiguate them. On the right side is an interactive faceted view of entities with extra information from world knowledge

- (55-65) years: 2.2%

Background

- Undergraduate students: 30.3%
- Graduate students: 23.2%
- Computer professionals: 19.2%
- Non computer professionals: 19.2%
- Researches: 8.1%

4.3.2 Performance Measures Assessment

In order to evaluate the performance of *Seed*'s annotations, we calculated the Precision, Recall and F-1 score for entity annotations for all text passages submitted by participants. Section A includes an explanation of all three performance measures.

Table 4.1 shows the average values for all three passages.

Table 4.1: Annotation performance measures assessment

Avg. Recall	Avg. Precision	Avg. F1
0.74	0.88	0.80

The values show that Seed has helped users capture most of the entity mentions in the text without prerequisite knowledge. The explicit annotations of entities provide strong clues for further contextualization.

In our attempt to subjectively interpret the outcome of the evaluation and figure out how it can be improved, we examined accompanying user feedback.

- Many of the test subjects reported that a domain-specific text composition task would make them annotate the entities better. We plan to take that into consideration in upcoming evaluations. We expect it to have a positive effect on the performance measures.
- The majority of the users said they imagine they would use Seed frequently. They also stated they found it easy to use.
- The majority of the users mentioned the annotations helped them understand the text passages better.

4.4 Software implementation and integration

Functional description	A text contextualisation tool
Input	Rich text
Output	Annotated rich text with information about its content
Contextualization Level	Both
Language/technologies	HTML, JavaScript, Java
Hardware Requirements	N/A
OS Requirements	Server: Any OS with Java support Client: Any OS with an internet browser
Other Requirements	

Table 4.2: Implementation and integration requirements

4.5 Discussion

In the previous sections we showed how our approach to text contextualization proved successful in capturing knowledge about unstructured text and used it to annotate the text with important entities mentioned. The use of Seed helped contextualize text in a

reliable way by allowing authors to immediately interact with annotations as they type. This eliminates the need for or at least facilitates later batch contextualization of the text. The evaluation involving a sizable group of test subjects presented results which support our approach.

5 World Knowledge Based Contextualization of Text

5.1 Problem statement

As discussed in Section 2 an important part of contextualization, of both text and images, is the process of linking a 'document' to real world entities; people, organizations, locations, events, etc. Such contextualization will benefit the project scenarios in multiple ways. Firstly the contextualized information can be used immediately (i.e. prior to preservation) to enhance searching within active data. More importantly such contextualization data allows archived documents to be reintegrated at any future date by providing the information required to link to the current users knowledge (both world and personal). Context can also be used as a way of navigating preserved information as it can act as a memory trigger which will in turn guide a user through their collection of preserved documents.

5.2 ForgetIT approach

The approach detailed in this section (which is an expanded form of that described in D6.2 [Ceroni et al., 2014a]) uses DBpedia as the source of world knowledge against which to contextualize text documents. DBpedia is a structured knowledge base in which the data has been extracted (and hence links to) Wikipedia. This makes it an ideal resource for automatic processing while at the same time providing additional human readable resources. We refer to the system as YODIE (Yet another Open Domain Information Extraction system).

Contextualization of a document with reference to DBpedia is essentially a three step process involving

1. linguistic pre-processing
2. candidate generation
3. disambiguation

The pre-processing components are by their very nature language specific, since they carry out the necessary low-level linguistic analysis. They are the tools used for recognising:

- word and sentence boundaries,
- part-of-speech categories for individual words,
- named entities, and
- English transliterations for entities written in languages other than English.

Given an entity or a span of text, the candidate generation components are used for obtaining a list of candidate URIs (from DBpedia). We use DBpedia resources to prepare a gazetteer with labels, names and aliases (including acronyms) of various entities. Execution of such a gazetteer creates Lookups highlighting candidate entities that should be disambiguated in the text.

Once the pre-processing is achieved and list of candidates are produced, a set of disambiguation components are executed. In particular, the following steps are taken:

1. Candidates produced in the first step are filtered out if they do not have at least one proper noun under the annotation span. This is to make sure that the spans being disambiguated are referring to entities only.
2. DBpedia redirects are applied to the candidate URIs and any candidate URIs referring to pages with disambiguation URIs are excluded [Ji and Grishman, 2011, Rao et al., 2013].
3. Each candidate URI is then assigned a score by four different similarity measures (these are described in detail in [Aswani et al., 2013]): string similarity, semantic (structural) similarity, contextual similarity, and commonness.
4. Finally, a machine learning based approach is used to decided among the candidates using the similarity scores. Specifically we use Mallet's maximum entropy approach trained using data from the 2010, 2011, and 2012 TAC KBP tasks[7] and the AIDA training set [Hoffart et al., 2011].

5.3 Experimental Evaluation and Comparison

5.3.1 Corpora

To evaluate our approach to contextualization we have used testing set B from the AIDA collection. This consists of 4485 named entities spread over 230 documents. As far as we are aware, this corpus is human-annotated, both at the stage of identifying the entities and linking the correct referents. It has some idiosyncrasies; demonyms are included, for example, where other corpora have considered only proper nouns. However, it is a valuable corpus of well-formed news text.

5.3.2 Other Systems Used for Comparison

Reporting evaluation numbers (using the metrics in Appendix A) of a single system in isolation, while useful, never tells the whole story. In an attempt to show that our approach performs well we have also evaluated six other systems which identify entities in text and link these to DBpedia URIs.

[7] http://www.nist.gov/tac/2013/KBP/

System	Prec	Rec	F1	Acc
YODIE	0.62	0.65	0.64	0.65
AIDA	**0.70**	**0.74**	**0.72**	**0.74**
Lupedia	0.58	0.31	0.40	0.31
DBpedia Spotlight	0.22	0.49	0.31	0.49
TagMe	0.18	0.45	0.26	0.45
TextRazor	0.35	0.58	0.43	0.58
Zemanta	0.51	0.29	0.37	0.29

Table 5.1: Comparison of performance on the AIDA B news corpus

- **AIDA** [Hoffart et al., 2011] is an open source entity linking system developed at the *Max Planck Institut für Informatik*. All evaluations were done using a locally installed version using the dataset prepared August 1st 2014[8].

- **Lupedia**, a free service from Ontotext[9] developed as part of the NoTube project, provides automatic entity look-up and disambiguation in text documents either from within a browser or via an API.

- **DBpedia Spotlight** [Daiber et al., 2013] is available[10] both as a free-to-use web service and as open source software which can be installed and deployed on one's own servers. We used the free web services in our evaluation.

- **TagME** [Ferragina and Scaiella, 2012] is a system developed at the University of Pisa and can be used via a web demo[11] as well as a RESTful API[12].

- **TextRazor**[13] is a UK startup founded in 2011. It offers a web API for semantic annotation, which can be used for up to 500 requests a day for free.

- **Zemanta**[14] is a Slovenian startup founded in 2007, providing a platform for automatic enrichment of blogs and other online content.

Many of theses systems are highly configurable, however in this evaluation we used the default parameters in all cases in order to avoid tuning any of the systems to the evaluation corpus.

5.3.3 Results

Our approach, referred to as YODIE as noted above, achieves a convincingly superior performance on this test set (see Table 5.1), with wide margins separating performance from the nearest competitor. The only exception is the AIDA system itself, which unsurprisingly shows a particularly good performance on its own corpus.

5.4 Software implementation and integration

Our approach to contextualization of real world knowledge, as detailed in this section, has been made available as a RESTful web service, as well as an interactive demo, using GATE WASP (see [D4.3, 2015]) to allow for easy experimentation and integration within the use case tools. The service can be found at `http://services.gate.ac.uk/yodie/`

Functional description	world knowledge contextualization
Input	text
Output	DBpedia URIs
Contextualization Level	World
Limitations/Scalability	none
Language/technologies	Java, GATE, DBpedia
Hardware Requirements	sufficient RAM for large indexes
OS Requirements	64bit Java support
Other Requirements	none

Table 5.2: Functional Description

5.5 Discussion

The results presented above show that our approach to contextualizing a document against world knowledge (by disambiguating entities against DBpedia) performs well. The extra context added to a document by this approach is useful not only for future re-integration of content but it can also be used immediately as a source of knowledge for search and navigation within a corpus. This aspect of the approach will be investigated further and reported in the next prototype deliverable (D6.4). Further work will also

[8] `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/`
[9] `http://lupedia.ontotext.com/`
[10] `http://dbpedia.org/spotlight`
[11] `http://tagme.di.unipi.it/`
[12] `http://tagme.di.unipi.it/tagme_help.html`
[13] `https://www.textrazor.com/`
[14] `http://www.zemanta.com/`

involve combining this approach with those documented in Section 4 to bridge the gap between contextualization against world knowledge and the personal knowledge of the system user

6 Re-Contextualization of Text Documents

This component, whose proof-of-concepts version was introduced in Deliverable D6.2 [Ceroni et al., 2014a], tackles the problem of providing context for textual document, particularly for old ones, for which the original context is missing. A user interface has been developed to showcase the behaviour of the component. From a research perspective, we improved the *proof of concepts* version described in D6.2, which resulted in a publication to SIGIR'14 [Ceroni et al., 2014b], under two aspects: (i) we automatize the process of query formulation from documents to be contextualized, and (ii) we provide advanced approaches for retrieval of contextualization candidates and ranking them taking into consideration complementarity. This work has been accepted as full paper at WSDM'15 [Tran et al., 2015]. In the rest of this section we will describe both the developed user interface and the improved version of the re-contextualization component.

