Extraction and normalization of simple and structured entities in medical documents

Perceval Wajsbürt

THESE DE DOCTORAT DE
SORBONNE UNIVERSITE

Spécialité « Science des données »

ECOLE DOCTORALE PIERRE LOUIS DE SANTE PUBLIQUE A PARIS :
EPIDEMIOLOGIE ET SCIENCES DE L'INFORMATION BIOMEDICALE

Présentée par

Perceval Wajsbürt

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITE

Sujet de la thèse :

Extraction et normalisation d'entités simples et structurées dans les documents médicaux

(Extraction and normalization of simple and structured entities in medical documents)

soutenue le 14/12/2021

devant le jury composé de :

Directeur de thèse	Xavier Tannier, *Sorbonne Université, LIMICS*
Co-encadrante de thèse	Christel Daniel, *Assistance Publiques des Hôpitaux de Paris, LIMICS*
Rapporteurs	Vincent Claveau, *CNRS, IRISA*
	Tim Miller, *Harvard University, Boston Children's Hospital*
Examinatrices	Sandra Bringay, *Université de Montpellier, LIRMM*
	Anita Burgun, *Hôpital Européen Georges Pompidou, Centre de Recherche des Cordeliers*
Membre invité	Dongfang Xu, Harvard University, Boston Children's Hospital

Sorbonne Université
Bureau d'accueil, inscription des doctorants et base de données
Esc G, 2ème étage
15 rue de l'école de médecine
75270-PARIS CEDEX 06

Tél. Secrétariat : 01 42 34 68 35
Fax : 01 42 34 68 40
Tél. pour les étudiants de A à EL : 01 42 34 69 54
Tél. pour les étudiants de EM à MON : 01 42 34 68 41
Tél. pour les étudiants de MOO à Z : 01 42 34 68 51
E-mail : scolarite.doctorat@upmc.fr

Abstract

Hospital clinical documents are rich sources of information for various applications such as patient recruitment for clinical research, epidemiological surveillance, medical coding, and decision support tools. However, being primarily written in natural language, these documents are not easily amenable to large-scale computer processing and must first be structured. We aim to extract entities mentioned in these documents, whether simple or structured, i.e., containing several labels or parts, and normalize them with concept bases. We contribute to several natural language processing (NLP) tasks, namely named entity recognition (NER), medical entity normalization, and structured entity extraction. In particular, we investigate training deep learning models in low data settings, for languages other than English and in the clinical domain. We structure our approach in three steps: tag, normalize, and compose. We first propose two methods to tag simple entities, especially when they overlap in texts. We then develop a large-scale multilingual model to normalize them in several languages. Finally, to compose simple entities into structured entities, we propose a new method based on mention cliques and scope relations. We evaluate it to a new annotated dataset of breast imaging reports.

Résumé

Les documents cliniques hospitaliers constituent de riches sources d'information pour diverses applications telles que le recrutement de patients pour la recherche clinique, la surveillance épidémiologique, le codage médical et les outils d'aide à la décision. Cependant, étant essentiellement rédigés en langue naturelle, ces documents ne se prêtent pas aisément à des traitements informatiques à grande échelle et doivent d'abord être structurés. Nous visons à extraire les entités mentionnées dans ces documents, qu'elles soient simples ou structurées, c'est-à-dire contenant plusieurs étiquettes ou parties, et à les normaliser selon des bases de concepts. Nous contribuons à plusieurs tâches de traitement du langage naturel (TAL), à savoir la reconnaissance des entités nommées, la normalisation des entités médicales et l'extraction d'entités structurées. Nous nous intéressons notamment à l'entraînement de modèles par apprentissage profond (deep learning) dans des conditions de données limitées, pour des langues autres que l'anglais et dans le domaine clinique. Nous structurons notre approche en trois étapes : surligner, normaliser et composer. Nous proposons d'abord plusieurs méthodes pour surligner des entités simples, notamment lorsqu'elles se chevauchent dans les textes. Nous développons ensuite une approche multilingue à grande échelle pour les normaliser dans plusieurs langues. Enfin, pour composer ces entités simples en entités structurées, nous proposons une nouvelle méthode basée sur les cliques de mentions et les relations de portée. Nous l'évaluons sur un nouveau corpus annoté de comptes rendus cliniques de mammographies.

Remerciements

Je voudrais remercier toutes les personnes qui m'ont accompagné et soutenu durant ces trois années de thèse et sans le soutien de qui, l'aboutissement de cette aventure n'aurait pas été possible.

Je remercie tout d'abord infiniment mes encadrants de thèse, Christel et Xavier pour leur soutien tout au long de ce parcours. Merci d'avoir accepté de m'encadrer et de m'avoir fait confiance pour mener ces travaux. Ce sont tous ces échanges, aussi riches que réguliers, ces conseils et cette disponibilité qui m'ont permis de réaliser ce travail.

Je remercie les membres de mon jury de thèse. Merci à Tim Miller et à Vincent Claveau d'avoir accepté d'être rapporteurs de ma thèse. Merci également à Sandra Bringay et Anita Burgun d'avoir accepté de l'évaluer.

Merci à tous mes collègues pour les échanges passionnants et réguliers que nous avons eus. Un merci tout particulier à Jacques, Yoann, Chyrine et Jean pour leur enthousiame, leurs conseils et leur disponibilité.

Merci également à Yoann, Isabelle, Antoine, Nesrine, Christel et Ali avec qui j'ai pu collaborer de façon plus étroite dans le cadre de publications et grâce à qui j'ai pu faire avancer ma recherche.

Je souhaite également remercier mes amies et mes amis, pour leur soutien indéfectible et leurs encouragements constants.

Merci également à Isabelle, Marie-Christine, Elise et à tous les autres d'avoir fait tourner le laboratoire et de m'avoir ainsi assuré un cadre de travail idéal.

Merci aux équipes de l'Entrepôt de Données de Santé (EDS) de l'APHP. C'est grâce à eux que j'ai pu tester, développer et évaluer mes méthodes et les travaux présentés dans ce manuscrit.

Merci également à l'Institut des Sciences du Calcul et des Données (ISCD) de m'avoir financé pendant ces trois ans.

Merci à ma famille de m'avoir soutenu et conseillé durant toutes ces années d'études.

Enfin et surtout, merci à Margaux, de m'avoir accompagné au cours de ces trois ans, d'avoir été une oreille attentive à mes doutes et questionnements, et de m'avoir soutenu dans les moments les plus difficiles.

