UNIVERSIDAD
POLITÉCNICA
DE MADRID

Task 6.5: AI4Healthcare: Improve quality constancy of medical images reports

Elvira Amador-Domínguez
Asunción Gómez-Pérez
Daniel Manrique
Emilio Serrano

Index

1 Available radiology datasets

After an extensive search, five radiology-specific datasets have been identified. Each dataset has unique features that will be described below, but the following general lines can be drawn:

- They are composed mostly by images, some of them associated with a brief medical report
- Those containing reports are generally written in English
- The language used in the reports is full of acronyms and abbreviations. While some refer to medical terms, others can be just contractions of common words.
- The available data is usually labelled with respect to a specific medical terminology (MeSH, SNOMED, UMLS…)

1.1 MIMIC-CXR

This dataset belongs to the PhysioNet project, developed by the MIT. This dataset is supposedly one of the best dataset regarding radiology reports. Though it is publicly available, you need to be a credentialed user to access. In order to become a credentialed user, a preparation course (about 2/3 hours long) is required to learn the basics of medical user data treatment. Due to its size (about 5TB), its available not only for direct download, but also as a Google Cloud Bucket.

The data contained within has the following specifications:

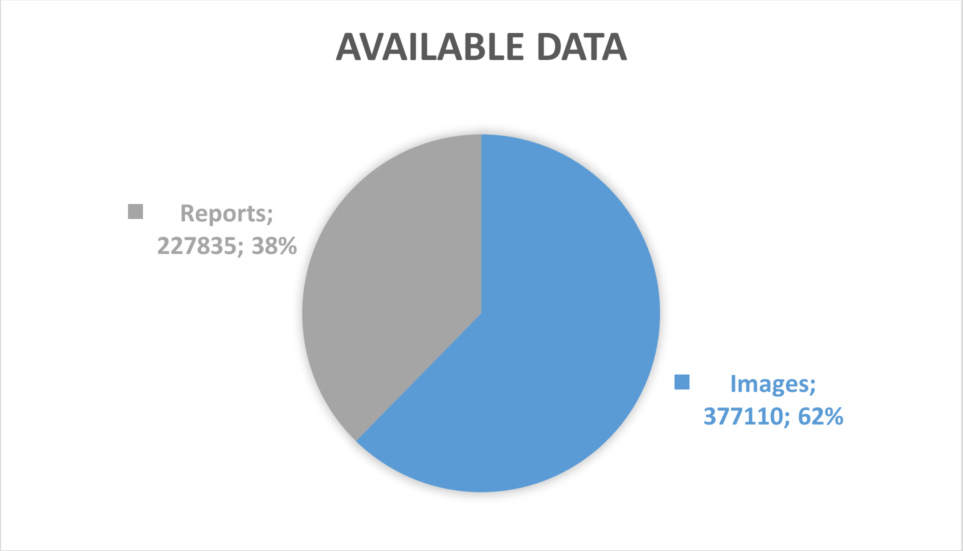
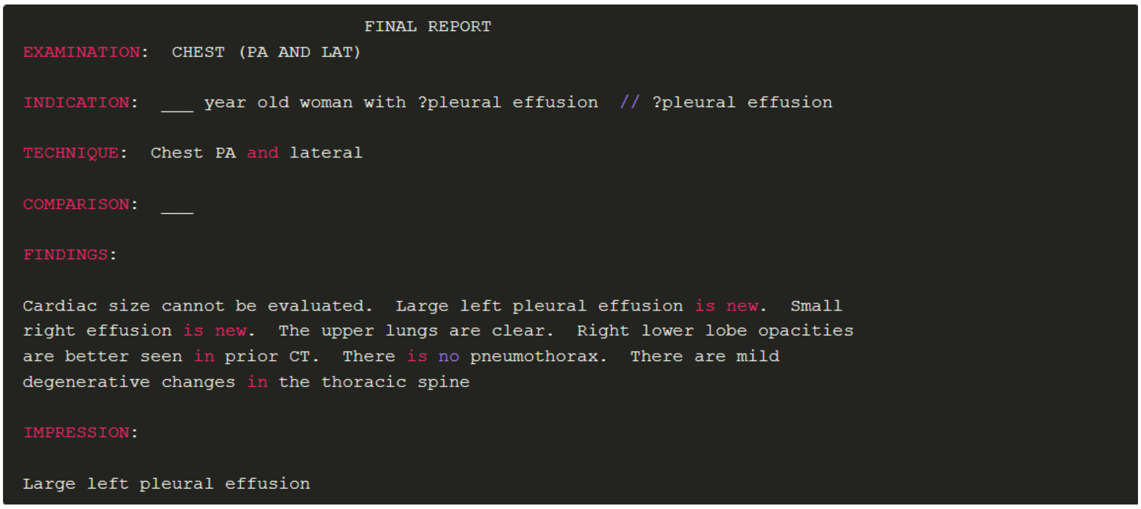


Figure 1 MIMIC-CXR Data Distribution

Each report is associated with one or more images. These studies have been obtained from 64,588 patients, between 2011 and 2016.

- **Images:** The extracted images refer to chest radiographs, which are compliant with the DICOM (Digital Imaging and Communications in Medicine) format. Thus, for each medical image, a set of structured meta-data in DICOM format is provided.
- **Reports:** Reports come in plain txt format. They have, however, a semi-defined structure. Reports are templated, and each individual line is comprised by no more than 79 characters, and three underscores have been placed where sensible information should be, in order to anonymize patient data. However, the template can be modified by the radiologist at will. Generally, a report is comprised by six sections: *Examination,*

Indication, Technique, Comparison, Findings and *Impression*. The following image shows an example of a radiology report from MIMIC-CXR:



```
                                 FINAL REPORT
EXAMINATION:  CHEST (PA AND LAT)

INDICATION:  ___ year old woman with ?pleural effusion  // ?pleural effusion

TECHNIQUE:  Chest PA and lateral

COMPARISON:  ___

FINDINGS:

Cardiac size cannot be evaluated.  Large left pleural effusion is new.  Small
right effusion is new.  The upper lungs are clear.  Right lower lobe opacities
are better seen in prior CT.  There is no pneumothorax.  There are mild
degenerative changes in the thoracic spine

IMPRESSION:

Large left pleural effusion
```

Figure 2 MIMIC-CXR Sample Report

1.2 ChestXRay

This dataset was first introduced in 2017 by the NIH. It comprises only images (about 100,000), which have been manually labelled. For each image, three different radiologists were asked to label the image. An iterative labelling procedure was followed, where radiologists were asked until consensus on the final label was reached. If there was no consensus between the radiologists, the majority vote label was assigned.