6.1 User Interface

A user interface, available online at [ReC,], has been developed to showcase the component. The back-end core of the application is currently the work presented in [Ceroni et al., 2014b], and we plan to replace it with the one presented in [Tran et al., 2015] in the next future.

Once the user has picked one news article from the available ones, he/she can read its lead paragraph, annotate the terms that requires contextualizing information (i.e. hooks), and retrieve a ranked list of context for the article. An example summarizing this interaction is given in Figure 6.1. In the right-hand part of the interface, the lead paragraph of the chosen news article is visualized, along with the title and the publication date, and the user can specify what requires contextualization by typing in the dedicated fields. Then the user can click on "Submit" to start the re-contextualization process. First, the annotated hooks are queried together as a unique global query to retrieve context candidates. More query formulation options will be available in the next version of the application, according to [Tran et al., 2015]. Second, the the context candidates are re-ranked according to the model presented in [Ceroni et al., 2014b] and visualized to the user on the left-hand part of the interface. The text of the top-ranked results (consisting in three adjacent sentences of Wikipedia) are visualized along with the link of the original Wikipedia page the text belongs to.

6.2 Computational model

Before describing the details of our approach, we summarize the main concepts that we used and we give an overview of the entire process. In the general contextualization model underlying our approach we distinguish the information items d to be contextualized and the context source, where the information for the contextualization comes from. Within d a contextualization hook h is an aspect or part of d that requires further infor-

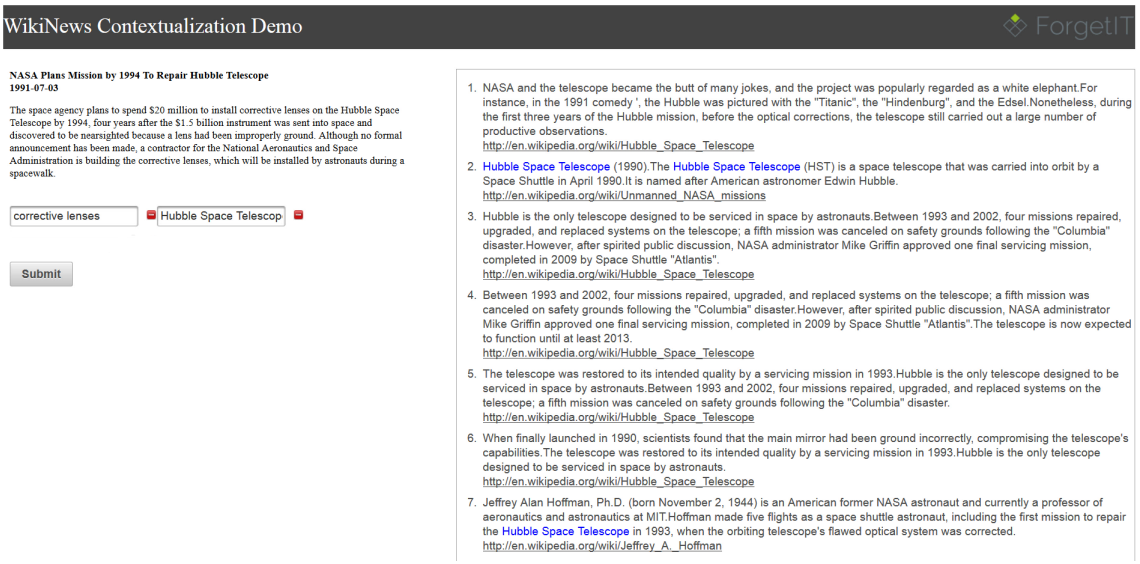

Figure 6.1: GUI of the component.

mation for its time-aware interpretation. The context source is organized into contextualization units cu. In our approach, we have pre-processed a Wikipedia dump as the context source resulting in annotated and indexed Wikipedia paragraphs as contextualization units (see figure 6.2). As information items d to be contextualized we use articles from the New York Times Archive [15] with manually annotated contextualization hooks, i.e. we assume that a reader has marked the places she finds difficult to understand.

The contextualization process, sketched in Figure 6.2, consists of two main steps: (1) formulating queries that are able to retrieve contextualization units, which are good candidates for contextualization; (2) retrieving and ranking the candidates from the context using the queries from step (1). For step (1) we explore document-based and hook-based query formulation methods and present a procedure that selects good queries based on recall-oriented query performance prediction. For step (2), we employ a retrieval method based on language modelling and re-rank the retrieved contextualization candidates based on a variety of features and a learning to rank approach for ensuring complementarity.

6.2.1 Basic Query Formulation methods

The goal of the query formulation phase consists in generating a set of queries Q_d for a given document d to retrieve contextualization candidates as input for the re-ranking phase. We explore two families of query formulation methods, one using the document to be contextualized itself as a ”generator” of queries, and the other using contextualization hooks as generators. Since some of these methods can generate more than one query from an input document, we will discuss two procedures to merge the ranked result lists

[15] http://catalog.ldc.upenn.edu/LDC2008T19

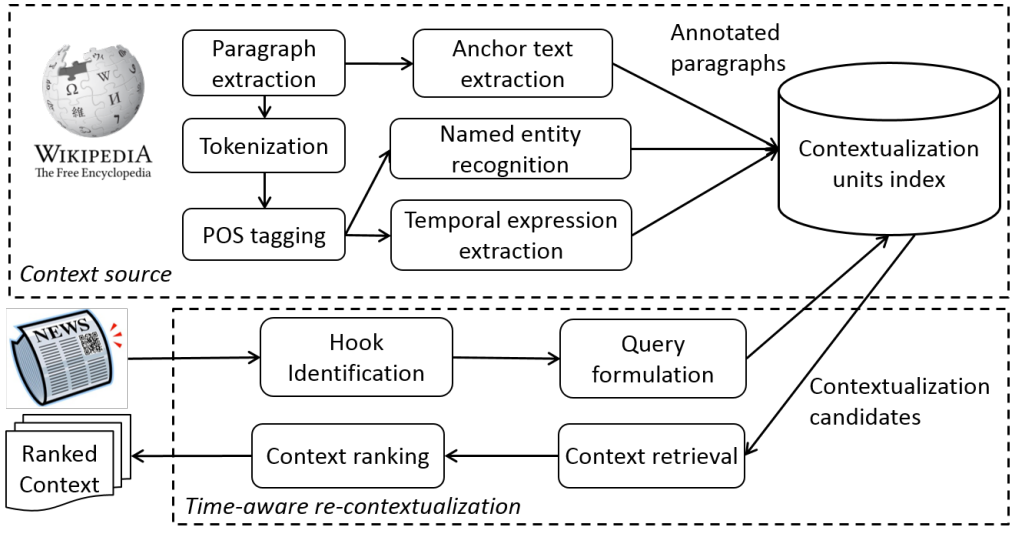

Figure 6.2: Overview of the Re-contextualization approach.

in Section 6.2.3.

Document-based Query Formulation. The first family of query formulation methods exploits the document content and structure. Similarly to [Tsagkias et al., 2011], we use three methods to formulate queries from documents: *title*, *lead*, and *title+lead*. *Title* formulates a query consisting in the document title, which is indicative of the main topic of the article. *Lead* uses the lead paragraph of a document, representing a concise summary of the article and including its main actors. *Title+lead*, as a combination of the previous two methods, formulates a query consisting in both the title and the lead paragraph of the document. Before being performed, all the queries are pre–processed by tokenization, stop-word removal, and stemming. We did not investigate further information extraction approaches for query formulation, since it has been already proven in [Tsagkias et al., 2011] that the methods described above perform better than more complex information extraction techniques, e.g. key-phrase extraction.

Basic Hook-based Query Formulation. As already introduced in Section 6.2, documents in our model are assumed to contain a set of hooks explicitly representing the information needs of the reader or, more precisely, what requires contextualization to be understood and interpreted. The analysis done in [Ceroni et al., 2014b] showed that contextualization hooks are not only entity mentions, concept mentions, but also general terms and even short phrases. We consider two basic hook–based query formulation methods: *all_hooks* and *each_hook*. *All_hooks* includes all the hooks for a document in a single query, representing a tailored perspective of the user's combined information needs for the document. *Each_hook* queries each hook separately, focusing on specific information about single actors, aspects, or sub–topics of the document. The queries generated by these methods are augmented with the title of the document, under the assumption that it is a good representative of the document's topic. We also experimented with more advanced methods based on identifying hook relationships, for instance considering their co–occurrence in a document collection. However, since these approaches

did not perform better than the *all_hooks* method described before, we will not discuss them further.