Contents

List of Figures

List of Tables

Glossary

ACR American College of Radiology. ACR is commonly used to refer to the BI-RADS grading system like "ACR 1" meaning "Grade 1 in the BI-RADS system of the ACR". 25

APHP Assistance Publique des Hopitaux de Paris (Paris public Hospitals). 3, 76

Attention The attention mechanism is a pooling mechanism that acts on sets of elements, where every element has a key and a value. A query is used to build an attention score for each element by computing a similarity score with its key, and using that score to weight the value of the element in the final pooled result. 88–91, 97, 100, 101, 107, 108

Autoregressive model An autoregressive model predicts a given output at a given step using past predictions from the same model as inputs. 35, 43, 45, 46

Batch A subset of samples that is used to compute gradients to optimize the parameters of a model. 36, 60, 95, 96

Batch normalization A process used to normalize the input or output of the activation functions inside a neural network. 60

BERT Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique for natural language processing pre-training. 9, 11, 22, 37, 71, 85

BIRADS Breast Imaging Reporting and Database System score. It's a scoring system radiologists use to describe mammogram results. 25, 29, 77

Breast quadrant A single breast can be divided into four quadrants: UO, upper inner (UI), lower outer (LO), and lower inner (LI) by two perpendicular planes intersected at the nipple. 25, 27, 29, 95

Clique A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are connected by an edge. The clique is maximal if no new vertice can be added to the clique without it not being a clique anymore. 5, 88, 103

CLS Classification token used in BERT to represent to full text sample. 60

Comité Scientifique et Éthique Comité Scientifique et Éthique (CSE): Scientific and Ethics Comity. 3, 76

Concept Unique Identifier Concept Unique Identifier (CUI) are identifiers used in the UMLS lexicon for concepts. 8, 20, 23, 57–59

Conditional Random Fields Conditional Random Fields (CRFs) are a class of discriminative probabilistic models that encode conditional dependencies between variables, like word labels in a sentence, by exploiting local neighbourhood information. 13–15, 38–43, 91, 93, 94, 100

Convolutional Neural Network A Convolutional Neural Network (CNN) is a class of neural networks that process the input (image or text) with internal transformations on small sliding windows. 9–11, 13, 22, 37, 38, 44, 46

Coreference A coreference occurs when two or more expressions in a text refer to the same person, thing or event. 76, 81, 82, 95, 103

Cosine similarity The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is defined to equal the cosine of the angle between them, which is also the same as the inner product of the same vectors normalized to both have length 1. 60, 64

Cross-entropy A quantification of the difference between to probability distributions (usually a computed one and a target distribution), often as an objective to minimize to train a neural network. 22, 42

Data augmentation A process to artificially boost the number of training samples by producing many variants of a same sample (such a splitting a document in multiple samples, replacing some words, etc). 63, 94, 95, 101

Decoder A part of a neural network that converts internal representations to the desired output. viii, 16, 35, 38, 39, 41, 42, 44–46, 85–87, 94

Distant supervision Distant supervision is a learning scheme in which a model is learned given a weakly labeled training set (training data is labeled automatically based on heuristics / rules). By contrast, conventional supervised learning uses training samples from a gold-standard dataset. 63, 65, 66

Dropout A technique to disable a subset of hidden connections a each step in order to prevent overfitting. 64

Embedding A categorical feature represented as a set of continuous-valued features (a point in a high-dimensional space). 9–11, 13, 14, 22, 35–38, 42, 44–46, 51–53, 56, 59–65, 71–73, 85, 87, 93

Encoder A part of a neural network that converts a input sample to internal representations. 36, 44, 59, 60, 62, 63, 70

Ensemble A model that merges the output predictions of multiple models. 35, 43–47, 50, 104

Epoch A full training pass over the entire dataset. 44, 64, 66

F1 F1 measure, the harmonic mean between Precision and Recall. 17, 24, 44, 63, 109

Fine-tuning The process of optimizing an already optimized model to adjust its weights to fit a new task. 36, 37, 44–50, 52, 53, 60, 62, 71

Freezing Freezing is the process of preventing the weights of a neural network layer from being modified during the backward pass of training: frozen weights cannot be modified. 37, 44, 45, 53, 62, 70

Gating A gating mechanism allow a neural network to combine different embeddings to iteratively build a representation, instead of computing it from scratch. The simplest gating mechanism is the residual mechanism that adds an "update" embedding to the last produced embedding in a multi-layer network. 86, 97, 100

Gazeteer A gazetteer consists of a set of lists containing names of entities such as cities, organisations, days of the week, etc. These lists are used to find occurrences of these names in text, e.g. for the task of named entity recognition. 12

GPU Graphical Processing Unit, a hardware accelator that is intensively used for deep-learning. 64

Heuristic A simple and quickly implemented solution to a problem or a sub-problem. 12, 93, 100

Hyperparameter A value that is not directly optimized by gradient optimization, but instead chosen manually or automatically changing its value over multiple experimennts. 44, 60, 64, 96

Inductive bias The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs of given inputs that it has not encountered. For example, a recurrent neural network assumes that modelling texts as sequences is beneficial to the target task, whereas a Transformer assumes that modelling them as a set is better. 101, 107

Interpretability The ability to explain or to present an ML model's reasoning in understandable terms to a human. 12, 27

Logits Non normalized scores assigned to each class of a classification problem. Applying a softmax function to a set of logits transforms it into a probability distribution. 43

Loss A measure of the error that the model makes that can be optimized. It is usually diffentiable and optimizable by gradient descent. 22, 40, 42, 94, 95

LSTM Long Short Term Memory networks: improved recurrent neural network, introduced by Hochreiter and Schmidhuber (1997). 10, 13, 14, 22, 36, 38, 85, 86, 101

Machine Translation Machine Translation (MT) is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one language to another. 65

Negative A process to only use a subset of the incorrect possible classification candidates to optimize a classification model, instead of all the negative samples. 61

NER Named Entity Recognition, an standard NLP task that aims a automatically highlight spans of text. 2, 4, 9, 11–17, 23, 24, 30, 33, 35, 36, 56, 83, 87, 94, 104, 106, 109, 113

NLP Natural Langage Processing. 1–5, 7–9, 14, 20, 29, 30, 113

PDF Portable Document Format. 3

Pooling Reducing a set of values to a single one, for example by taking the minimum, the maximum or the mean. 46, 53

Precision The fraction of correctly retrieved items over all the predictions items. 17, 24, 44, 63, 66, 70

Recall The fraction of correctly retrieved items over all the target items. 17, 24, 44, 63, 70

Scope Annotations of text zones on which a named entity referred to as a "cue" applies its meaning. ix, 5, 76, 90–94, 97, 99, 100, 103

SEP Separation token used in BERT to represent the end of a text sample part. 60

Softmax A function that rescale a set of scores to obtain a probability distribution. 45, 60–62, 65

Synset A synset is the set of synonyms that share a same concept. 59, 63

Transformer A transformer is a deep learning model that adopts the mechanism of self-attention to iteratively refine its internal representations of the input (a sequence of text tokens in NLP). 22, 37, 44, 46, 59, 62, 64, 70, 85

True positive A correct match between predicted item and a target item. 17, 109

UMLS The Unified Medical Language System (UMLS) is a compendium of many controlled vocabularies in the biomedical sciences. 3, 8, 19, 20, 23, 24, 56, 57, 63–66, 70–72