Eight different labels, corresponding to the most common thoracic diseases: *Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia* and *Pneumothorax*.

This dataset does not contain any radiology reports, and subsequently can only be used for image-related tasks, such as the development of image diagnosis systems. It is available for direct download and as a Google Cloud Bucket.

1.3 CheXpert

Similar to ChestXRay, CheXpert, a dataset from the Stanford ML Group, comprises a set of 227,943 imaging studies. These images are labelled according to 14 different observations, which can have four different values: *blank*-unmentioned, *0*-negative, -*1*-uncertain, *1*-positive.

These labels are extracted by an automatic rule-based labelling tool, which takes the free text radiology reports assigned to the images as input. To label each report, three distinct steps are performed: mention extraction, mention classification and mention aggregation. In mention extraction, the labeller extracts mentions from the *Impression* section of the report. Then, in the mention classification stage, each of the extracted mentions is classified as negative, uncertain or positive. Finally, in the mention aggregation step, a final label for each observation is extracted based on the value given to each mention in the previous stage.

1.4 Stanford’s LanglotzLab Datasets

The Radiology Informatics Research Group from Stanford has two different datasets: one for image processing tasks and one for natural language processing tasks. According to their webpage, they possess:

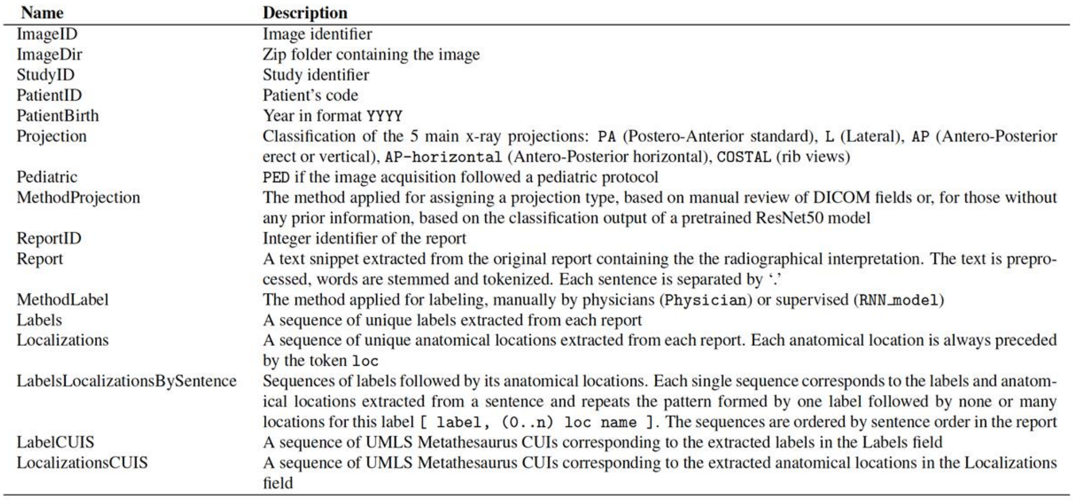
- **Images**: 1000 ICU chest radiographs, 831 bone tumor annotated radiographs, 4,000 annotated mammographs, 4,000 pediatric hand radiographs.
- **Reports**: 4.4 million narrative radiology reports from Stanford, 1 million narrative radiology reports from other institutions, 430,000 radiology reports identified as normal, 110,000 chest ct reports annotated for presence/absence of pulmonary embolism and 150 chest CT reports with all concepts annotated.

Even though this is the biggest dataset and the most complete regarding radiology reports, it has not been disclosed to the general public. You can only access this data if you are part of one of the Stanford projects or if you are granted access as a collaborator.

1.5 PadChest

This dataset was set up by the Universidad de Alicante, containing over 160,000 images with their corresponding text reports. The reports are labelled with 174 radiographic findings, 19 diagnoses and 104 anatomical locations, which are organized hierarchically following the UMLS (Unified Medical Language System) standard terminology. About 30% of the reports were manually labelled by experts, while the remainder was labelled by a RNN. The generated labels, however, are still highly accurate.

The dataset comes into two parts: a folder containing PNG images in DICOM format, and a csv file containing the metadata of the images, alongside with the assigned labels, the associated report and some additional information. The following table provides a summary of the data contained in the dataset:

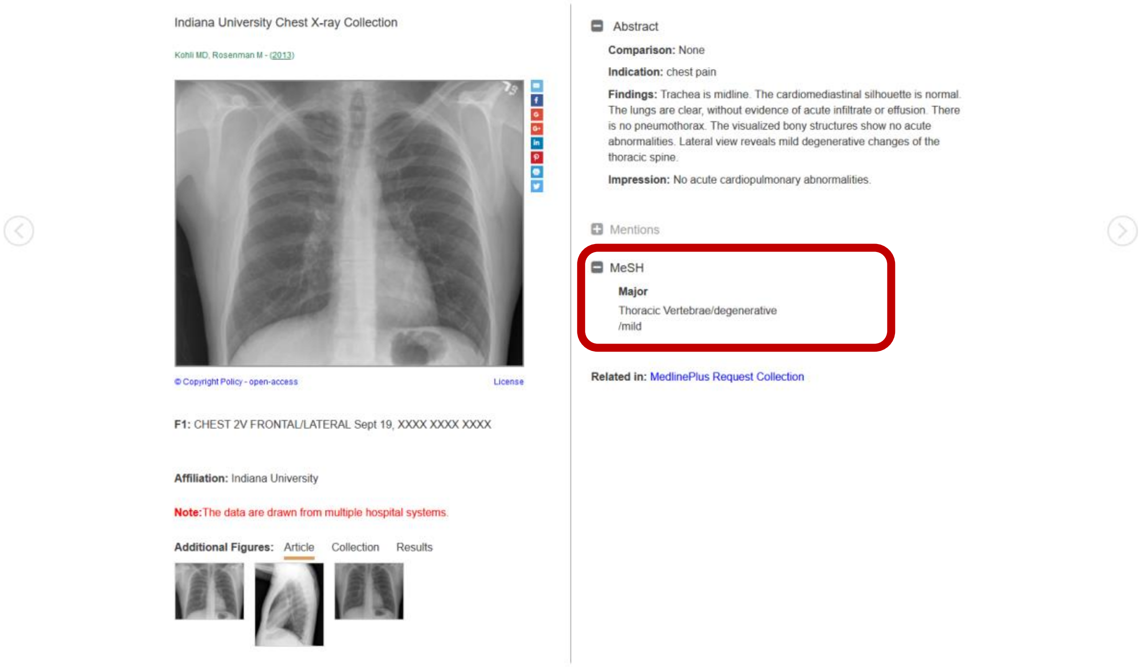


Name	Description
ImageID	Image identifier
ImageDir	Zip folder containing the image
StudyID	Study identifier
PatientID	Patient’s code
PatientBirth	Year in format YYYY
Projection	Classification of the 5 main x-ray projections: PA (Postero-Anterior standard), L (Lateral), AP (Antero-Posterior erect or vertical), AP-horizontal (Antero-Posterior horizontal), COSTAL (rib views)
Pediatric	PED if the image acquisition followed a pediatric protocol
MethodProjection	The method applied for assigning a projection type, based on manual review of DICOM fields or, for those without any prior information, based on the classification output of a pretrained ResNet50 model
ReportID	Integer identifier of the report
Report	A text snippet extracted from the original report containing the the radiographical interpretation. The text is preprocessed, words are stemmed and tokenized. Each sentence is separated by ‘.’
MethodLabel	The method applied for labeling, manually by physicians (Physician) or supervised (RNN_model)
Labels	A sequence of unique labels extracted from each report
Localizations	A sequence of unique anatomical locations extracted from each report. Each anatomical location is always preceded by the token loc
LabelsLocalizationsBySentence	Sequences of labels followed by its anatomical locations. Each single sequence corresponds to the labels and anatomical locations extracted from a sentence and repeats the pattern formed by one label followed by none or many locations for this label [label, (0..n) loc name]. The sequences are ordered by sentence order in the report
LabelCUIS	A sequence of UMLS Metathesaurus CUIs corresponding to the extracted labels in the Labels field
LocalizationsCUIS	A sequence of UMLS Metathesaurus CUIs corresponding to the extracted anatomical locations in the Localizations field

Figure 3 PadChest Dataset Description

1.6 Open-I

Open-I is not a radiology dataset, but a set of different collections of medical data, such as ortopedical surgical reports, images for tuberculosis detection, etc. Amongst these data, there is a collection named *Indiana U.Chest X-rays*, which contains radiologies and radiology reports. While in the previous datasets images needed to be downloaded alongside the reports, in this datasets images and reports are disassociated. Thus, images can be downloaded in PNG or DICOM format, while reports come in XML format. To associate images with their corresponding reports, an XML tag is set with the identifier of each image. Moreover, images can be retrieved via an API. The reports contain the following information: *Comparison, Indication, Findings* and *Impression*. It also contains a MeSH field, containing the specific headings of the report. In this case, the radiography shows thoracic vertebrae in a degenerative state with mild grade.



Indiana University Chest X-ray Collection

Kohli MD, Rosenman M - (2013)

© Copyright Policy - open-access

License

F1: CHEST 2V FRONTAL/LATERAL Sept 19, XXXX XXXX XXXX

Affiliation: Indiana University

Note: The data are drawn from multiple hospital systems.

Additional Figures: Article Collection Results

Abstract

Comparison: None

Indication: chest pain

Findings: Trachea is midline. The cardiomediastinal silhouette is normal. The lungs are clear, without evidence of acute infiltrate or effusion. There is no pneumothorax. The visualized bony structures show no acute abnormalities. Lateral view reveals mild degenerative changes of the thoracic spine.

Impression: No acute cardiopulmonary abnormalities.

Mentions

MeSH

Major

Thoracic Vertebrae/degenerative
/mild

Related in: MedlinePlus Request Collection

According to MeSH taxonomy, the observed concept would be Thoracic Vertebrae, which is also part of a taxonomy whose root is 'Musculoskeletal System'. Moreover, *degenerative* acts as a qualifier of the concept, and mild acts as a grader of the qualifier.

This linking with MeSH not only benefits further operations with the reports, but enables interoperability with other datasets such as PadChest, as it has been mapped to UMLS.

2 External Resources

The proposed framework comprises several subtasks, making it necessary to find suitable resources to achieve them. These subtasks are: Named Entity Recognition, Image Processing, Text Classification and Text Embedding. In order to make the framework customizable and easily adaptable to different languages, these models can be changed accordingly. For this first version of the framework, the following libraries were selected to fulfil the specified tasks:

2.1 SciSpacy

SciSpacy is an extension of Python's Spacy library developed by the Allen NLP group. This extension focuses on the biomedical domain, thus providing models specifically trained for the purpose. This powerful tool not only performs most of the usual NLP basic tasks, such as dependency parsing, POS tagging, tokenization, etc., but performs Named Entity Recognition over the input text. Additionally, SciSpacy includes the option of performing abbreviation disambiguation, as well as entity linking. The later, however, is a feature that is still under development. This feature just associates each named entity with its corresponding entry on the UMLS terminology, based on regular expressions, and provides a confidence score for each matching.

Aside from performing the aforementioned NLP tasks, pretrained SciSpacy models also include embedding models, generating for each input text a tensor matrix containing the vectors for each detected token, a single vector representing the text that is obtained by averaging each token and a list of each token detected with its associated vector. Aside from English, this library contains several other language models, including French. **This library is included in our framework as the text embedding module**, as well as being used for minor NLP tasks such as tokenization. The embeddings generated by this module are also used in the text classification task.

2.2 CliNER

CliNER is a modern, named entity recognition framework specifically developed for the clinical domain. This open source, powerful framework, uses two different algorithms to perform entity recognition: Conditional Random Fields and Long-Short Term Memories. Moreover, it provides a pretrained model, making it usable straightforwardly. The pretrained model contains data from the MIMIC-II dataset, thus only works for English-written reports, and its capable of distinguishing between three types of entities: Problem, Treatments and Tests. However, it can be easily retrained on a new and sufficient labelled corpus.

Though these types may seem quite general, they are specific enough for our application domain, as no further information is usually reflected in radiology reports. Thus, **this framework was selected to serve as the Named Entity Recognition module.**

2.3 OpenCV

OpenCV is the cornerstone library of image processing in Python. This library contains several functionalities, ranging from simple image loading functions, to keypoint detection algorithms. Thus**, this library is used as the image processing module** (alongside with Pytorch).