6.2.2 Learning to Select Hook-based Queries

Different methods based on ranking and selection of query terms from an initial query might be employed [Bendersky and Croft, 2008, Lee et al., 2009, Maxwell and Croft, 2013], considering the entire set of hooks for a document as the initial query. We explore an adaptive method which formulates queries based on the characteristics of the input document and hooks. Our approach consists in predicting the performances of candidate queries representing sub–sets of hooks for a given document, ranking them according to the predicted performance, and selecting the top-m of them to be actually performed for the document. The value of m is identified through experiments. In contrast to previous works in query performance prediction, the prediction model is trained on recall performances instead of precision. Furthermore, we define novel features for query performance prediction that explicitly take the temporal dimension into account. Finally, our method assesses performances of subsets of query terms (hooks) and can generate more than one query (subsets of hooks).

Candidate Queries. Given a document d and the set of its hooks H_d, we compute its power set $\mathcal{P}(H_d)$ and we create a candidate query for each set of hooks $p \in \mathcal{P}(H_d)$. Again, candidate queries are augmented with the title of the document. The effort of the computation of features for each element in the power set is not critical in our scenario for two reasons. First, working with short text like news articles limits the number of hooks within the text. Second, the features employed to predict the query performances (discussed in the next paragraph) are either pre-retrieval measures, which can be computed off–line, or do not require heavy post-retrieval computation.

Features. We measure the performances of each candidate query in terms of its recall because, as already explained, at retrieval phase we are interested in retrieving as much contextualization candidates as possible. In this work we predict query performances with a regression model learned via Support Vector Regression (SVR) [Drucker et al., 1997]. In this model, each learning sample $s = (\boldsymbol{f}_q, r_q)$ consists in a feature vector $\boldsymbol{f}_q$ describing query q (as well as the document it refers to) and its recall r_q, i.e. the label to be predicted. Note that different numbers of top–l results can be used to compute the recall, i.e. the labels, and this choice is discussed in Section 6.2.5. The feature set that we use to represent queries and the document it belongs to are described in the rest of this section. It is composed of novel temporal features for query performance prediction, along with more standard features [Carmel and Kurland, 2012, He and Ounis, 2004, Mothe and Tanguy, 2005].

We compute a family of *linguistic features* [Mothe and Tanguy, 2005] for a query by considering its text and the document it refers to. This results in a set of features both at query and document level: the length of the query, in words; the number of duplicate terms in the query; the number of entities (people, locations, organization, artifacts) in the

query; the number of nouns in the query; the number of verbs in the query; the number of hooks in the query; the length of the document's title; the length of the document's lead paragraph; the number of entities in the document (title and lead paragraph); the number of nouns in the documents; the number of verbs in the document; the number of hooks for the document; the number of duplicates in the document.

The *Document Frequency* of a hook h represents the percentage of contextualization units in the corpus containing h and it is computed as:

$$df(h) = \log \frac{N_h}{N} \tag{6.1}$$

where N_h is the number of contextualization units in the corpus containing h and N is the size of the corpus. At document level, we compute the document frequency for every hook of the document the query belongs to, i.e. $df\,(h)\,\forall h \in H_d$, and then we derive aggregate statistics like average, standard deviation, maximum value, minimum value. Similarly, at query level, we compute $df\,(h)$ for every hook in the query and we derive the same aggregate statistics as before. In the following, we will refer to average, standard deviation, maximum value, and minimum value simply as *aggregate statistics*.

In order to restrict the popularity of a term to a particular time period $T = [t_0 - w; t_0 + w]$, we compute Equation 6.1 only for those contextualization units having at least one temporal reference contained in T. This can be done efficiently since contextualization units in our corpus have been annotated with the temporal references mentioned in them. The time period we are interested in is centered around the publication date of the document, i.e. $t_0 = p_d$, and the parameter w determines the width of the interval. After experimenting different values of w, we set $w = 2$*years* for our dataset.

The *scope* of a query has been defined in [He and Ounis, 2004] as the percentage of documents (contextualization units in our case) in the corpus that contain at least one query term. Besides the scope of the query itself, we also compute the scope of the document title and the scope of the document hooks H_d when queried together.

We define the *temporal scope* of a query as the percentage of contextualization units in the corpus that contain at least one query term and at least one temporal expression within a given time period. The time period that we consider is the same as the one considered for the computation of temporal document frequency, i.e. a period centered around the publication date of the document and with temporal window equal to $2w$. Again, we experimented different values of w and we set $w = 2$*years*.

For a given query q, we retrieve the top-k contextualization units and we compute aggregated statistics of their relevance scores given by the underlying retrieval model. The value of k has been empirically set to 100 after experimenting different candidate values. We also computed relevance features at document level, using both document's title and document's hooks as queries.

For a given query q generated from a document d and every retrieved contextualization unit c in its top-k result set (again, $k = 100$), we compute the *temporal similarity* between q and c and we derive aggregated statistics over the elements in the result set. Tempo-

ral similarity between time points t_1 and t_2 is computed through the time-decay function [Kanhabua and Nørvåg, 2010]:

$$TSU(t_1, t_2) = \alpha^{\lambda \frac{|t_1 - t_2|}{\mu}} \tag{6.2}$$

where α and λ are constants, $0 < \alpha < 1$ and $\lambda > 0$, and μ is a unit of time distance. The temporal similarity between a query q and a result c is computed as $\max_{t \in T_c}\{TSU(t, p_d)\}$, where T_c is the set of temporal references mentioned in c and p_d is the publication date of the document q refers to. This can be done efficiently since temporal references mentioned in contextualization units have been extracted and stored at indexing time.

We also computed temporal similarity features at document level, using both document's title and document's hooks as queries. The computation of the features is the same as the one described above.

We observed that changing the function parameters did not affect the correlation capabilities of the feature, and we set $\lambda = 0.25$, $\alpha = 0.5$, and $\mu = 2$*years* in our experiments.

6.2.3 Context Ranking

We now describe the methods used in addressing the second part of the re-contextualization process: retrieving and re-ranking context. For the retrieval step, given the queries generated from different methods for each document described in previous section, we use a retrieval model based on language modelling to create a ranked list of contextualization candidates. Later, learning to select relevant context items is applied to this ranked list.

Retrieval Model. For the retrieval step, we use query-likelihood language modelling [Ponte and Croft, 1998] to determine the similarity of a query with the context. In particular, given a query q generated by using one of the methods described in Section 6.2.1 for the document d, we compute the likelihood of generating the query q from a language model estimated from a context c with the assumption that query terms are independent.

$$P(c|q) \propto P(c) \prod_{w \in q} P(w|c)^{n(w,q)} \tag{6.3}$$

where w is a query term in q, $n(w, q)$ the term frequency of w in q, and $P(w|c)$ the probability of w estimated using Dirichlet smoothing:

$$P(w|c) = \frac{n(w, c) + \mu P(w)}{\mu + \sum_{w'} n(w', c)} \tag{6.4}$$

where μ is the smoothing parameter, $P(w)$ is the probability of w in the collection.

To combine the rankings produced by each query of a document, we exploited two combining methods namely round-robin, which chooses one result from each ranked list, skipping any result if it has occurred before, and CombSUM, which sums up a result's scores

from all ranked lists where it was retrieved. In the experiment, we observed that round-robin method achieves better performance than CombSUM especially in terms of recall, which also reported in [Tsagkias et al., 2011]. Therefore, we decided to use round-robin method for combining different ranked lists.

Learning to Rank Context. Once we have obtained a ranked list of contextualization candidates for each document, we turn to context selection (re-ranking) where we need to decide which of the context items are most viable. Our ranking algorithm needs to balance two goals, i.e., high topical and temporal relevance for the document, as well as complementarity for providing additional information. In this work, we use supervised machine learning, that takes as input a set of labelled examples (context to document mappings) and various complementarity features of these examples similar to diversity features [Zhang et al., 2002].

The first class that we employ is *Topic Diversity*, which is aimed to compare the dissimilarity between document d and context c on a higher level by representing them using topics. We use latent Dirichlet allocation (LDA) [Blei et al., 2003] to model a set of implicit topics distribution of the document and context. We define this feature as follows.

$$R_1(c,d) = \sqrt{\sum_{k=1}^{m}(p(z_k|d) - p(z_k|c))^2}$$

where m is the number of topics, z_k is the topic index.