Wordpiece WordPiece is a subword-based tokenization algorithm. The algorithm split sentences into lists of tokens from a fixed-size vocabulary, where a word might be split into multiple subwords or "wordpieces". 36, 37, 45, 46, 53, 59, 64, 71, 72, 85

Chapter 1

Introduction

Hospital clinical documents (e.g., hospitalization or consultation reports, nursing transmissions, discharge letters and prescriptions, or physicians' letters) constitute rich sources of information for various applications such as patient recruitment for clinical research, epidemiological surveillance, medical coding, and decision support tools (Wang et al., 2018c). These documents are primarily written in natural language, which helps to ensure completeness and accuracy of the information, accommodate special cases, and facilitate data entry. Indeed, it is estimated that more than 80% of hospital data are collected in the form of texts (Raghavan et al., 2014). Unfortunately, the free text format is not easily amenable to the use of standard computer processing programs. In contrast, structured representations increase the quality and reuse of patient data for clinical care (including decision support), clinical audit and research, medical coding for resource allocation, and health service planning. In health care facilities, efforts have been made to replace manual reports with forms that ensure structured representations. However, the descriptive needs of clinicians change over time, and it has been shown that the "additional remarks" fields tend to contain more and more information, reflecting a lack of flexibility in the forms (Steichen et al., 2007). Another approach, which is the one we are interested in in this thesis, is the automatic structuring of text documents. One of its main advantages is the possibility to modify the algorithm a posteriori without disrupting the activity of hospital practitioners. This discipline, commonly referred to as information extraction (IE) in natural language processing (NLP), encompasses many research areas.

Structuring Structuring is the process of transforming a free text sample into an organized view of the information it contains. The sample text can be a single sentence, a paragraph, an entire report, or even a patient record containing multiple reports. These structured representations can take different forms, as illustrated in Figure 1.1. In the case of classification, we can assign each sample a unique label from a predefined list, such as the type of report or the gender of a patient, or a yes/no answer to a question. Multi-label classification allows samples to be classified with multiple labels, such as the report type and a cancer risk score

if it is a mammogram. Another kind of structure focuses on the notion of entities. Entity recognition aims at extracting a variable number of objects from the text sample, such as the observed lesions in a radiology report. The different entities are usually mentioned explicitly in the text by a keyword or a keyphrase but can also be composed of several parts or be implicit. As in classification tasks, each entity can be characterized by one or more labels. Entity extraction has been studied for several decades and many solutions have been proposed. The well-known Named Entity Recognition (NER) task (① in Figure 1.1) corresponds to the extraction, or tagging, of simple entity mentions with a beginning, an end and a single label. Yet, the task of extracting overlapping mentions in documents is still under active research. Furthermore, the extraction of more exotic entities containing multiple labels and/or parts (③ in Figure 1.1) is still far from being solved, despite the relevance of these entities in areas such as clinical information extraction. In this work, we will refer to entities characterized by multiple labels or parts as structured entities, as opposed to the classic simple named entities. The labels themselves can be defined specifically for the task at hand or drawn from existing databases of medical concepts. The process of mapping entities to these concepts is known as normalization (② in Figure 1.1). These databases have been built over time by the biomedical informatics community and are rich in information: ontologies provide relations between concepts, and terminologies provide synonyms to define these concepts and identify them in text. In addition, their use promotes interoperability between upstream and downstream systems through concept standardization.

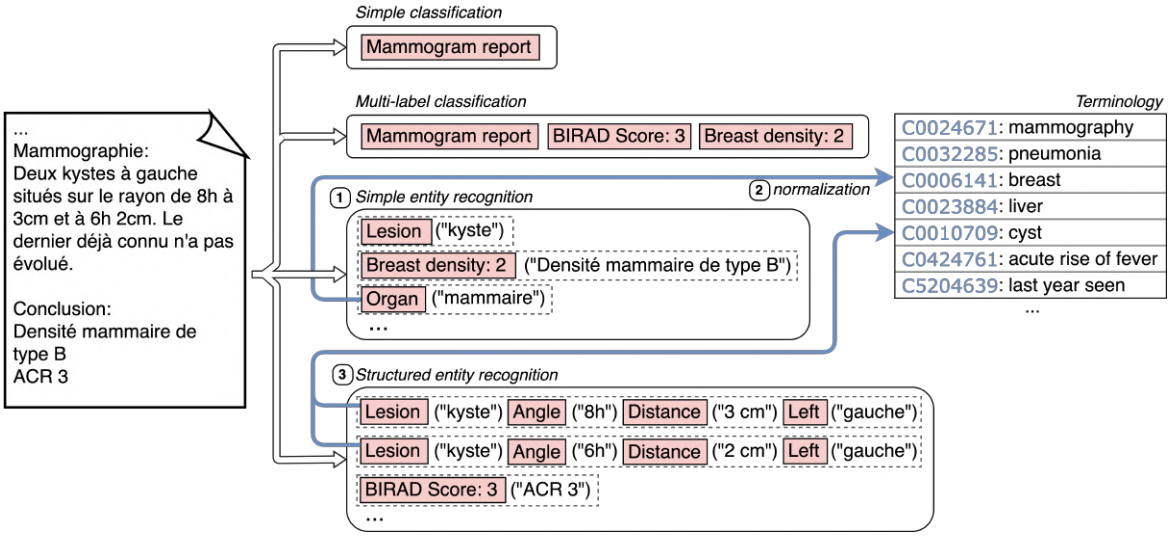


Figure 1.1 Overview of different structuration objectives, with concept normalization

Supervision challenges Over the past few decades, the need for medical document analysis, coupled with the rapid growth of health data warehouses and the increasing number of biomedical scientific publications, has led to the development of NLP approaches in the general

and biomedical domains. The advent of machine learning, especially deep learning, has come with the promise of describing a task with examples from which to generalize rather than building hand-crafted domain- and language-specific rules. These methods have gained an immense popularity and demonstrated their superiority in a wide range of domains. However, with the possibilities of these methods has come a ravenous appetite for annotated data: many modern learning methods fall into the category of fully supervised learning, i.e. they require the creation of an annotated dataset (by human experts) to allow the training of a model that can then be applied on new data. The time cost of annotating documents and the high annotation requirements of deep learning approaches represent a barrier to automating information extraction. However, in many cases, there exist auxiliary medical knowledge resources, such as terminologies, that are not in the form of annotated examples. Efficiently injecting this knowledge into learning models is still under active research. The annotation process itself is also far from trivial, as designing a scheme that reconciles simplicity, expressiveness and consistency is a challenge in itself.