2.4 Scikit

Scikit is also one of the most spreadly-used Python libraries, being one of the most complete scientific libraries there are. This library comes in several submodules, including scikit-learn, containing all machine learning related functionalities, and scikit-image, dedicated to image processing. However, as we selected OpenCV as our image processing library, only scikit-learn is used in this project, **for text classification,** implementing the scoring model**.** If new data needs to be used, it is first required to retrain the text embedding module, as the output of this module serves as entry for the current.

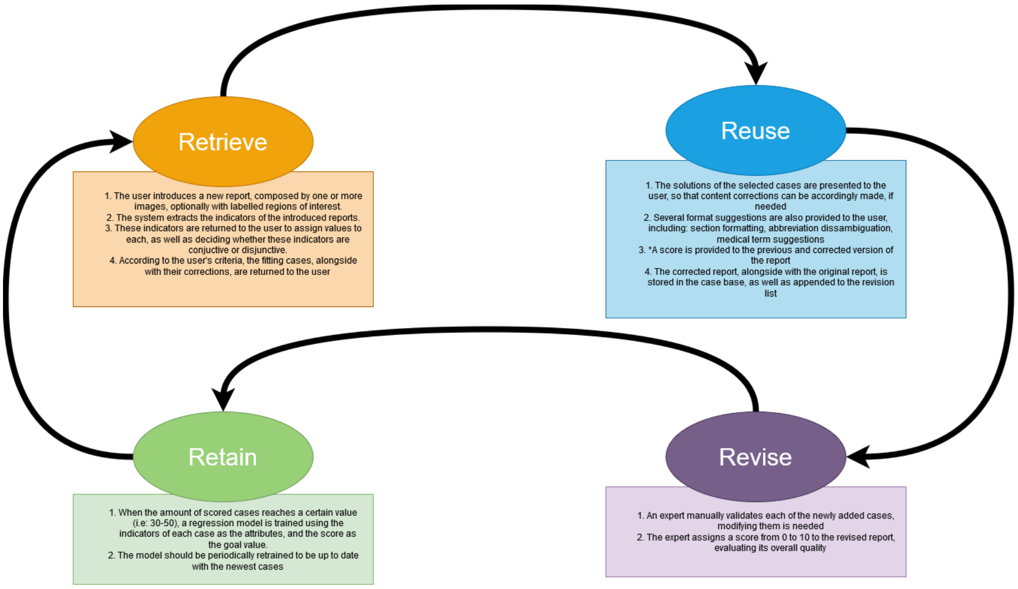
2.5 Pytorch

Finally, to support those functionalities related with deep learning, Pytorch is used. As in the case of Scikit, Pytorch also comes separately on two libraries: one general library and one image-specific library. Both are used in this project. As mentioned above, OpenCV is used jointly with Pytorch for image processing. In this module, Pytorch is used to extract image features using pretrained CNNs, as this functionality is unavailable in OpenCV.

Moreover, Pytorch is also used for text classification, implementing the sectioning model.

3 Proposal

The following diagram represents the current proposal of the model, formulated as a Case-Based Reasoning model supported by different deep learning techniques. For the first sprint of the project, synthetic data was used to develop and test the system.



1. The user introduces a new report, composed by one or more images, optionally with labelled regions of interest.
2. The system extracts the indicators of the introduced reports.
3. These indicators are returned to the user to assign values to each, as well as deciding whether these indicators are conjuctive or disjunctive.
4. According to the user's criteria, the fitting cases, alongside with their corrections, are returned to the user

1. The solutions of the selected cases are presented to the user, so that content corrections can be accordingly made, if needed
2. Several format suggestions are also provided to the user, including: section formatting, abbreviation dissambiguation, medical term suggestions
3. *A score is provided to the previous and corrected version of the report
4. The corrected report, alongside with the original report, is stored in the case base, as well as appended to the revision list

1. When the amount of scored cases reaches a certain value (i.e: 30-50), a regression model is trained using the indicators of each case as the attributes, and the score as the goal value.
2. The model should be periodically retrained to be up to date with the newest cases

1. An expert manually validates each of the newly added cases, modifying them is needed
2. The expert assigns a score from 0 to 10 to the revised report, evaluating its overall quality

Figure 4 Overview of the CBR Cycle

3.1 Overview of the CBR Cycle for Report Correction

As shown in the diagram, this module implements a Case-Based Reasoning model to aid on the generation of radiology reports. These systems comprise four differentiated steps, namely: Retrieval, Reuse, Revise, Retrain. Prior to defining the cycle, it is necessary to define what comprises a case. The following figure illustrates the format and the content of each case:

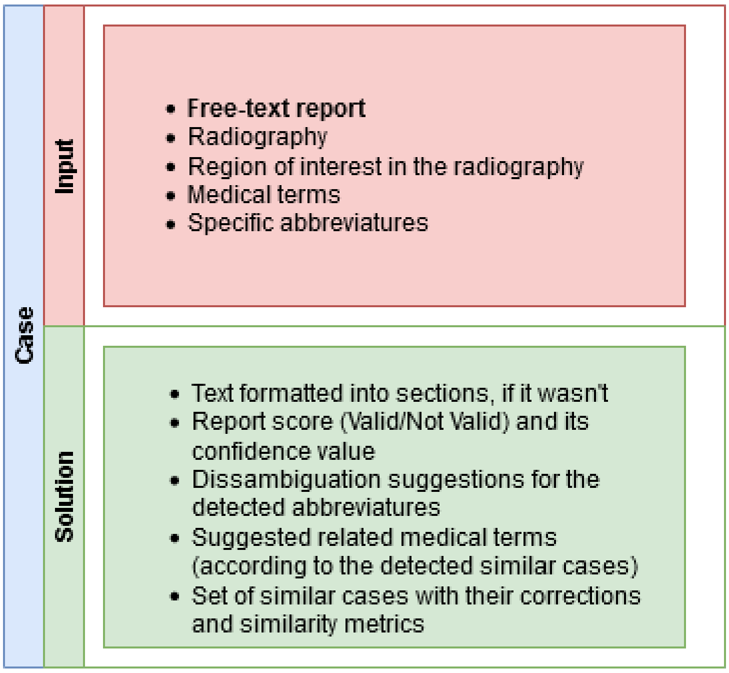

Case
Input
Solution
• Free-text report
• Radiography
• Region of interest in the radiography
• Medical terms
• Specific abbreviatures
• Text formatted into sections, if it wasn't
• Report score (Valid/Not Valid) and its confidence value
• Dissambiguation suggestions for the detected abbreviatures
• Suggested related medical terms (according to the detected similar cases)
• Set of similar cases with their corrections and similarity metrics

Figure 5 Case Summary

As highlighted in bold, the only required element to define a problem is a report, either sectioned or non-sectioned. Additionally, relevant information such as radiographies, indicating their regions of interest, glossaries of abbreviatures and set of detected medical terms can be provided. Though this information is not mandatory, it can improve the generated solution, as well as improving the results of the retrieval process.