We also considered *Text Difference* as feature: in this case, we represent the document and context as a set of words. The novelty of context c is measured by the number of new words in the smoothed set representation of c. If a word w occurred frequently in context c but less frequently in document d, it is likely that new information not covered by d is covered by c. For computation, document and context are represented by a set of informative words (removing stop words, stemming) denoted by $Set(d)$ and $Set(c)$ respectively. We compute this feature as follows.

$$R_2(c,d) = \|Set(c) \cap \overline{Set(d)}\|$$

The way of computing *Entity Difference* is similar to the one for text difference, with the difference that document and context are represented by a set of entities. The feature is denoted as $R_3(c,d)$.

Anchor texts can be regarded as a short summary (i.e., a few words) of the target document and captures what the document is about. This feature can be computed similarly as text and entity features, and is denoted as $R_4(c,d)$. We extract anchor texts using WikiMiner [Milne and Witten, 2008] with a confidence threshold γ.

The next feature we use is *distribution similarity*, which is denoted as $R_5(c,d)$.

$$R_5(c,d) = -KL(\theta_c, \theta_d) = -\sum_{w_i} P(w_i|\theta_c) \log \frac{P(w_i|\theta_d))}{P(w_i|\theta_c)}$$

where θ_d and θ_c are the language models for document d and context c, respectively and are multinomial distributions. We compute θ_d (and similarly for θ_c) using maximum likelihood estimation (MLE) given as:

$$P(w_i|d) = \frac{tf(w_i, d)}{\sum_{w_j} tf(w_j, d)}$$

The problem with using MLE is that if a word never occurs in document d, it will be a zero probability $P(w_i|d) = 0$. Thus, a word in context c but not in document d will make $KL(\theta_c|\theta_d) = \infty$. In order to solve this problem, we make use of Dirichlet smoothing method.

$$P_\lambda(w_i|d) = \frac{tf(w_i, d) + \lambda p(w_i)}{\sum_{w_j}(tf(w_j, d) + \lambda p(w_j))}$$

There are several ways to compute geometric distance measure, such as, Manhattan distance and Cosine distance. We leverage Cosine distance because of its robustness to document length.

$$R_6(c, d) = \cos(c, d) = \frac{\sum_{k=1}^{n} w_k(c) w_k(d)}{\|d\| \, \|c\|}$$

In our experiment, we used each unique word as one dimension, the *tf.idf* score as the weight of each dimension.

In order to retrieve high topical and temporal relevant contextualization candidates for the document, we consider also relevance and temporal features. For the former one, we exploit the retrieval scores of context returned by our retrieval model. For the later one, we apply temporal similarity measurement, i.e., TSU which has been described previously.

6.2.4 Experimental Setup

Document Collections. In our experiments, we used the New York Times Annotated Corpus, which contains 1.8 million documents from January 1987 to June 2007, as the document collection to be contextualized. For context source, we employed Wikipedia because it is considered the largest and most up-to-date online encyclopedia covering a wide temporal range of general and specific knowledge. We obtained the Wikipedia dump of February 4, 2013 and considered *paragraphs* as contextualization units. In this particular snapshot, we have 4,414,920 proper articles that contain 25,708,539 paragraphs. For each paragraph, we used Stanford CoreNLP [Manning et al., 2014] for tokenization, entity annotation and temporal expression extraction. In addition, *anchor* texts found in the paragraph hyperlinks are also extracted. We used Apache Solr[16] to index the annotated paragraphs.

[16] https://lucene.apache.org/solr/

Ground-truth Dataset. In order to obtain ground-truth dataset (both for training and evaluation), we ultimately picked a set of 51 articles that spanned a wide range of topics (business, technology, education, science, politics, and sports) focusing on the older ones (29 articles published in 1987, 2 articles in 1988, 6 articles in 1990, 7 articles in 1991, and 7 articles in 1992) and recruited six human annotators to manually annotate those articles.

The annotators were presented with an annotation interface with which they can evaluate article/context pairs (relevant or non-relevant). The annotation guidelines specified that the annotators should assign relevant to the context that provides *additional information* which complements the information in the article and does help to understand the article to some extent. For each article, we retrieved up to 20 candidate context with each query formulation method and removed duplicates afterwards.

In total, our annotation dataset consists of 9,464 article/context pairs, where the annotators evaluated 26.9 relevant context per article on average. To foster further research on this challenging task, our ground-truth dataset is publicly available.[17] We measured the inter-annotator agreement using Cohen's kappa statistic. We averaged the pairwise kappa values of all possible combinations of annotators that had overlapping candidates they had annotated and we obtained a fair agreement of κ_c = 0.37 given the high complexity of this contextualization task, which includes objectivity and subjectivity.

Parameter Settings. For query performance prediction, the regression model described in Section 6.2.2 was built by using the Support Vector Regression implementation of LibSVM[18]. In particular, we trained a n–SVR model with Gaussian Kernel through 10–fold cross validation. The open parameters were tuned via grid search to $C = 3$, $\gamma = 0.5$, $\nu = 0.75$. Linguistic features were extracted using Stanford CoreNLP [Manning et al., 2014].

For re-ranking context, we performed 5-fold cross validation at document level. We reported scores averaged over all testing folds. We conducted experiments using several machine learning algorithms to confirm the robustness of our approach, i.e., it does not depend on any specific algorithm. In this paper, we employed Random forests (RF), RankBoost (RB) and AdaRank that are implemented in RankLib[19]. In order to compute topic-based feature, we employed a topic modelling tool Mallet[20] by specifying the number of topics to 100, for this task. In addition, we set the confidence threshold to $\gamma = 0.3$ for extracting anchor texts using WikiMiner. For smoothing, we set μ = 2000 and $\lambda p(w_i)$ = 0.5.

Evaluation Metrics. The evaluation metrics, we considered precision at rank 1, 3, 10 (P@1, P@3, P@10 respectively), recall, and mean average precision (MAP). These measures provide a short summary of quality of the retrieved context. In our experiment, a context is considered relevant if it is marked as relevant by an annotator, otherwise we consider it as non-relevant. We used the top-20 returned context for evaluation because

[17] http://www.l3s.de/~ntran/contextualization/

[18] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[19] http://sourceforge.net/p/lemur/wiki/RankLib/

[20] http://mallet.cs.umass.edu/topics.php

it is not expected that readers consider more than 20 contextualization units. Statistical significance was performed using a two-tailed paired t-test and is marked as ▲ and △ for a significant improvement (with p ¡ 0.01 and p ¡ 0.05 respectively), and significant decrease with ▼ and ▽ (for p ¡ 0.01 and p ¡ 0.05 respectively).

Baselines. For comparing to our approach, we considered three following competitive baselines.

M&W. The method proposed by Milne and Witten [Milne and Witten, 2008] which represents the state-of-the-art in automatic linking approaches. We use the algorithm and best-performing settings as described in [Milne and Witten, 2008]. In order to apply this method for our task, we consider all paragraphs of all linked pages as a candidate set.

LM. The standard query-likelihood language model is used for the initial retrieval as described in Section 6.2.3 which provides the top retrieved documents as a candidate set for the contextualization task.

Time-aware Language Model (LM-T). Since we aimed at adding context to past stories, the temporal dimension is important. We selected a state-of-the-art time-aware ranking method, which has been shown very effective for answering temporal queries, as our third baseline. It assumes the textual and temporal part of the document d are generated independently from the corresponding parts of the context c, yielding

$$P(d|c) = P(d_{text}|c_{text}) \times P(d_{time}|c_{time}) \tag{6.5}$$

where d_{time} is the document's publication date, c_{time} is the set of temporal expressions in the context c.

The first factor $P(d_{text}|c_{text})$ can be computed by Eq. 6.3 and Eq. 6.4. The second factor in (6.5) is estimated, based on a simplified variant of [Berberich et al., 2010], as

$$P(d_{time}|c_{time}) = \frac{1}{\mid c_{time} \mid} \sum_{t \in c_{time}} P(d_{time}|t) \tag{6.6}$$

If the document has zero probability of being generated from the context, Jelinek-Mercer smoothing is employed, and we estimate probability of generating the document's publication date from context c as

$$P(d_{time}|c_{time}) = (1-\lambda)\frac{1}{\mid C_{time} \mid} \sum_{t \in C_{time}} P(d_{time}|t) + \lambda \frac{1}{\mid c_{time} \mid} \sum_{t \in c_{time}} P(d_{time}|t) \tag{6.7}$$

where $\lambda \in [0, 1]$ is a tunable mixing parameter which is set to $\lambda = 0.5$ in our experiment (changing this parameter does not affect our results), and C_{time} refers the temporal part of the context collection treated as a single context and $P(d_{time}|t)$ is estimated by using time-decay function, i.e., TSU computed as in Eq. 6.2.

6.2.5 Results of Query Formulation

We evaluate and compare the performances of the different query formulation methods described in Section 6.2.1, focusing on recall metric. The results reported in the rest of this section are averaged over the 51 documents in our dataset.