French clinical language processing The difficulties related NLP are numerous. Indeed, natural language is subject to semantic and syntactic ambiguities. As any written document, a clinical report may contain spelling mistakes, grammatical errors, or even contradictions. In addition, the computerization of these reports and their conversion from and to PDF can introduce artifacts that are difficult for computers to handle. Apart from these "errors," understanding the natural language in clinical reports requires common sense and background medical knowledge. It is common to encounter terms that are not part of the resources provided to the machine, despite the considerable number of synonyms in many terminologies. When developing models, especially in the clinical domain, specific formulations such as elliptical conjunctions and hierarchical segmentation of relationships must also be taken into account. Despite recent improvements in natural language models, machine understanding of language, let alone of clinical documents in French, is still far from being solved. English has many more processing tools and terminology resources than other languages, and not all English approaches are directly transposable to French. Moreover, although there is much work in French on general domain texts, the biomedical domain is still lagging behind (Névéol et al., 2018). As an example, despite being the 5th most represented language in the 2019 version in the UMLS terminology, French has synonyms for only 3.5% of its concepts. Therefore, an important aspect of this work is the development of methods for clinical NLP in French.

A case study In this thesis, we will address the task of structuring radiology reports (Chapter 5). Solving this task involves the various research topics mentioned above. This study, approved by the institutional review board at APHP (CSE 190022), is part of the EZMammo project, which main objective is to optimize the clinical data-warehouse of the Assistance Publique des Hopitaux de Paris (APHP) and validate the predictions of a deep learning imaging

algorithm on mammograms. A preliminary task of this evaluation is to build a dataset of mammograms labelled with the cancer diagnosis and the lesions found in the corresponding reports. In the case of suspicious lesions, the radiological examination is followed by a cytological analysis. We must then match the findings of both reports to label the original mammogram with the definitive diagnosis. This processing entails the ability to extract medical entities (procedures, scores, lesions) and spatial, temporal, and morphological features. Using these extractions, we can filter and align the results between radiological images, mammography reports, and anatomo-pathology reports. The target entities are composed of multiple labels and multiple textual parts. Thus, they fall into the category of structured entities. This structured entity extraction task involves multiple subtasks, namely named entity extraction to locate object mentions and their characteristics, normalization to finely label them, and composition of these mentions to construct structured entities.

1.1 Research questions

A first line of questioning arises from the problems related to structured representations. Simple entity extraction and normalization may not be sufficient to adequately represent the information present in a clinical report. Therefore, **which structure is better suited to the extraction of information in the clinical domain?** In the case of structured entities, **how do we model a system to group the different parts of the same entity?** More generally, **in the case of both simple and structured entities, what challenges are encountered when these entities overlap, and what methods can be used to overcome them?**

Our second series of questions comes from the language domain itself. Since English is the predominant language of NLP research, **can we build NLP for languages other than English, such as French?** A subsidiary question arises: **when few resources are available in languages other than English, as in the case of normalization, is it still possible to apply learning models to these languages?**

Finally, our last question comes from the requirement of annotated data in deep learning. Since the cost of annotating medical documents is high, **what techniques can be implemented to train deep learning algorithms in the low-data regime?**

1.2 Contributions

To answer the previous research questions, we present the following contributions related to steps ①, ② and ③ in Figure 1.1. Our works on named entity recognition and structured entity recognition introduce multiple methods to handle the extraction of overlapping entities. In the case of simple entity recognition, we show that sequence labelling methods are better suited for the extraction of long and ambiguously annotated entities. In the case of structured

entities, we introduce the concept of mention cliques to compose structured overlapping entities, as well as a new mechanism of relation prediction with mention scopes.

We also address the issue of training models in languages other than English. We evaluate all of our models on French datasets and develop a new annotated corpus of clinical radiology reports. We also demonstrate the benefit of training with multiple languages jointly in the case of medical concept normalization.

Finally, in the low-data regime, we showcase multiple techniques to inject external medical knowledge into the training of learning-based algorithms, while alleviating the need for language or domain specific pre-processing methods. In the context of radiological entity extraction, we show that the hybridization of a set of output constraints, a terminology and a learning-based method enables our method to be effective with few annotated reports.

1.3 Outline

We structure our work in four main chapters and our work can be summarized by these three verbs: *tag*, *normalize* and *compose*.

— The first chapter contextualizes our objectives by focusing on computer text representation, which is transversal to most NLP tasks.

— In the second chapter, we propose two methods to extract simple overlapping named entities (*tag* ①) and evaluate our method on medical- and general-domain datasets, in English and French.

— In the third chapter, we address the problem of normalization (*normalize* ②) of medical terms in languages with low terminology coverage, and propose a normalization algorithm using supervised or distantly supervised learning.

— In the fourth chapter, we focus on the issue of extracting structured entities (*compose* ③) in clinical reports. In particular, we design an annotation scheme and present a new structured entity dataset of annotated clinical radiology reports. We also propose a method to extract these structured entities and evaluate it on the dataset.

Finally, we close this thesis with several research perspectives in the last chapter.

1.4 Published work

The material presented in Chapter 3 is based on three publications, one at the 2021 AIME conference (Wajsbürt et al., 2021b) and two as part of the TALN-DEFT challenge, dedicated to the analysis of clinical cases in French in 2019 (Wajsbürt et al., 2020) and 2021 (Gérardin et al., 2021). The material presented in Chapter 4 is based on a journal article in JBI (Wajsbürt et al., 2021a). The material presented in Chapter 5 has not been published yet.

Chapter 2

Background

Contents

To introduce the objectives of extraction and normalization of simple or structured entities, we must first introduce the recent developments in computer representation of text, which are transversal to many NLP tasks.

We will then discuss the work that has been done on simple entity extraction in texts, and the issues that remain. This task is an essential sub-task of text processing for many information retrieval applications, and as such constitutes a preliminary step for both the normalization of medical entities and the composition of structured entities.

Once these simple entities have been extracted, we will address the issue of their normalization, a topic that aims at improving the interoperability of systems that use these extractions.

Finally, we will consider the specific issue of structured entities, focusing on the case of breast imaging reports. We will make the connection between our objective and various existing NLP tasks studied in order to better define it.

2.1 Computer representations of text

Semantic representations of text in computing have been the subject of several decades of studies. This line of research aims at producing representations of words or characters that are globally "useful" for downstream NLP tasks. This encompasses several topics such as text segmentation, robustness to spelling errors or application to new contexts, domains or languages in order to improve the generalizability and robustness of downstream NLP models.

We will focus on some aspects of these developments, which are transversal to all NLP disciplines, and thus to the topics addressed in this thesis.

2.1.1 Textual units

In order to be processed by computers, texts must first be broken down into small units called tokens. This splitting affects the generalizability of a system, since a never-before-seen sample can be treated as a composition of several previously observed subsamples. For example, if a model has learned to detect "breast cancer", and "lung melanoma", it could be able to generalize to "breast melanoma" by splitting the phrase into words.

Words The granularity of the splitting is thus often set intuitively by splitting the sentences word by word. This splitting also affects the outputs produced by the system. Indeed, a named entity recognition system will not be able to predict an entity stopping in the middle of a word if the splitting is done around the words.