Given the definition of a case, the designed workflow is as follows:

3.1.1 Retrieval

The input of the system will be a radiology report, composed by text of free format and, optionally, a radiography. In terms of a CBR Cycle, this would be a **new case**. Before initiating the retrieving process, the system should separate image data from textual data, as they require different treatments. Textual data can come in different file formats, so before any operation is performed, the system should extract the clean textual data from the input.

Once the text has been extracted, and separated from the images, the retrieval stage begins. One of the key elements of the retrieval phase is defining ***what makes a case similar to another***. Similarity can be measured according to several criteria, which varies across the different domains. In the medical domain, however, there is usually an expert knowledge required to set these criteria, which the developer usually does not have. Thus, to tackle this issue, four indicators are considered to retrieve similar cases, leaving to the expert user the establishment of the similarity criteria.

These indicators are:

- *Reports containing similar radiographies*: As previously mentioned, most works beneath the radiology area focus on the development of image diagnosis systems. Thus, as images in this context provide such powerful information, their importance should also be extended to our problem. So, given a report containing a radiography, two possibilities could be considered to retrieve similar images. The first possibility is based on *supervised learning*: given a model trained to classify radiographies into categories, one could just simply predict over a new radiology, get its label and then retrieve all the existing elements labelled with the same value. This approach, however, is expensive in terms of computational power, as well as quite complex in terms of data heterogeneity (not all radiology image datasets are labelled with the same values). Thus, the second approach relies on *unsupervised learning*: if there is a set of feature vectors, each associated with a case, the feature vector of the new image can be compared with those existing, using vector operations. The second approach not only is less computationally expensive but can deal with data heterogeneity. As we are also considering Regions of Interest (or ROIs), the user can also decide whether to compare the full image with the existing ones, or only the region of interest.

- *Document embedding:* Similar to the case of images, textual data can also be embedded in fixed-size vectors, thus enabling comparison between texts via vector comparison. Using the text embedding module, vectorial representations for each report can be generated, then compared using vector operations such as pairwise cosine distance or Euclidean distance.
- *Named Entities Detected:* Named-Entity Recognition is one of the key NLP tasks, being particularly important in the medical domain. As mentioned previously, CliNER is used in this framework to implement this functionality. After performing NER over a new report, a set of named entities is obtained. Then, the system could retrieve all the cases containing one or several of the identified entities.
- *Identified Abbreviations:* A key issue within medical reports is the overuse of abbreviations. These abbreviations generally refer to medical terms that have been contracted for its easier typing, however, sometimes they do not have a clear meaning correspondence, therefore introducing noise. Thus, the degree of correctly identified abbreviations could also be considered a discriminative retrieval metric.

Given these four indicators, identified in Figure 6 as I1, I2, I3 and I4, the user can provide a threshold for each, filtering the retrieved results. These indicators can be combined either in a conjunctive or disjunctive way. All this criteria is then used to query the existing cases, returning only the top N (or all) cases meeting the user-provided constraints. The following is an example of a valid retrieval criteria:

```
Criteria = {I1>0.8, I2>0.7, I3=['Pulmonary Disease']
                ,I4>0.6,N=5,op='OR'}
```

This query means that the system will retrieve the top 5 reports that either:

- Contain images that are at least 80% similar to the given radiology
- Contain text that is similar in at least a 70%
- Contain Pulmonary Diseases
- Have at least a 60% of used abbreviations identified.

As the considered indicators relate to data of different format and nature, the following similarity measures are considered for each of them:

- **I1 (Image comparison):** Regarding image feature extraction, we can distinguish two main categories: black-box feature extraction models (convolutional neural networks) or white-box feature extraction models. Though white-box model results are explainable, which is a highly desirable feature; they are prone to error, as well as sensitive to image noise. Thus, in this framework a hybrid feature extraction approach, using both black and white box models is used. Once the extraction procedure finishes, the image is represented by a vector. Vectors support several distance and similarity measures. To obtain a similarity value that is normalized, the cosine similarity is used. Thus, given two image vectors Iv_1 and Iv_2, the similarity between the images would be $cos(Iv_1,Iv_2)$.
- **I2 (Document comparison):** Like images, documents can also be embedded and represented as vectors. Opposite to the case of images, where a hybrid approach was taken, a deep learning model is used for document embedding. This decision is supported by the fact that deep text embedding models can capture underlying semantic information that leads to better vector representations. Document embeddings are compared to obtain a similarity value using the cosine distance, as in the case of images.
- **I3 (Named Entities Detected):** This indicator serves as an additional filter for the case retrieval. As this metric consists of a list of words, a naïve way to measure the similarity would be to simply count the coincidences between lists. However, as this can be a very restrictive metric, as if no named entities are detected no cases would be retrieved. To tackle this potential issue, the user is the one who marks which of the detected entities, if there are, are considered for comparison. If no entities are detected in the input report, the user can select one or more entities from the recognized set of named entities. Thus, only those cases containing entities marked by the user are retrieved.
- **I4 (Identified abbreviatures):** As previously stated, this indicator can be considered as a noise filter. Thus, it can be used to expand or restrict the search scope, by considering or discarding as potential cases those containing a certain percentage of unidentified abbreviatures.