In order to fairly evaluate and compare the recall capabilities of the different methods, which can generate different numbers of queries, we allow each method to retrieve the same number of results k. The choice of the method that we used to create a single result set of k elements from different ranked lists have been discussed in Section 6.2.3.

Prediction Performances. The query formulation method described in Section 6.2.2 is based on predicting the performances (recall in our case) of candidate queries, ranking them according to the prediction, and then using the top-m queries to retrieve results. Thus, the quality of the query performance prediction itself has to be evaluated before assessing and comparing the performances of the whole query formulation method.

The regression model has been trained via 10–fold cross validation, and the results reported hereafter have been averaged over the 10 folds. The Correlation Coefficient is equal to 0.973, the Root Mean Squared Error equal to 0.056, and the Mean Absolute Error equal to 0.037. The low error values and high correlation value, if compared with the performances in predicting query precision reported in previous works (e.g. [Raiber and Kurland, 2014, Carmel and Yom-Tov, 2010]), show that the recall of queries in our task can be predicted quite accurately by using the features described in Section 6.2.2.

Feature Analysis. In order to analyse which are the most important features in our model, we identified the top–10 features according to their absolute correlation coefficient. Referring to Section 6.2.2, these are: *max query relevance*, *number of hooks in document*, *min document's hooks df*, *max document's hooks temporal df*, *document's hooks scope*, *avg query temporal similarity*, *document's title temporal scope*, *std query relevance*, *avg document's title temporal similarity*, and *std query temporal similarity*. The presence of temporal document frequency, temporal similarity, and temporal scope shows that the temporal features that we defined play an important role in the model. We can also note that both query–level and document–level features are important, since the set is made of 4 features from the former and 6 features from the latter class. Finally, there is only one linguistic feature in the set, namely the number of hooks in the document, confirming that this class of features alone does not correlate well with query performances [Carmel and Kurland, 2012].

Comparison of Query Formulation Methods.

We now compare recall values for the document–based methods (*title*, *lead*, *title+lead*), the basic hook–based methods (*each_hook*, *all_hooks*), as well as the method based on query performance prediction, hereafter called qpp. For the latter method, we report the performances achieved when using prediction models trained with different labels: we experimented with different l values, namely $l = 50, 100, 200$, for the computation of the

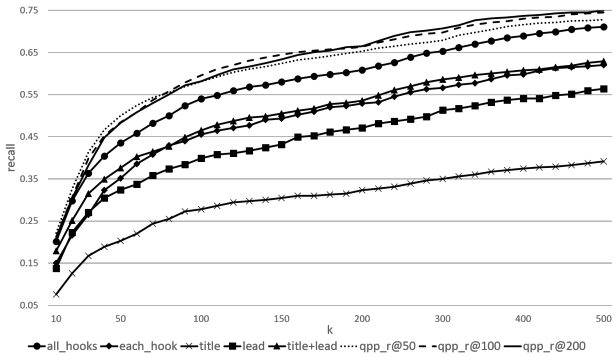

Figure 6.3: Recall curves of document-based and hook-based methods.

recall at l to be used as label.

These three methods will be called *qpp_r@50*, *qpp_r@100*, *qpp_r@200* respectively in the rest of the experiments. Note that each qpp method considered here uses the top-2 queries, according to their predicted performances, to retrieve the results.

The choice of selecting $m = 2$ queries will be explained and motivated later in this section.

The recall curves of the different methods, for different values of top–k results, are shown in Figure 6.3. The curves of $title$ and $lead$ are the lowest ones, while their combination ($title + lead$) becomes comparable with $each\_hook$. Querying using all the hooks of a document together, i.e. all_hooks, exposes higher recall values than all the aforementioned methods, showing that performing hook–based queries does lead to better performances in terms of recall with respect to document–based methods. The difference in performances between $each\_hook$ and all_hooks is due to the fact that querying all the hooks together prefers contextualization candidates that contain many hooks. These are potentially more relevant, as they refer to different aspects (hooks) of the same document.

Regarding the qpp methods, for $k > 20 - 30$, the recall values achieved are between 3% and 7% higher than the ones obtained by all_hooks. For larger values of k, e.g. $k > 400$, the difference between the qpp methods and all_hooks reduces because the prediction models used by the qpp methods have been optimized for lower values of k (recall that $l = 50, 100, 200$).

This means that, if the number of k results to be retrieved for the re-ranking phase is known and fixed in advance, this information can be exploited early in the training of the query performance prediction model by setting $l = k$, leading to higher recall values for that particular k.

Another comparative analysis between qpp methods and all_hooks can be done by categorizing the documents according to their *difficulty*, which we define in terms of the amount of relevant context that can be retrieved for a given document. This means that difficult documents are those for which few relevant context can be retrieved, before the re-ranking phase. We categorize documents in $easy$ and $hard$ with respect to the all_hooks method, since it represents a baseline in this comparative analysis with qpp methods.

	R@50		R@100		R@200	
	qpp	*all_hooks*	*qpp*	*all_hooks*	*qpp*	*all_hooks*
easy	0.6208	0.5666	0.7361	0.6969	0.7951	0.7686
hard	0.3837	0.3094	0.4606	0.3892	0.5391	0.4550

Table 6.1: Recall of *all_hooks* and *qpp* methods over different classes of documents (based on their retrieval difficulty).

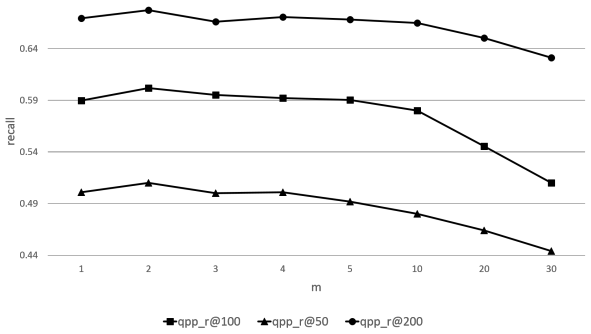

Figure 6.4: Recall values of $qpp\_r@50$, $qpp\_r@100$, and $qpp\_r@200$ by varying the number of top–m queries.

The splitting of the documents in easy and hard was performed by considering the recall at $k = 200$ achieved by all_hooks for the different documents. Since the recall values associated to the different documents exhibited a uniform distribution, we split the document set in two equal parts, one representing easy documents and the other representing hard documents.

Table 6.1 shows the performances of $qpp\_r@50$, $qpp\_r@100$, and $qpp\_r@200$ compared to the ones of all_hooks for the different categories of difficulty. The comparison between each qpp method and all_hooks is done considering the recall at those k value used to train the prediction model (i.e. $k = l$, $l = 50, 100, 200$). Besides $qpp\_r@50$, $qpp\_r@100$, and $qpp\_r@200$ are on average better than all_hooks both for easy and hard documents, their improvements are greater for hard documents. In case of $qpp\_r@100$, for instance, the relative improvement with respect to the recall value achieved by all_hooks is 5.6% for easy documents and 18.3% for hard documents. We believe that the capability of getting higher recall improvements for documents whose relevant context units are difficult to retrieve is a considerable characteristic for the qpp methods.

As a conclusion, in this section we proved that exploiting hooks in query formulation is more effective, in terms of recall, than document–based query formulation methods. Moreover, we showed that learning to select candidate hook-based queries can be better, again in terms of recall, than the basic hook–based query formulation methods.

Number of Queries. The number of top ranked queries that qpp methods perform is an open parameter, which we tuned via an empirical analysis observing the recall performances when selecting different numbers of top–m ranked queries. Recall that, for sake

	P@1	P@3	P@10	MAP	Recall
Document-based query models					
title	0.2156	0.1895	0.1745	0.2446	0.1211
lead	0.4902▲	0.4641▲	0.3333▲	0.4908▲	0.2603▲
title + lead	0.5294▲	0.4705▲	0.3901▲	0.5161▲	0.2723▲
Basic hook-based query models					
each_hook	0.3333	0.3464	0.2745	0.4003	0.1969
all_hooks	0.5490	0.5098	0.4137	0.5640	0.2979
Query performance prediction model					
qpp_r@100	**0.5882**	**0.5490▲**	**0.4529▲**	**0.5802▲**	**0.3097▲**

Table 6.2: Retrieval performance of document-based and hook-based query models. The significance test is compared with Row 1 (within the first group) and Row 3 (for the second and third groups).

of fair comparison, we allow each method to pick the same number of results k from the result lists retrieved by the queries that it generated for a given document. This means that increasing the number of queries to be selected and performed does not necessarily lead to higher recall.