Characters It is also possible to segment the text into character n-grams. For example, the word "melanoma" could be split into multiple sub-strings of arbitrary size "mel", "ela", "ano", etc. Some efforts have also been made to represent the text character by character. These systems lend themselves well to morpho-syntactically rich languages and enable the representation of rarer words. For "breast melanoma", this segmentation would produce the sequence "b r e a s t _ m e l a n o m a"

Subwords More recent works (Kudo, 2018; Sennrich et al., 2016; Wu et al., 2016) have introduced sub-words as the main processing units. These segmentation techniques split words such that every generated subword is part of a given limited vocabulary (between 30,000 and 100,000 words most of the time). They solve the problem of rare and unseen words, while keeping a balance between the size of the vocabulary and the size of the tokenized sequence. An example of subwords sequence would be "breast mela_ noma".

2.1.2 Terminologies and hand-engineered features

2.1.2.1 Hand-engineered features

After segmenting the text into units, each of these units is commonly mapped to a set of features. Features can be described as numerical characteristics associated with each textual unit, and can be integer, boolean or real.

Early NLP methods relied on word case, punctuation, presence of digits, morphological properties such as affixes or suffixes, or Part Of Speech (POS) labeling, among others. For example, the word "Apple" has an uppercase feature of 1, a POS verb feature of 0, and a contains-digit feature of 0, and could therefore be represented by the vector [1, 0, 0]. Interested readers can refer to Nadeau and Sekine (2007) for a more detailled review of such features.

2.1.2.2 Terminologies and term lists

Early NLP systems made extensive use of terminologies. These terminologies can be described as dictionaries in which a variety of expressions are represented according to different characteristics. The expression "breast melanoma" can thus be associated with an identifier (ex: CUI C0346787 in the UMLS) a label (Disease) or other features useful to a downstream system. The search for these entities in the texts was then mainly done by exact match, or distance calculation between pieces of text and terminological entries at the word or character level. In particular, this step was commonly part of the preprocessing stage of early systems, rather than an objective itself. The matched entries could then either be used as inputs to decision systems, or be converted into features for further processing of the text sample. Other features could be derived from the word themselves.

2.1.3 Modern input features

2.1.3.1 Word embeddings

A word embedding is a set of real features associated to a word and computed by machine learning on a set of tasks. This term is also used to denote embeddings of sub-words embeddings when a different tokenization algorithm is used, as mentioned in Section 2.1.1. Word embeddings were introduced to the NLP community by Collobert and Weston (2008) and have become the de facto standard for analyzing text with machine learning. It is not clear what the exact meaning of any of these features is, but it is commonly assumed that they capture the implicit semantics of words. Word embeddings can be learned from scratch, or computed from morphological features using character embeddings for instance (Akbik et al., 2018; Bojanowski et al., 2017; El Boukkouri et al., 2020; Klein et al., 2003; Peters et al., 2018).

This term is typically used in the context of neural networks. However, in modern NLP systems, it is often not clear which part of a model is responsible of text representation and which one is responsible for the specific task that is being addressed. We will assume that a word embedding refer to any representation that we can map to the original tokenized sequence. A model can therefore produce multiple word embeddings for the same word, for example by focusing on different characteristics. For instance, in the BERT model, the multiple embeddings are assumed to represent increasingly refined versions of the initial embedding and some studies have shown that word embeddings of lower layers in a language model encode more local syntax while higher layers capture more complex semantics (Tenney et al., 2019).

These features are then combined through a set of operations that compose the different layers of a neural network. An exhaustive review of the different types of layers is beyond the scope of this thesis but we will list a few standard components of these systems. Most of these transformations are built upon feed forward networks that allow non-linear transformations in the feature space.

2.1.3.2 Convolutional neural networks

Convolutional neural networks (Krizhevsky et al., 2012) operate as transformations on small sliding windows of words (or images). They are best suited for local pattern detection. They have been used for text classification (Kim, 2014), NER (Collobert et al., 2011), normalization (Li et al., 2017; Limsopatham and Collier, 2016), as well as character-level pattern extraction (Klein et al., 2003).

2.1.3.3 Recurrent neural networks

Recurrent neural networks, in particular Long Short Term Memory networks (LSTM) (Hochreiter and Schmidhuber, 1997), work as continuous state machines that process each word of a text successively by updating an internal memory. The LSTM cell uses a forget gate and an input gate to store, retrieve and overwrite a memory state which allows it to better "remember" the previously processed tokens on longer ranges. There also exists other variants like Gated Recurrent Units (Dey and Salemt, 2017). These networks are generally slower than CNNs but are well suited to sequences and the detection of patterns involving a particular ordering of words or interactions over a longer distance.

2.1.3.4 Attention

The attention mecanism (Bahdanau et al., 2017; Vaswani et al., 2017) operates as a fuzzy search mechanism in a list of embeddings. Each word in the text computes two "key" and "value" vectors, and a "query" can be performed by computing a weighted sum of the word values and a similarity score between their key and the query vector. This mechanism is useful for modelling long-distance interactions, or for samples without a specific order (like graphs) and is nowadays at the core of many deep learning models.

2.1.4 Pretrained representations

The idea of learning textual representations before specializing them on a specific task has acquired a considerable popularity since the last decade. These representations have in common that they are the result of optimizations of a representation model on large corpora of texts. However, they differ in the architecture of the pre-trained models, the granularity of the textual units and the learning objectives of the pre-training.

2.1.4.1 Static word embeddings

Training the input word embeddings through auxiliary tasks such as language modelling has been a crucial step to enable their use in neural networks (Collobert and Weston, 2008; Collobert et al., 2011; Mikolov et al., 2013; Turian et al., 2010). The specific pre-training task of language modelling on a large corpus was introduced as Word2Vec by Mikolov et al. (2013), followed by GLOVE (Pennington et al., 2014). The language modelling objective builds on the idea that "a word is characterized by the company it keeps" (Firth, 1957; Harris, 1954). This was pinned as the distributional Hypothesis by (Sahlgren, 2008) and more thoroughly studied as distributional semantics (Baroni and Lenci, 2010; Turney and Pantel, 2010). Other variants such as FastText (Bojanowski et al., 2017) build their word representations from character n-grams and have become a popular solution for representing previously unseen words.

However, these embeddings do not take into account the context of the word when used in a new sentence. This can severely limit their usefulness in some cases, such as representing homonyms (does "bear" refer to the animal or the verb?), or referent words like pronouns.

2.1.4.2 Contextualized word embeddings

The ELMO contextualized word embeddings (Peters et al., 2018) improved static word embeddings by pretraining a full deep recurrent language model and using the hidden representations as features for downstream tasks. It was followed by the BERT model (Devlin et al., 2019) with the masked language modelling objective. Many variants have since been designed, either modifying the model and its training (Clark et al., 2020; Dong et al., 2019; Kong et al., 2020; Liu et al., 2019; Yang et al., 2019), or the pre-training corpus domain (Beltagy et al., 2020a; Lee et al., 2020; Martin et al., 2020; Ruder et al., 2019). A comprehensive review of this research field can be found in Qiu et al. (2020). It is worth mentioning that the HuggingFace library (Wolf et al., 2020) contributed to the popularity of these models by simplifying their implementation and sharing.