One of the key challenges of the system is to successfully implement a robust, but efficient retrieval system. Different storage options can be considered at this stage, such as relational databases, or tree structures, such as kd-Trees. Though kd-Trees can be useful when dealing with purely numerical attributes, in this context where the input data is presented in a symbolic format, it is not easily applicable. To keep the application as lightweight as possible, the cases will be stored externally to the framework in a SFTP Server. However, an index CSV file will be stored locally, containing the indicator values of each stored case, alongside with their location in the server. If this file already exists in the SFTP Server, it will be copied locally and synchronized when changes are performed. This way, the retrieval is performed efficiently, as only a single file is stored locally. Figure 6 presents the overview of the retrieval procedure:

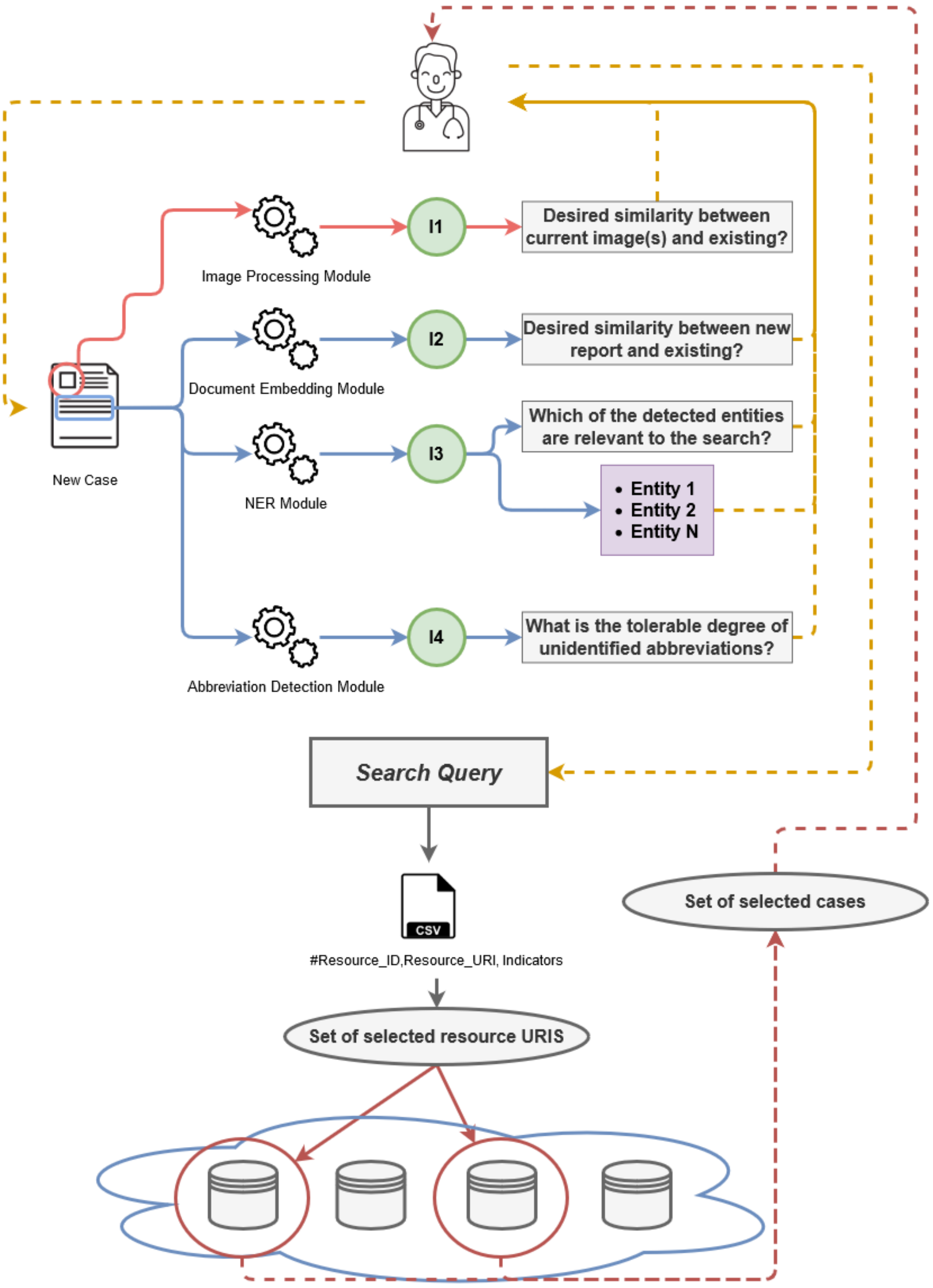

Desired similarity between current image(s) and existing?
Image Processing Module
I1
Desired similarity between new report and existing?
Document Embedding Module
I2
Which of the detected entities are relevant to the search?
NER Module
I3
• Entity 1
• Entity 2
• Entity N
New Case
What is the tolerable degree of unidentified abbreviations?
Abbreviation Detection Module
I4
Search Query
CSV
#Resource_ID,Resource_URI, Indicators
Set of selected resource URIS
Set of selected cases

Figure 6 Retrieval Procedure

3.1.2 Reuse

Once the user sets the retrieval criteria, the existing cases that fit the constraints, presented to the user. For each retrieved case, a brief indicator on **why** it has been returned is provided. I.E 'this report contains a radiology that has a 95% similarity with your provided data, as well as containing the entity "Pulmonary Disease"'. This way, the user is informed on the retrieval decisions taken by the system and can consider them accordingly.

Aside from this content corrections, that should be implemented by an expert user given the considered domain, several format correction suggestions are provided by the module, being:

- A validation score for the current report, indicating whether it is valid or not with a certain confidence value.
- A potential section division of the text, into the four general sections a radiology report should have: *Indication, Comparison, Findings* and *Impressions*.
- Suggestions for correcting the unidentified abbreviations.
- A list of medical terms that are related with the ones detected in the report, in case some additional information needs to be provided.

All the provided suggestions should be validated by the user. Once the user has implemented the pertinent changes on the original report, this new case *(problem, correction)* tuple is appended onto the revision list, as well as including this case into the case base under the flag “Pending on validation”.

3.1.3 Revision

Once several new cases with their corresponding solutions have been appended onto the revision list, an expert manually revises all these new cases, indicating which ones are valid, and which are not valid. If some of the unrevised new cases, which at this stage would have been included in the case base, have imprecise or wrong corrections, the expert can modify these cases and store the modifications in the case base.

3.1.4 Retain

When a considerable number of cases has been scored, a classification model is trained using the indicators of each report as attributes and the given score (valid or rejected) as label. Once this model reaches a maturity (it has been trained over a sample big enough and showing good metrics), this model can substitute the figure of the expert, which will not always be available, as well as serving as a referee to decide whether a new case should be included, or not, in the case base.

In order to keep this model up to date, regular retrainings would be programmed.