Figure 6.4 shows the recall values achieved by $qpp\_r@50$ (computed at top–50 results), $qpp\_r@100$ (computed at top–100 results), and $qpp\_r@200$ (computed at top–200 results) for different numbers of top–m queries selected. A common trend over the different curves can be observed: they stay quite stable for small values of m, exhibiting a little peak for $m = 2$, and then they decrease for increasing values of m. After observing this behaviour, we decided to fix the number of performed queries to $m = 2$.

6.2.6 Results of Context Ranking

We report the retrieval performance of different query formulation methods and analyse the effectiveness of our context ranking methods trained using different machine learning algorithms. Firstly, we investigate the performance of the standard, well-known Wikification technique, i.e., the M&W method, in retrieving contextualization candidates. Our experiment considers all paragraphs of all linked pages as candidates. The results obtained using the M&W method achieve the low recall value of 0.2290; thus indicating that current semantic linking approaches are not appropriate for the contextualization task.

In Table 6.2, we show the results of different query formulation methods. The first group (top) reports results for candidate retrieval based on document-based query models in which the best performing model is *title + lead* that uses content from the article's title and lead paragraph. Turning into models derived from contextualization hooks, Table 6.2 shows that the *qpp_r@100* model is performing the best among all hook-based query models and significantly improves over *title + lead* on all metrics.

	P@1	P@3	P@10	MAP	Recall
all_hooks	0.5000	0.3462	0.2885	0.4487	0.2217
qpp_r@100	0.5000	**0.4743**$^{\triangle}$	**0.3730**$^{\triangle}$	**0.5048**$^{\triangle}$	**0.2357**

Table 6.3: Retrieval performance of *all_hooks* and *qpp_@100* on a set of difficult documents.

Similar to the previous experiment, Table 6.3 reports the results of *all_hooks* and *qpp_-@100* retrieval baselines on a subset of difficult documents (here recall is computed on top-20 candidates). On this subset, *qpp_r@100* also shows significant improvement over *all_hooks* in terms of precision. In short, the results on different query formulation methods indicate that using hook-based approaches outperform the document-based approach that based on merely article internal structure. Using the query performance prediction method obtains the highest performance on all metrics, followed by *all_hooks*.

We now present the results of our re-ranking approach when using a set of innovative complementarity features to further improve performances of the context ranking step, especially in terms of precision. We select *title + lead* for the document-based approach and *all_hooks, qpp_r@100* for the hook-based approach.

	P@1	P@3	P@10	MAP	Recall
title + lead					
LM	0.5294	0.4705	0.3901	0.5161	0.2723
RF	**0.7672**$^{\blacktriangle}$	0.5757$^{\triangle}$	**0.4909**$^{\blacktriangle}$	**0.6170**$^{\blacktriangle}$	**0.3522**$^{\blacktriangle}$
RB	0.6036	**0.5945**$^{\triangle}$	0.4694$^{\blacktriangle}$	0.5945	0.3417$^{\blacktriangle}$
Adabank	0.6254	0.5406	0.4143	0.5457	0.3249
all_hooks					
LM	0.5490	0.5098	0.4137	0.5640	0.2979
RF	**0.8272**$^{\blacktriangle}$	**0.6630**$^{\blacktriangle}$	**0.5014** $^{\blacktriangle}$	**0.6427**$^{\triangle}$	0.3611$^{\blacktriangle}$
RB	0.7855$^{\blacktriangle}$	0.6593$^{\blacktriangle}$	0.5009$^{\blacktriangle}$	0.6475$^{\triangle}$	**0.3637**$^{\blacktriangle}$
AdaRank	0.6472	0.5836	0.4687	0.6034	0.3372$^{\triangle}$
qpp_r@100					
LM	0.5882	0.5490	0.4529	0.5802	0.3097
RF	**0.8054**$^{\blacktriangle}$	**0.6993**$^{\blacktriangle}$	0.5140$^{\blacktriangle}$	0.6498$^{\blacktriangle}$	**0.3951**$^{\blacktriangle}$
RB	0.7218	0.6915$^{\blacktriangle}$	**0.5300**$^{\blacktriangle}$	**0.6632**$^{\blacktriangle}$	0.3792$^{\blacktriangle}$
AdaRank	0.6072	0.6139	0.4895	0.6109	0.3479$^{\blacktriangle}$

Table 6.4: Retrieval performance of different machine-learned ranking methods compared to the best performing retrieval baselines.

The first (top) group in Table 6.4 shows the results when applying machine learning to *title + lead* retrieval baseline. All three algorithms are able to improve precision at rank k, MAP and Recall. Random forest (RF) and RankBoost (RB) obtain significant improve-

ment where RF achieves the highest scores on most metrics, except precision at rank 3 where RB is the best. The second (middle) group reports the results of *all_hooks* retrieval baseline, augmented by the re-ranking step. In this case, RF and RB are again able to significantly improve over *all_hooks* on all metrics while Adarank is also performing significantly better than *all_hooks* in terms of recall. Among three algorithms, RF achieves the highest results, except for recall. Similarly, all three machine learning algorithms perform significantly better than the *qpp_@100* retrieval baseline. Again, in this case RF obtain the highest performances, closely followed by RB.

In order to compare our approach to time-aware language model which takes into account temporal information, we use the queries derived from query performance prediction method, i.e., *qpp_@100* that obtain the highest results among our query formulation methods. Table 6.5 shows that using time-aware language models is not efficient in our case. This is possibly due to that lots of relevant context (paragraphs in our case) do not have any temporal information as shown in Figure 6.5. Consequently, these candidates are ranked low (e.g., higher than 20) in the ranked list returned by LM-T. This result indicates that purely using the time dimension in context retrieval is not sufficient in the contextualization task. It also confirms the importance of complementarity that is used in our re-ranking step.

qpp_@100	P@1	P@3	P@10	MAP	Recall
LM-T	0.5882	0.4967	0.4176	0.5446	0.2796
LM	0.5882	0.5490	0.4529	0.5802	0.3097$^{\triangle}$
RF	**0.8054**$^{\triangle}$	**0.6993**$^{\blacktriangle}$	**0.5140**$^{\blacktriangle}$	**0.6498**$^{\blacktriangle}$	**0.3951**$^{\blacktriangle}$

Table 6.5: Retrieval performance of our proposed ranking method and the state-of-the-art time-aware language modelling approach. The significance test is compared against LM-T.

News article - *Maj. Gen. Richard V. Secord, a main organizer of the Iran arms sales and the contra supply operation, testified today that he had been told that President Reagan had been informed that proceeds from the sales to Iran had been diverted to the Nicaraguan rebels.*

Context - *Speaking of the Iran-Contra affair, a Reagan administration scandal that involved the diverting of funds being shipped to Iran to the contras in Nicaragua, Reagan says, "None of the arms we'd shipped to Iran had gone to the terrorists who had kidnapped our citizens." Of the scandal, Reagan writes, " and, I presume, knew how deeply I felt about the need for the contras' survival as a democratic resistance force in Nicaragua. Perhaps that knowledge... led them to support the contras secretly and saw no reason to report this to me." He also says of himself, "As president, I was at the helm, so I am the one who is ultimately responsible." Also, Reagan discusses his political rivalry and personal friendship with former Speaker of the House Tip O'Neill.*

Figure 6.5: Example of contextualization candidate for a given document with no explicit temporal information.

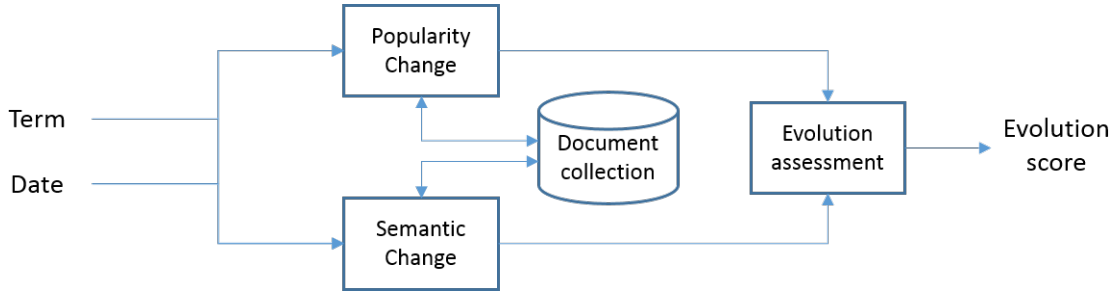

Figure 7.1: Overview of the term temporal evolution assessment.

7 Context Evolution

The long-term digital preservation of documents, which is the main goal of the project, imposes to consider potentially wide time intervals between archiving a document and bringing it back to the active system. In such periods, the context of a document is likely to evolve, and any proposed archiving strategy has to be aware of it. Common changes in context would include changes in organizational roles, personal relationships, as well as the general meaning of terms. Some of the main issues are, for instance, how big should the amount of context change be in order to trigger the archiving of a new version, and how to update the original context in the archived documents based on it.