2.1.5 Large language models

Recently, a paradigm shift has been brought by deep autoregressive language models. Several information extraction tasks can in some cases be written in text format through a question and an expected answer. The answer can then be binary, multiple choices or open. For example, a classification task sample could be represented as "Is the following text about NLP? Image classification has known many successes since CNNs. Answer: no". Similarly, a NER task sample could be written as "Extract the different locations mentioned in this text: I moved to London in 2000 before returning to Paris a year later. Answer: London, Paris". It has been shown that language models pre-trained on large amounts of text can correctly complete these questions with the most likely answers (Lewis et al., 2020; Radford et al., 2018, 2020; Raffel et al., 2019), sometimes with relatively few task-specific examples (Brown et al., 2020). Thus, the entire pre-trained model serves as a common backbone for various tasks, without the necessity of redesigning a specific architecture for each. Although these models hold much hope and promise, their enormous size, the biases associated with their training, and the potential abuses surrounding their use raise many ethical questions (Bender et al., 2021).

2.2 Named entity recognition

The term "named entity" emerged during the MUC program in the early 1990s. Formally, a named entity is characterized by a textual beginning and end, and a possible type. While earlier efforts focused mainly on entities in the form of noun phrases, the task of entity recognition

has evolved and now aims at extracting any entities, sometimes long and comprising several noun phrases or verbs. This task constitutes a cornerstone of information extraction tasks, as it allows the decomposition of a text into semantic units that can be more easily processed by a computer, and interpreted by a human (Ehrmann, 2008).

To a lesser extent, variants of the task also allow disjoint entities (with gaps) and have been addressed by several works but we will not focus on this case in this section.

A notable difference between the different NER methods is their ability to extract overlapping entities. The overlapping NER problem is commonly referred to as "nested NER" or "overlapping NER". In contrast, the non-overlapping NER problem is referred to as "flat NER". The overlapping entities may be of different types, suggesting the use of several specialized models for each type. However, they can also be of the same type, which makes their extraction more difficult.

2.2.1 Proposed methods

2.2.1.1 Earlier works

The first published work that addressed the task of detecting entities in a text was the one of Rau (1990). The first NER systems relied heavily on handcrafted rules and various heuristics. As described in Section 2.1.2.1, these rules and heuristics used lexical functions, gazetteer lists, POS labels, and other handcrafted features. To address the ambiguity of the language and the need for annotation for similar terms, multiple methods performed an augmentation of the initially annotated data by building a set of context from their entities and building a set of candidate entities from their context. These gathered entities and contexts are turned into a set of heuristics and handcrafted rules to allow generalization. Brin (1999) applies lexical rules to detect movie names in websites and complete the initial rules. Collins and Singer (1999) gather entities rules and context rules iteratively to recognize general domain entities, starting with a set of entities rules. Riloff and Jones (1999) apply Mutual Bootstraping and perform these steps automatically, starting from a set of candidates. The formal work of Lin (1998) on language distributionality is used by Paşca et al. (2006) to produce a set of similar words to further augment the entity rules and context rules. Alfonseca and Manandhar (2002) use the WordNet graph (Miller et al., 1990) to define seeds by listing the most frequent co-occurrences between the nodes and the target entity class. They subsequently use the graph children to generate candidates entities. Etzioni et al. (2005) use web queries similarity defined as Pointwise Mutual Information and Information Retrieval (PMI-IR) by Turney (2001) to define the similarity between candidates and contexts.

2.2.1.2 Sequence labelling systems

Ramshaw and Marcus (1999) have formally cast the NER task as a word classification problem. Until recently, most machine learning systems have approached the problem using this formulation. In sequence labelling NER, each word is assigned a single tag (or label) describing its relative position in an noun phrase and the produced tag sequence can be parsed to recover noun phrase entities. The first tag schemes were IOB or IOE variants in which each word is classified as being (I)nside an entity, (O)utside an entity, at the (B)eginning of an entity or at the (E)nd of an entity. When dealing with multiple entity types, these scheme use specific tags for each entity. The (O)utside tag is shared and represent the absence of any entity of any type at a given position. The IBES tags are declined as I-A, B-A, E-A, S-A where A refers to a given entity type. This prevent the system from producing multiple non O tags at a given position, and therefore impose the flatness of the produced solution. Ratinov and Roth (2009) further study the BIOUL (or equivalently IOBES) tag scheme and find that it obtains the best performance of the CoNLL dataset. This scheme encodes the end of entities and single words entities with specific tags E and S [1].

Supervised methods such as Random Forest and chain graphs became an topic of growing interest in NER since 1997. These models were often given a list of handcrafted features about each word of a sequence, and learned to predict if a word was part of an entity as well as the entity type: Hidden Markov Models (HMM) (Bikel et al., 1997), Decision Trees (Sekine, 1998), Maximum Entropy Models (ME) (Borthwick et al., 1998), Support Vector Machines (SVM) (Asahara and Matsumoto, 2003), and linear chain Conditional Random Fields (CRF) (McCallum and Li, 2003). The latter model was introduced by Lafferty et al. (2001) and is still used as a building block of modern systems. Deep neural networks were introduced to the NER task by Collobert and Weston (2008), as they developed a deep neural network to jointly learn NER and other tasks such as language modelling. Since then, sequence labeling NER systems have essentially evolved with advances in deep learning representations. Huang et al. (2015) incorporated LSTMs in the design of their system. Klein et al. (2003) proposed a character CNN encoding of the words. Lample et al. (2016) improved their system in various ways and propsosed a LSTM based character word embedding. As Devlin et al. (2019); Peters et al. (2018) proposed contextualized embeddings, they improved the performance of NER systems significantly as a result. However, these systems only focused on flat NER.

2.2.1.3 Nested NER via iterative sequence labelling

The GENIA corpus (Kim et al., 2003) led to the first work focusing on nested NER (Gu, 2006; Shen et al., 2003; Zhang et al., 2004; Zhou et al., 2004; Zhou, 2006), mainly involving focusing on either the outermost or innermost entities in a sentence, or specific entity types.

1. The (E)nd tag is also commonly referred as the (L)ast tag, and (S)ingle tag as (U)nary tag, hence the BIOUL scheme.

Since 2018, nested named entity recognition has been the subject of renewed attention in the biomedical NLP community, leading to many different approaches.

Alex et al. (2007) study multiple problem transformations to frame the nested NER task as cascaded flat NER tasks, each focusing on either a specific nesting level, or a specific label. However, their approach did not model overlapping entities of the same type. Ju et al. (2018) designs a layered architecture that predicts entities at each layer and merges the word representations before applying the next layer. Fisher and Vlachos (2019) uses a fixed number of layers and updates spans representations using a novel neural architecture. Shibuya and Hovy (2020) compute tag scores for each word and decode the spans by applying the Viterbi algorithm multiple times on a previously extracted subsequence, starting from the full sentence.

