Document ID: PT001

PSEUDONYMISATION IN CLEF

Subject:	*A summary of the pseudonymisation issues facing the CLEF project and the tools available at Sheffield that could be modified to support the process.*
Original Author:	*Ian Roberts <i.roberts@dcs.shef.ac.uk>*
Document Status:	*Draft*
Reference:	*PT001*
Summary:	*This document describes the challenges faced by the CLEF project in terms of pseudonymisation of patient records. It details existing technologies developed at Sheffield for other projects, which could form part of a pseudonymisation tool, and outlines the additional work that would be necessary to produce a usable system.*

1. Background

1.1. The need for pseudonymised records

To achieve the aims of the CLEF project, we will need to process potentially sensitive medical information about cancer patients. Ethically, the source hospital or hospitals cannot release such information to us if it could potentially cause harm to patients. However, without the information, key parts of the project cannot make progress. There are at least two ways around this problem.

1. Only use records from patients who have already died. We cannot cause any more harm to such patients, though harm or embarrassment may still come to their relatives.

2. Remove personally identifiable information from the records before releasing them to the project. This may still be desirable for the records of dead patients, for the reasons stated above.

Therefore, if we cannot be granted access to original, identifiable records, some way must be found to de-identify them.

1.2. Steps to pseudonymisation

Pseudonymisation of a document is essentially a three-step process

1. Identify the sections of the document that contain potentially identifying information

2. Find and mark the identifying information in the document (noting the type of information, and maybe linking instances which relate to the same individual)

3. Replace the identifying information with pseudonymous information in a consistent way, i.e. so that instances which related to the same individual in the original text bear the same relation to each other in the pseudonymised version.

It could be argued that steps 1 and 2 are part of the same step, the reasons for separating them will become clear below.

For step 3, there are a number of approaches that could be adopted, e.g.:

- Simply remove the identifying data, replacing it with e.g. "xxxx"

- Replace identifying data with a class indicating the type of the original data, e.g. "[HOSPITAL]"

This is fairly straightforward, but information is lost in the conversion, as it is no longer apparent from the text whether all the "[HOSPITAL]" tags refer to the same hospital, or different ones. An improvement on this scheme would be:

- Add a tag to the class-based identifiers, to distinguish between different entities of the same class: "[HOSPITAL-1]", "[HOSPITAL-2]", etc.

To enable the building of an accurate longitudinal record of care, these cross-references would need to be consistent across all the documents in a single patient's record. If it were desirable for the CLEF system to answer queries such as "show me 5-year survival rates for this type of cancer, grouped by the hospital where they were treated," then the identifiers would need to be consistent across all patients' records.

The final possibility is to replace the identifying data with realistic faked data, i.e. replace names with other names, dates with different dates (maintaining the chronological ordering, at least within a single record), etc. This approach would mean that the tools developed in CLEF would work without change on raw, non-pseudonymised patient records in the future.

In an ideal world, the whole pseudonymisation process would be carried out by a fully automatic tool. In practice, steps 1 and 3 could be fully automated, but step 2 will generally require human intervention, if the pseudonymisation is to be 100% complete and accurate.

2. Existing tools

2.1. Amilcare

Amilcare is an adaptive information extraction system, able to learn how to mark up strings in text with XML tags. Given a corpus of training documents that have been manually marked up, Amilcare uses some of them to learn rules that would allow it to duplicate the manual markup. The rules are tested by applying them to the remaining documents, and tuned appropriately. If Amilcare's suggestions can be manually checked and corrected, the system can continue to learn, eventually reaching a high level of accuracy. At this point, the system can be left to mark up unseen texts unsupervised.

2.2. Melita

Melita builds on Amilcare by providing a more intuitive and user-centred interface to generate the training data. It provides for point-and-click annotation of the original bare documents, and feeds the user's annotations into Amilcare to use for training. As Amilcare learns, it is used to generate suggested annotations for the documents the user is yet to annotate, these suggestions can be confirmed or ignored by the user, and will feed back to Amilcare for further training. Once the system is fully trained, Melita can be removed from the process entirely, and the learned rules can be used by a standalone Amilcare system to perform automatic annotation.

3. Requirements for a pseudonymisation tool

Melita forms a good starting point for building a pseudonymisation tool, however it lacks a number of features which would make it significantly more suited to the task. These can be roughly divided into bugs in the current implementation, those features that are essential to make the tool usable at all, and those that would improve usability but are not essential.

3.1. Some terminology

The following glossary describes some terms used when referring to Melita and Amilcare.

Entity the string marked up with an annotation. Two occurrences of the same string refer to the same entity.

Annotation a mark signifying that a particular string in the document refers to an entity of a certain type.

Concept the type of entity an annotation refers to. A Melita session specifies an ontology of concepts that defines the types of annotation available to the user.

3.2. Essential features

If a user annotates a string in a document with a particular concept, Melita will assume that all other identical strings in the same document refer to the same entity, and will suggest them as annotations for the same concept. This feature will need extending to cover all occurrences of the identical string across *all* documents in a given patient's record. Then, for example, a user could annotate the patient's name once, and Melita would then annotate all other occurrences of the same name in the other documents.

Amilcare incorporates GATE (a General Architecture for Text Engineering). Within GATE, there are modules available which attempt to identify strings in a document which represent names, locations, dates, etc. Amilcare bases the rules it learns on these identified "named entities". While GATE can identify a string as a possible name, it cannot in general tell whether a recognised name is that of a patient, doctor, relative, etc. However, Melita could be modified to make some initial suggestions for recognised entities. This would not affect the learning process for Amilcare, but would speed up the process of training. An example would make this clearer:

If GATE has recognised the string "S ANTHONY" as a name, and the Melita ontology lists "doctor" and "patient" as types of name, the user would be given the option to accept the suggested string as a doctor name, or as a patient name, or to reject the suggestion entirely.

Melita does not currently allow existing tags to be extended or contracted. For example, if a document contains the string "the patient saw Dr. Anthony last week", and Melita initially suggests "Anthony" as the name of a doctor, the user should be able to visually extend the tag to cover the word "Dr." as well. Currently, the misplaced tag would have to be removed and a new tag created in the correct location, requiring the user to relocate the appropriate concept in the hierarchy.

3.3. Useful features

The documents we will be processing in CLEF will likely have a fairly consistent structure – letters tend to begin with a reference number and date, followed by an address, a salutation (which includes a doctor's name), the body of the letter, and finally the closing "yours sincerely" and associated names and titles. Some of these sections, e.g. the address, would have to be completely replaced with pseudonymous information. If it is possible to automatically identify these sections, Melita could be set up to warn the user if they had failed to pseudonymise them completely. This would improve the accuracy of the training data, particularly if a single user needs to pseudonymise a large amount of data.

Other workflow-related improvements could include support for defining pseudonymisation protocols to be followed for large data sets, for example, first pseudonymise the address section of every document, then once these have been verified they can be ignored for the rest of the process, which concentrates solely on the body of the narrative. A method of marking which documents had passed through which

stages of the protocol would enable more rigorous quality control of the results, and make the whole process less onerous for the annotators.

3.4. A new tool?

In view of the above requirements, it may be preferable to create a new tool with a more appropriate user interface to support pseudonymisation. This would consist of a core annotation engine, based on Melita and Amilcare, with additional pre- and post-processing components to provide the section recognition, initial named entity suggestions, multi-document markup of identical strings and final replacement of the annotated data with pseudonyms. We would envisage such a tool being used to train an adaptive system, which could then be used standalone for large scale batch pseudonymisation.