A first, necessary step towards the design of archiving strategies to handle context evolution is indeed the detection of context changes themselves. Further moving to a lower level of abstraction, the context evolution can be modelled based on the evolution of its building blocks, which are terms in case of the textual domain. For this reason, we adopt a bottom-up approach by first considering how time affects the meaning and common knowledge of terms. Since context can be regarded as a set of terms (apart from multimedia context), we believe that other forms of context-evolution at higher levels will benefit and work on top of such low-level assessment of term evolution. In this section, we present the high-level overview, ideas, and work in progress of our approach to term evolution, which will be developed and expanded in the next future.

The high-level workflow for assessing the temporal evolution of terms is shown in Figure 7.1. Our definition of terms, besides actual single-word terms, include also multi-word concepts and entities like ”Financial Crisis” or ”Angela Merkel”. The input is represented by a term along with a date, indicating the starting time to consider when assessing the evolution of the term (it is likely the publication date of the document the term belongs to). Two criteria are considered to assess the evolution of the input term: changes in popularity and semantics of the term. These criteria, described afterwords, are estimated taking into account a given document collection whose document have to span a *sufficiently* wide time span. The meaning of the term *sufficiently* is likely to depend on the scenario and should be empirically estimated. Finally, the two individual measures are merged together to assess the overall amount of evolution of the input term. We give more details about each component in the following subsections.

7.1 Document Collections

A document collection is required as a ground truth for performing any kind of temporal evolution assessment. For instance, the popularity of a term at a given point in time can be estimated by counting how many times a term is mentioned within the documents belonging to that time point. In order to make such kind of analyses, the documents in the collection have to span a sufficient time period, where the actual meaning of the term *sufficient* is likely to depend on the analysis to be performed and it is subject to investigations. Currently, we are considering two document collections: the New York Times Annotated Corpus[21], spanning a time period of 20 years from 1987 to 2007, and the Annotated English GigaWord dataset[22], containing news articles of different news agencies from 1994 to 2010.

7.2 Popularity Changes

We consider the change in popularity of a term as a possible indicator of how much it is universally known. The underlying hypothesis is that the more a term is used in every day documents like news articles, the more it is popular and commonly known by people. This assessment can be useful for both contextualization and re-contextualization tasks. For the former, when preparing a document for being archived, more context might be provided for those terms that are perceived as not currently popular and then possibly hard to be understood. For the latter, when re-contextualizing an old document, more context might be retrieved for those terms that are not popular at re-contextualization time, although they might have been popular at the publication time of the article. Still referring to re-contextualization, the fact that a term was not popular at publication time might be an indicator of peculiarity and importance of the term in the context of the whole article, thus demanding for re-contextualization as well.

7.3 Semantic Changes

The semantic change (or drift) of a term is defined as changes in meaning and usage of the term over time. For instance, the term *egregious* nowadays means something offensive or deplorable, but in the past it was used to describe something remarkably good. Cases like this might lead to misunderstandings when reading an old document, and re-contextualization should be aware of such changes and make both the interpretations of the same term clear to the reader. On one hand, the current interpretation of a term or entity, e.g. *Barack Obama*, would help the user to collocate him into the current context, being aware that he is the president of the United States. On the other hand, the information that at the time of the document Barack Obama was a senator might be crucial to

[21] http://catalog.ldc.upenn.edu/LDC2008T19
[22] http://catalog.ldc.upenn.edu/LDC2012T21

fully understand the document. These twofold aspects will likely play a role in the update of the context already stored in the archive. Updating the contextual information for a document or an entity, either when significant changes happens or just periodically, might help in having a fresh updated perspective of the context. For instance, in a personal scenario, when one gets married he/she might want to update the context information of the partner accordingly. However, some of the content store within the archive could be fully understandable only when being aware of the original context, thus keeping it would be necessary as well. One possibility to tackle this problem could be keeping track of the overall evolution of the context by storing significant snapshots. This would have implications in terms of storage space required to store such information, and possible trade-offs should be experimented.

Our first attempt to model semantic changes of terms is based on considering the other terms it co-occurs with. The intuition is the following: if the set of terms that co-occurs with a given term changes over time, then it might be an indicator of usage and semantic changes of the given term. Going beyond the surface level of terms, semantic changes of terms could be modelled with changes in the topic of the text or excerpt the terms belongs to.

7.4 Evolution Assessment

The previously described measures of popularity and semantic change have a standalone validity and utility for contextualization, re-contextualization, and context-evolution. We also plan to investigate the merging of these and other indicators that might come up, in order to associate an overall evolution measure to each term. The complexity of such combination might span from simple weighted averages to supervised machine learning, where the degree of evolution assessed by human evaluators would be used as label.

8 Conclusions

8.1 Summary

This deliverable has described the second release of components for contextualizing both images and text which are based upon the state-of-the-art as previously documented [Ceroni et al., 2013]. These components build upon the first prototypes released earlier in the project [Ceroni et al., 2014a]. We have also included a more approachable foundation for contextualization (see Section 2) based around a user-centric view of the process.

8.2 Assessment of Performance Indicators

This section contains three short discussions explaining how, and to what extent, the components described in this deliverable fulfil the success indicators of the expected outcomes of this work-package, as given in the description of work.

8.2.1 Documented Conceptualization of Information Contextualisation

The success indicators for this objective are the *availability of a framework for modelling context* and the *availability of a suitable evaluation scheme*. Both of these indicators are discussed in Section 2. This user centric discussion of contextualization, coupled with a more formal approach [Ceroni et al., 2014a], provide a framework for modelling context. While evaluation is discussed in less detail than the framework, the section makes it clear that as well as evaluating the components independently an end-to-end evaluation is required. This is, as noted, difficult, however, as it is unclear what context is required until a document is used in the future; context acts as a memory trigger and the required context will differ based upon elapsed time, who is wanting to use a document, and the reason for use. The plan (as discussed further in Section 8.3) is to evaluate the role of contextualization within the tools developed for WP9 and WP10.

8.2.2 Contextualisation Tools and Framework

The success indicators for this objective are the *availability of a set of context-identification modules that perform adequately according to the . . . evaluation scheme* and the *successful integration of the contextualisation tools with the pilot implementations in WP9 and WP10*. This deliverable describes three components for contextualization that have been evaluated in isolation and shown to perform well. These components are being integrated into the use case tools (see for example Section 4. This integration will, as discussed previously, allow for a full end-to-end evaluation to also be performed.

8.2.3 Techniques for Dealing with Information Evolution

The success indicators for this objective are context *evolution supported by the designed and implemented methods and tools to deal with information evolution*, the *degree of change in active use context that can be covered by re-contextualization*, and the *successful evaluation . . . of information re-contextualisation when faced with information evolution*. This deliverable describes one component for performing re-contextualization which uses assigns up to date contextual information to old documents which while not addressing context evolution directly does implicitly handle a changing world environment. Section 7 deals directly with the evolution of context and presents an initial approach to the problem. This, of the three expected outcomes for this work-package, is the least developed a situation which will be rectified before the third and final release of the ForgetIT contextualization components.

8.3 Next steps

It should be clear from this deliverable that a significant amount of work has taken place within the workpackage on both the development of components for contextualization as well as the underlying framework that forms the basis of our understanding of contextualization within ForgetIT. It is also clear, however, that there are a number of areas which should be the main focus of work within the final period of the project. These areas are context evolution, integration with the use case tools, and a thorough end-to-end evaluation.

As reported in Section 7 work is underway on techniques for context evolution although this is currently in the initial stages of development. There has been some integration of the components included in this deliverable within the use case tools (notable in WP9) and plans are in place to accelerate this integration a natural outcome of which will be tools which can be used to perform an end-to-end evlaution of contextualization. The results of these planned activities, along with updated and improved versions of the components already delivered will be documented in the final WP6 deliverable due at the conclusion of the project.

References

[ReC,]

[Aswani et al., 2013] Aswani, N., Gorrell, G., Bontcheva, K., and Petrak, J. (2013). D2.2.2: Multilingual, Ontology-Based IE from Stream Media - v2. Technical report, TrendMiner Consortium.

[Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

[Bendersky and Croft, 2008] Bendersky, M. and Croft, W. B. (2008). Discovering key concepts in verbose queries. In *SIGIR '08*.

[Berberich et al., 2010] Berberich, K., Bedathur, S. J., Alonso, O., and Weikum, G. (2010). A language modeling approach for temporal information needs. In *ECIR '10*.

[Bizzozero et al., 2004] Bizzozero, I., Lucchelli, F., Prigione, A., Saetti, M., and Spinnler, H. (2004). What do you remember about Chernobyl? A new test of memory for media-mediated events. *Neurological Sciences*, 25(4):205–215.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*

[Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

[Carmel and Kurland, 2012] Carmel, D. and Kurland, O. (2012). Query performance prediction for ir. In *SIGIR '12*.