4 System and Data Requirements

r.AID.ologist comes as a Docker image, with a total size of 3.2GB. This image contains not only the code of the framework, but a set of sample cases generated from the Open-I dataset. Port 5000 of the Docker image is open to receive requests, though this port can be mapped to the preferred local port using the **-p** flag when running the Docker image. The framework has been successfully run under a machine using 8GB RAM and a GPU capability of 2GB. The system contains several machine learning techniques, which can work either in GPU or CPU. We recommend, whenever possible, to use GPU computation, as it significantly reduces processing time.

After reviewing the existing open radiology datasets, two different format profiles were defined according to the representation of text reports: plain format and structured format. General guidelines and constraints about each of these profiles were defined to properly define parsing and handling mechanisms for each. The developed parsers can then be used to generate cases from heterogeneous data to populate the initial case base. A sample of case generated from each format is provided in the attached file Appendix 1. This file contains both an XML and a plain text file with their corresponding images (in PNG and DICOM format) parsed into cases.

All machine learning models contained within the system read directly from the case set, as to minimize the interactions between the original data and the system. Aside from the format, two main constraints are also imposed at this stage: system models are trained to function only over **English-written data** and only **four sections are considered for the reports (Indication, Comparison, Findings and Impression). Sections need to be uppercased.** Regarding image, the following formats are currently supported by the framework: DICOM, PNG, JPG and GIF.

However, this does not apply to the Named Entity Recognition module, as it is the only element that has been integrated into the system from an external source. The employed Named Entity Recognition pretrained model (CliNER) has been trained over the MIMIC-II dataset. Thus, if the current model needs to be replaced by a new one, two options can be considered: retrain CliNER with new data to generate a new NER model (following the data constraints imposed by their developers) or using a different algorithm to generate a suitable NER model. Whichever option taken, the only constraint is that **the new model needs to recognize the same entity types as the original model, namely: problem, treatment, and test.**

The two data formats considered for automatic case generation are defined as follows:

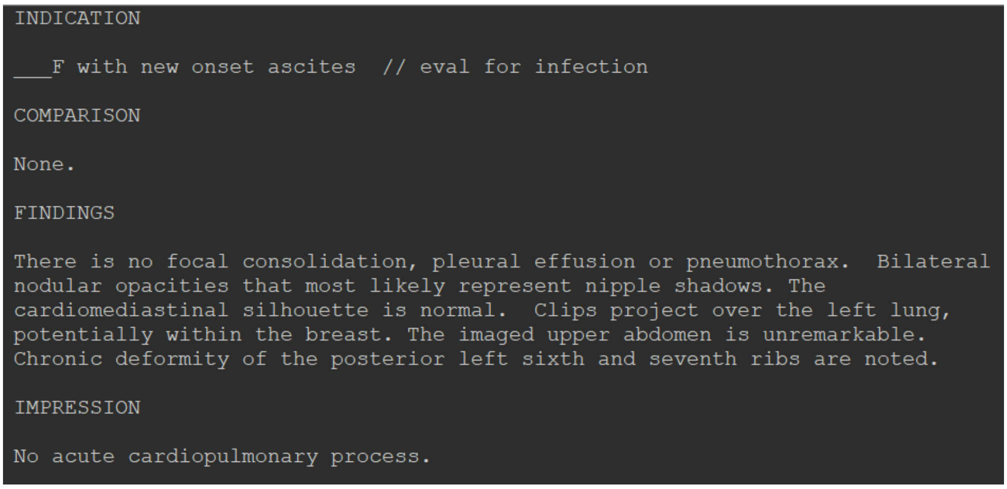
4.1 Plain data

This format will be used when the report comes as a single textual file i.e: a txt file. In this format, report files only contain text, either sectioned or non-sectioned, but no further data is provided i.e.: no specified image files, no specified abbreviations, no specified terms. Therefore, little information is contained within these reports. Nonetheless, as their use seems to be quite spread, they are also considered. In order to find the radiography image file associated to the report, a specific naming criterion is used. To identify which report is associated to which file, the same unique identifying prefix must be used for each. This prefix must be composed only by alphanumerical characters and must be ended by an underscore. After the underscore, the name of the file is specified, which can contain any character except for underscores and slashes. Finally, reports and images must be separated into homonymous folders, as shown below.

```
      MIMIC_CXR_FILES
      |____reports
           |___mcxr1_report
           |___mcxr2_report
      |____images
           |___mcxr1_radiography1
           |___mcxr1_radiography2
           |___mcxr2_radiography1
```

In this example, two pairs of (*report, images*) is presented, identified by the prefix *mcxr* followed by a number, which in this case are 1 and 2. Thus, the first case will be created from the report file *mcxr1_report*, and its corresponding image files *mcxr1_radiography1* and *mcxr1_radiography2*, while second case will be created from the report file *mcxr2_report* and its corresponding image file *mcxr2_radiography1*.

Reports can be presented either sectioned or non-sectioned. If the reports are non-sectioned, a sectioned version will be inferred using the pretrained sectioning model. However, as this solution has not been directly introduced by the user, the generated case will be labelled as ‘Pending on validation’. Otherwise, if the report comes sectioned, a corrupted non-sectioned version will be inferred from the report by removing the headers and other sectioning elements such as new lines or carriage returns. The original, sectioned report will be considered as the solution of the corrupted version. As this solution has been specified directly by the user, the generated case will be considered as Valid. Below there is an example of a correct, sectioned report in plain text format:



```
INDICATION

___F with new onset ascites  // eval for infection

COMPARISON

None.

FINDINGS

There is no focal consolidation, pleural effusion or pneumothorax.  Bilateral
nodular opacities that most likely represent nipple shadows. The
cardiomediastinal silhouette is normal.  Clips project over the left lung,
potentially within the breast. The imaged upper abdomen is unremarkable.
Chronic deformity of the posterior left sixth and seventh ribs are noted.

IMPRESSION

No acute cardiopulmonary process.
```

4.2 Structured data

Aside from plain text radiography reports, different structured, richer data formats were also detected amongst the studied existing sets. These formats included tabular data, such as CSV files, and non-tabular data, such as XML or JSON files. While there is a considerable heterogeneity between structured data sources, most of them can be easily translated from one