2.2.1.4 Nested NER via non linear tag sequences

Another approach is to create a hypergraph of the words in the sentence, such that it captures the structure of the overlapping entities. Finkel and Manning (2009) model the nested NER task as a constituency parsing graph extraction. Their approach could extract nested entities of the same type, at the cost of expensive computations and the need for Part-of-Speech (POS) features. Some methods model the span detection with hypergraphs to account for the non-linear structure of the tag sequences. Lu and Roth (2015) design a CRF hyper-graph with various node types to model entity types and boundaries. However, cycles in the graphs of some samples required that the CRF normalization term had to be approximated, leading to a decreased performance (Muis and Lu, 2017). Muis and Lu (2017) model the mention edges and transitions instead of solely modeling token tags. Their method, however, requires multiple graphs when there are more than one entity type. Alternatively, Katiyar and Cardie (2018) only model mention tags and not their transitions, but allows a multi-label prediction for each token. They modify an LSTM layer to represent multiple tags for a single word and perform decoding during the recurrent neural network execution.

2.2.1.5 Exhaustive NER systems

Another class of methods addresses the problem by enumerating all possible spans of the input sequence and classify each one with its label, including a "no entity" class. Sohrab and Miwa (2018) compute a representation for each span from its word embeddings and classify each entity. Xu et al. (2017) propose a similar model but consider the left and right context when classifying the spans. Wang et al. (2020) use an LSTM cell (Hochreiter and Schmidhuber, 1997) to model dependencies between spans that differ by one token. Zheng et al. (2019) first filters candidate mentions by predicting all possible start and end tokens and then predicting a label for every mention that starts or end at one of the boundaries. Luan

et al. (2019) also enumerate and classify spans but allow them to communicate through a graph attention mechanism.

2.2.1.6 Recent works

There are several other formulations of the NER task that do not involve sequence labelling or exhaustive enumeration of entities. Tan et al. (2021) redefine the task as a sequence-to-set problem and use a fixed number of entity slots where each slot fills in its start position, end position and label, or is classified as empty. Their method allows the prediction of any type of overlapping entities. Li et al. (2020); Mengge et al. (2020) conceptualize the problem as a machine reading comprehension task. In their work, a pre-trained language model is prompted with a query such as "Find the organizations in this sentence: ", followed by the sentence. The start and end boundaries of the relevant entities are then extracted by classifying each representation in the sequence. Combined with transfer learning, these methods show promising results in predicting new entities types without having to annotate these types. De Cao et al. (2020) use a pretrained deep language model to rewrite the input sequence with markup tags indicating the beginning, end and label of the entities. However, they do not adapt their method to overlapping entities. Finally, Yan et al. (2021) propose the combination of the BART (Lewis et al., 2020) Seq2Seq model with a pointer mechanism to extract flat, nested and overlapping entities.

2.2.2 A word about object detection

The field of research aiming at segmenting and labeling objects in images has developed in parallel with the research on entity recognition in texts. It is hard not to see some similarities between these two tasks. An exhaustive review of the proposed systems is beyond the scope of this thesis, but we will quickly describe the convergences between these domains. Readers interested in object detection can refer to Guo et al. (2018); Zhao et al. (2019).

Image object segmentation aims at classifying the pixels of an image according to different types, and thus at reconstructing the objects from the labels associated with the image. Some earlier works of object segmentation were based on the notion of superpixels (Felzenszwalb and Huttenlocher, 2004), and the classification of each superpixel according to a label. Although it is much more complex, the initial superpixels segmentation is akin to the initial tokenization step in word processing, which consists of breaking down the sample to be processed into simpler units. Each superpixel is then represented by several features such as its size, color or relative position in the image, and then labeled by models such as HMM, CRF in order to take advantage of local interactions between the labels (a piece of grass is likely to be close to another piece of grass).

Another similar task aims at predicting the bounding box of different features in an image. An analogy to NER would be to think of the begin-end span as the bounding box of an entity.

Most models perform prediction in two steps: a first selection of possible regions of interest of an entity is performed, and then for each candidate a second model labels whether the entity is.

Some NER works have drawn inspiration from advances in object detection: Li (2021) employ a two-stage decoder similar to Ren et al. (2015). They extracts region proposals and classify each region to either obtain a label or to choose not to predict it. The work of Tan et al. (2021) builds on the system of Carion et al. (2020) to transform the problem into a sequence-to-set prediction.

2.2.3 Annotated corpora

There are many NER corpora that vary according to different aspects such as the domain, the language, the overlap of the entities, their size or their type. We will use the GENIA (Kim et al., 2003), DEFT (Cardon et al., 2020) and CONLL 2003 (English) (Sang and De Meulder, 2003) datasets for the experiments in this thesis. Statistics about these datasets can be found in Table 2.1

2.2.3.1 DEFT

The DEFT corpus contains 167 texts describing french clinical cases, including 67 for testing. The different types of entities are, on the one hand, *pathologies* and *signs or symptoms* (DEFT task 3.1), and on the other hand, *anatomy*, *anatomy examinations*, *substances*, *doses*, *administration methods*, *treatments (surgical or medical)*, *values*, *time* (DEFT task 3.2). Named entities can nest up to 3 levels deep and two distinct entities of the same type can overlap. We used the provided train and test splits.

2.2.3.2 GENIA

The GENIA corpus contains 2000 MEDLINE abstracts, or 18546 sentences, including 1855 for testing. The annotations focus on transcription factors in human blood cells, and were named entities. Most evaluations follow Finkel and Manning (2009) and Lu and Roth (2015) and collapse all DNA subtypes into DNA, RNA subtypes into RNA, all protein subtypes into protein and kept cell line and cell type. Named entities can nest up to 4 levels and two distinct entities of the same type can overlap. We perform splits following Finkel and Manning (2009): the last 10% of the sentences are used to test the model, the remaining 90% are the training set.

2.2.3.3 CONLL 2003

The shared task of CoNLL-2003 concerns general domain NER in four languages: English, German, Dutch and Spanish. It annotates four types of named entities: persons, locations,

organizations and names of miscellaneous entities that do not belong to the previous three groups. The English data were taken from Reuters news articles published between August 1996 and August 1997. It contains 1393 articles, or 22,137 sentences, including 216 articles for development and 231 for testing. There are no overlapping entities in this dataset. Although this corpus does not contain biomedical nor nested entities, it is a classical open comparison point with other NER models.

2.2.4 Evaluation metrics

2.2.4.1 Precision and recall

The information retrieval systems are classically evaluated using three metrics: precision, recall and F1 measure.

$$\text{precision} = \frac{\text{number of true positives}}{\text{number of predicted entities}} \tag{2.1}$$

$$\text{recall} = \frac{\text{number of true positives}}{\text{number of gold entities}} \tag{2.2}$$

$$\text{f1} = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} \tag{2.3}$$

A system with a good recall but a low precision might be useful as a pre-filtering step of a larger sequential model. A system with a worse recall but a better precision might be useful when combined in parallel with other models.