[Carmel and Yom-Tov, 2010] Carmel, D. and Yom-Tov, E. (2010). Estimating the query difficulty for information retrieval. In *SIGIR '10*.

[Ceroni et al., 2013] Ceroni, A., Greenwood, M. A., , Kanhabua, N., Mezaris, V., Niederée, C., Nilsson, J., and Papadopoulou, O. (2013). D6.1: State of the Art and Approach for Contextualization. Technical report, ForgetIT Consortium.

[Ceroni et al., 2014a] Ceroni, A., Greenwood, M. A., Mezaris, V., Niederée, C., Papadopoulou, O., and Solachidis, V. (2014a). D6.2: First Release of Tools for Contextualization. Technical report, ForgetIT Consortium.

[Ceroni et al., 2014b] Ceroni, A., Tran, N. K., Kanhabua, N., and Niederée, C. (2014b). Bridging temporal context gaps using time-aware re-contextualization. In *SIGIR '14*.

[Conci et al., 2014] Conci, N., Natale, F. D., and Mezaris, V. (2014). Synchronization of multi-user event media (sem) at mediaeval 2014: Task description, datasets, and evaluation. In *MediaEval 2014 Workshop, Barcelona, Spain*.

[D4.3, 2015] D4.3 (2015). :information analysis, consolidation and concentration techniques, and evaluation - second release. Technical report, ForgetIT Consortium.

[Daiber et al., 2013] Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, pages 121–124, New York, NY, USA. ACM.

[Damhuis et al., 2014] Damhuis, A., Doan, P., Dobberkau, O., Dörzbacher, M., Goslar, J., Krasteva, V., Niederée, C., Schaffstein, S., Sprenger, S., and Wolters, M. (2014). D10.1: Organizational Preservation Use Cases and Mockup Development. Technical report, ForgetIT Consortium.

[Drucker et al., 1997] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1997). Support vector regression machines. In *NIPS '97*.

[Eldesouky et al., 2015] Eldesouky, B., Bakry, M., Maus, H., and Dengel, A. (2015). Supporting Early Contextualization of Textual Content in Digital Documents on the Web. In *Submitted to the 13th International Conference on Document Analysis and Recognition (ICDAR2015)*.

[Ferragina and Scaiella, 2012] Ferragina, P. and Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75.

[Goetz, 2011] Goetz, T. (2011). Harnessing the Power of Feedback Loops. *WIRED Magazine*.

[He and Ounis, 2004] He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *In Proc. Symposium on String Processing and Information Retrieval*. Springer Verlag.

[Hoffart et al., 2011] Hoffart, J., Yosef, M. A., Bordino, I., Furstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, 2011*, pages 782–792. Morgan Kaufmann, California.

[Ji and Grishman, 2011] Ji, H. and Grishman, R. (2011). Knowledge base population: Successful approaches and challenges. In *Proc. of ACL'2011*, pages 1148–1158.

[Kanhabua and Nørvåg, 2010] Kanhabua, N. and Nørvåg, K. (2010). Determining time of queries for re-ranking search results. In *ECDL '10*.

[Kuny, 1998] Kuny, T. (1998). The Digital Dark Ages? Challenges in the Preservation of Electronic Information. *International Preservation News*, (17).

[Lee et al., 2009] Lee, C.-J., Chen, R.-C., Kao, S.-H., and Cheng, P.-J. (2009). A term dependency-based approach for query terms ranking. In *CIKM '09*.

[Manning et al., 2014] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *ACL '14*.

[Maus et al., 2013a] Maus, H., Dobberkau, O., Wolters, M., and Niederée, C. (2013a). D9.1: Application Use Cases and Requirements Document. Technical report, ForgetIT Consortium.

[Maus and Schwarz, 2014] Maus, H. and Schwarz, S. (2014). D9.2: Personal Preservation Use Cases and Mockup Development. Technical report, ForgetIT Consortium.

[Maus et al., 2013b] Maus, H., Schwarz, S., and Dengel, A. (2013b). Weaving personal knowledge spaces into office applications. In Fathi, M., editor, *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, pages 71–82. Springer.

[Maxwell and Croft, 2013] Maxwell, K. T. and Croft, W. B. (2013). Compact query term selection using topically related text. In *SIGIR '13*.

[Mezaris et al., 2010] Mezaris, V., Sidiropoulos, P., Dimou, A., and Kompatsiaris, I. (2010). On the use of visual soft semantics for video temporal decomposition to scenes. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 141–148.

[Milne and Witten, 2008] Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *CIKM '08*.

[Mothe and Tanguy, 2005] Mothe, J. and Tanguy, L. (2005). Linguistic features to predict query difficulty. In *In ACM SIGIR 2005 Workshop on Predicting Query Difficulty - Methods and Applications*.

[Papadopoulou et al., 2014] Papadopoulou, O., Mezaris, V., Solachidis, V., Ioannidou, A., Eldesouky, B. B., Maus, H., and Greenwood, M. A. (2014). D4.2: Information Analysis, Consolidation and Concentration Techniques, and Evaluation - First Release. Technical report, ForgetIT Consortium.

[Ponte and Croft, 1998] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98*.

[Raiber and Kurland, 2014] Raiber, F. and Kurland, O. (2014). Query-performance prediction: Setting the expectations straight. In *SIGIR '14*.

[Rao et al., 2013] Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, Multi-lingual Information Extraction and Summarization*. Springer.

[Tran et al., 2015] Tran, N. K., Ceroni, A., Kanhabua, N., and Niederée, C. (2015). Back to the past: Supporting interpretations of forgotten stories by time-aware re-contextualization. In *WSDM '15 (to appear)*.

[Tsagkias et al., 2011] Tsagkias, M., de Rijke, M., and Weerkamp, W. (2011). Linking online news and social media. In *WSDM '11*.

[Zhang et al., 2002] Zhang, Y., Callan, J., and Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *SIGIR '02*.

A Text Contextualization Evaluation Metrics

This appendix discusses in detail the evaluation metrics used in Section 5 and how they are possibly affected by the systems and corpora under evaluation.

Precision describes the proportion of named entities found by the system that are correct. That is, of all the times that the system identified a particular entity as being referred to in a particular location, how often was it correct? This statistic penalizes overgeneration–although a system that finds many entities will be more likely not to miss one, it will generate more wrong answers and thus have a lower precision.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall describes the proportion of times that the system correctly found a named entity that is present. That is, of all the named entities in the text, how often did the system find it? This statistic penalizes undergeneration–although a system that only identifies named entities it is sure of will have a high precision, recall will be low.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Since precision and recall trade off against each other to a great extent, a combined metric provides a more comparable statistic. The F-measure offers this, and can be tuned for situations in which precision (or recall) is more important. In our task, we have no argument for why precision or recall would be more important, so we use the metric F1.

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Where an entity is located and correctly linked but the location is not exactly correct, we consider this to be a partially correct answer. Overlapping but not identical spans often occur in the NERD task because there can be room for opinion about where a named entity starts and ends. For example, in “The White House”, is “the” part of the named entity or not? In “Prime Minister David Cameron”, is “Prime Minister” part of a single named entity referring to David Cameron, or is it a separate named entity referring to a political position in the United Kingdom? We consider it in the spirit of the task to count partially correct answers as being correct, because span errors are of little consequence in the utility of a system. Responses in the correct location but linked to the wrong entity are, of course, not counted at all. For this reason, we use variants on the above metrics known as “lenient” metrics, which calculate precision and recall as follows:

$$Precision = \frac{TruePositives + Partials}{TruePositives + Partials + FalsePositives}$$

$$Recall = \frac{TruePositives + Partials}{TruePositives + Partials + FalseNegatives}$$

The separate task of evaluating the extent to which a system, given the correct location, is able to link the right referent requires a different metric. Accuracy describes the proportion of named entities in the document that were correctly linked by the system. It is a similar statistic to recall except that nils are handled differently, in that they are not excluded from the calculation. This is in the spirit of the TAC KBP task, where a large number of nil entities are given in the evaluation data (named entities that don't have a referent in DBpedia) and correctly identifying these nils is considered an important part of the task. In the statistics given above, an entity in an evaluation corpus that has not been linked, but instead has been annotated as a nil, is treated as though it is not there. For accuracy, however, nils that have not been annotated by a system or have been annotated as a nil are treated as being correct.

$$Accuracy = \frac{Correct}{TotalNamedEntities}$$

As above, we present, for all systems, a lenient accuracy. Particularly in the case of accuracy, the systems are not being evaluated for their ability to locate the named entities, but for their ability to correctly disambiguate them. Therefore, we do not penalize for span variations. In the case that the system creates two named entities overlapping the key span (for example, one overlaps the start of the key span and the other, the end) we evaluate only the first.

$$LenientAccuracy = \frac{Correct + Partial}{TotalNamedEntities}$$

Depending on the corpus, the impact of nils varies widely. For corpora with many nils, accuracy may be quite different to recall.