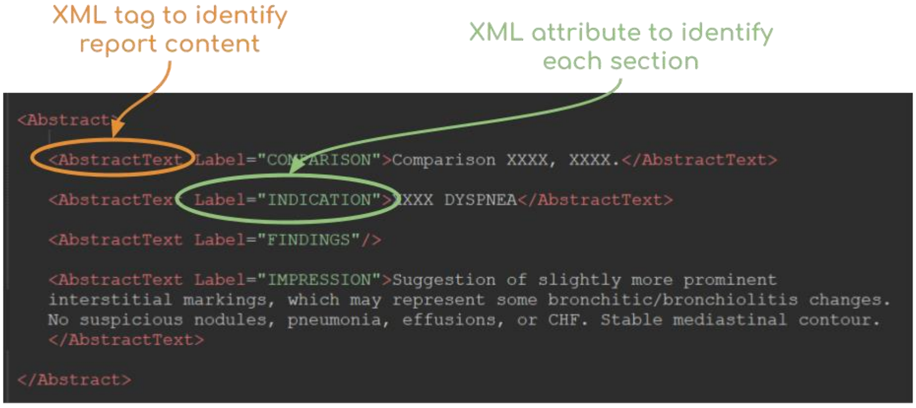
:i.e: JSON to XML, RDF to XML, XML to CSV, etc. Thus, considering only one structured data format may be enough. XML was selected as structured data format for text reports.

For XML files, no naming constraints are given, as all relevant information can be contained within the document. In this case, aside from the report itself, several additional information can be provided using XML tags, such as detected terms or abbreviatures, as well as indicating which are the associated image files. However, as in the previous case, reports and images must be separated into two folders: ‘reports’ and ‘images’. Thus, to indicate the image file, only its relative name with respect to the ‘images’ folder needs to be specified, instead of its absolute path.

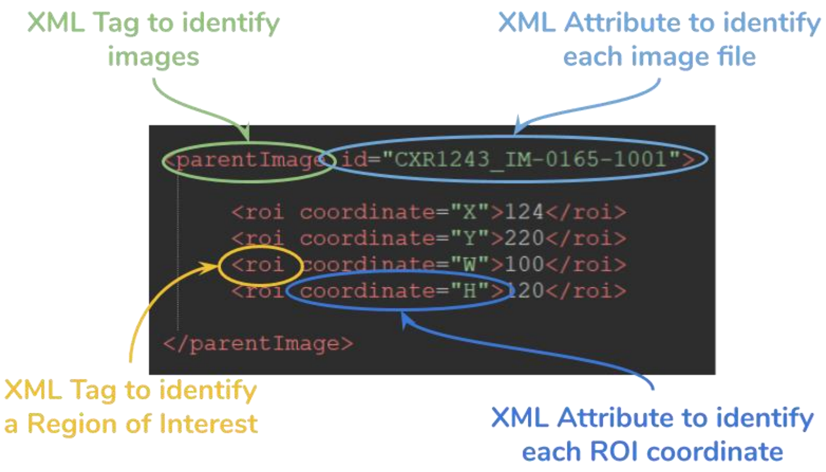
As to properly parse all the information contained in the XML file, the following criteria needs to be clearly defined prior to creating the data, which must be equal across all files:

- **XML Tag that identifies the report content.**
- XML Attribute that identifies each section. Default value is ‘Label’.
- XML Tag that identifies the associated radiography files (if there are).
- XML Attribute that identifies each associated image file. Default value is ‘id’.
- XML Tag that identifies the region of interest of each image.
- XML Attribute that identifies the coordinates of the region of interest. Default value is ‘coordinate’.
- XML Tag that identifies report terms.
- XML Tag that identifies abbreviations.

The following images show snippets of a right-formatted sample XML file, indicating which tags and attributes were used for each criterion specified above:

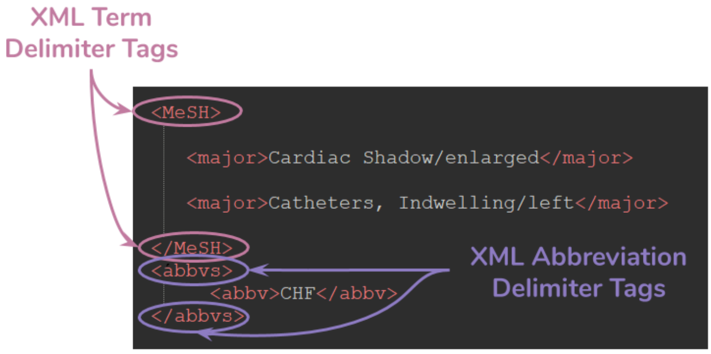


```
<Abstract>

  <AbstractText Label="COMPARISON">Comparison XXXX, XXXX.</AbstractText>

  <AbstractText Label="INDICATION">XXXX DYSPNEA</AbstractText>

  <AbstractText Label="FINDINGS"/>

  <AbstractText Label="IMPRESSION">Suggestion of slightly more prominent
  interstitial markings, which may represent some bronchitic/bronchiolitis changes.
  No suspicious nodules, pneumonia, effusions, or CHF. Stable mediastinal contour.
  </AbstractText>

</Abstract>
```



```
<parentImage id="CXR1243_IM-0165-1001">
    <roi coordinate="X">124</roi>
    <roi coordinate="Y">220</roi>
    <roi coordinate="W">100</roi>
    <roi coordinate="H">120</roi>
</parentImage>
```

XML Tag to identify images
XML Attribute to identify each image file
XML Tag to identify a Region of Interest
XML Attribute to identify each ROI coordinate



```
<MeSH>
    <major>Cardiac Shadow/enlarged</major>
    <major>Catheters, Indwelling/left</major>
</MeSH>
<abbvs>
    <abbv>CHF</abbv>
</abbvs>
```

Regarding tags and attributes, the following dependencies need to be considered:

- The XML tag (as well as the content) of the report must appear in all report files. The XML Attribute to indicate sections is not mandatory, though recommendable.
- If the report has a radiography file associated, both the XML tag and attribute are required to identify each image file.
- If the radiography has a region of interest, both XML tag and attributes to identify the ROI need to be specified. ROI coordinates need to appear in the sequence X,Y,W,H (and with that same naming), being X and Y the coordinates of the upper left corner of the region, W the width and H the height.

4.3 References and links

4.3.1 Datasets

MIMIC-CXR

ChestXray-NIHCC

CheXPert

Stanford's Langlotzlab

PadChest

Open-I

4.3.2 Resources

CliNER

SciSpacy

OpenCV

Scikit

Pytorch