2.2.4.2 What counts as a true positive ?

The outputs of a NER systems are triplets (begin, end, label). A common choice is to apply the precision, recall and F1-score measure metric on these triplets directly.

There are multiple options for what should be considered a correct prediction, or "true positive". The most common one is the "exact match" criteria: a predicted entity must share the exact same bounds and label as a gold entity to be counted as a true positive. Another useful criteria is the "approximate match" criteria: a predicted entity must share a certain fraction of words in common with a gold entity. Indeed, even when the bounds are not perfectly predicted, such as determinants, an entity might have enough words in common with a target entity to still be useful in downstream tasks.

We synthesize all the possible metrics with α, the minimum Dice coefficient between the words of the entities, or intuitively the fraction of words that two entities must have in common to be matched. In our experiments, we will use the "Exact" match metric, with $\alpha = 1$ (bounds must batch exactly), the "Half" match metric with $\alpha = 0.5$ (the number of correct words must be at least half the number of words in the target and predicted entity) and the "Any" match metric with $\alpha = \epsilon+ > 0$ (the target and predicted entity must have at least one word

	DEFT 3.1		DEFT 3.2		GENIA			CONLL EN 2003		
	train	test	train	test	train	val	test	train	val	test
Language	FR		FR		EN			EN		
Domain	Clinical		Clinical		Biomedical			General		
# docs	100	67	100	67	1599	190	213	946	216	231
# entities	5677		2167	1445	46185	4379	5515	23499	5942	5648
avg length	1.94	2.03	4.55	4.74	1.90	2.11	2.05	1.45	1.45	1.44
# unique labels	8	8	2	2	5	5	5	4	4	4
# unique texts	3449	2179	1878	1320	15441	2141	2681	8082	2809	2637
# nestings	475	422	14	4	4524	436	658	0	0	0
# same label nestings	8	2	2	1	2430	234	331	0	0	0
# crossing overlaps	1	0	0	0	0	0	0	0	0	0
# same label crossing	0	0	0	0	0	0	0	0	0	0
# superpositions	0	1	0	0	43	12	9	0	0	0

Table 2.1 Main statistics of the named entity recognition datasets used in this thesis

in common). Finally, a gold entity should not be matched twice, nor should a predicted entity, so we need a procedure to perform matching iteratively.

2.3 Medical entities normalization

Entity normalization (also called entity disambiguation, or entity linking) allows named entities to be linked to concept identifiers. The primary objective of this task is to represent key entities in a text (people, places, diseases, anatomical locations, etc.) by unique references, independent of variations in the form of these entities. This standardization improves the interoperability of the data and of the systems built to process these references.

The normalization problem is better known in the general domain as entity linking (Sevgili et al., 2020; Shen et al., 2015), but differs by the fact that the general domain annotated corpora can leverage larger annotated corpora such as Wikipedia. These make it possible to perform a single supervised training and rely on entity frequencies. However, in most cases, medical terminologies do not provide context nor accurate medical concept frequencies. As such, we will not cover entity linking in the general domain but rather the research done in the clinical and biomedical domain.

2.3.1 Terminologies

Concepts can be described by definitions, or most often a set of lexical variants called synonyms. These concept-synonym associations are collected in terminologies, which act as dictionaries and serve as bridges between medical document annotations and knowledge intensive applications. Terminologies can also be described as "oriented artifacts that relate the various senses or meanings of linguistic entities with each other" (Freitas et al., 2009). Theses terminologies can additionally provide semantic information about hyperonymy (broader meaning), hyponymy (narrower meaning).

Many terminologies have been designed to normalize entities in various domains such as diseases (Bramer, 1988; Organization, 1978), genes (Ashburner et al., 2000) or general medical concepts (Lipscomb, 2000; Spackman et al., 1997) to name a few. Some unification efforts have been made to merge these different terminologies together and provide a unique and large resource for the bioinformatic community. Among them, the Unified Medical Language System (Bodenreider, 2004) is the most noteworthy. Therefore, most target vocabularies can nowadays be referred to as subsets of the UMLS. We will use the UMLS and Mantra terminologies to evaluate our normalization models, so we will describe them now.

2.3.1.1 UMLS

The Unified Medical Language System (UMLS) is a large terminology that unifies concepts from several dozen terminologies in the biomedical domain. Each concept in the UMLS is

assigned a Concept Unique Identifier (CUI), a set of terms (or synonyms), possibly in multiple languages, and a semantic type. UMLS semantic types are grouped in 15 semantic groups and each concept is associated with one semantic group, with very few exceptions (McCray et al., 2001). For example, "Eicosapentanoic acid" (concept C0000545) is in the chemical (CHEM) group, while "Accountant" (concept C0000937) is in the living beings (LIVB) group. The UMLS 2014AB version contained 5,772,518 synonyms for 2,528,878 concepts, while the 2019AB version contained 9,187,793 synonyms for 4,258,236 concepts.

2.3.1.2 Mantra

The Mantra terminology was developed at the same time as the MantraGSC dataset (Kors et al., 2015) and contains a subset of the UMLS, consisting of all concepts from three terminologies: MeSH, SNOMED-CT, and the Medical Dictionary for Regulatory Activities (MedDRA). There are 3,164,910 synonyms for 591,918 concepts in five languages (English, Spanish, French, German and Dutch). The concepts were filtrered to only keep those that belong to one of the ten semantic groups *Anatomy*, *Chemicals and drugs*, *Devices*, *Disorders*, *Geographic areas*, *Living beings*, *Objects*, *Phenomena*, *Physiology*, and *Procedures*.

2.3.1.3 Non English terminologies

The UMLS terms are mostly in English. For all other languages, such as Japanese, Dutch or French, the number of terms was less than 5% of what is available for English in 2014. French is the 2nd (resp. 5th) most represented language in the 2014 (resp. 2019) version in the UMLS, but only 3.5% (resp. 3.6%) of the concepts have terms in French. Efforts have been made to improve this coverage by manual or automatic translation, or by mapping local terminologies, leading to more complete resources out of the official UMLS (Deléger et al., 2010; Grosjean et al., 2011; Marko et al., 2006; Névéol et al., 2014; Zweigenbaum et al., 2003). However, the gap is still significant, and this represents a real pitfall for the NLP systems in French, and more generally, in all languages other than English (Névéol et al., 2018).

2.3.1.4 Ontologies

Terminologies often complement ontologies. Ontologies express the semantic relations between different concepts through description logics. They allow decision systems to reason about individuals and their attributes, classes or relationships. The commonly accepted definition is that of Gruber (1993) "An ontology is an explicit specification of a conceptualization. [...] A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose".

Reasoning from facts defined in ontologies can be done by different reasoners like Protege (Musen and Team, 2015), using first-order formal logics. Reasoners and machine learning

techniques are not exclusive. Efforts to integrate propositional logics into machine learning models have been made to improve predictions (Deng et al., 2014), and Markov Logic Networks have proven successful in making first-order logic reasoning more flexible (Domingos and Lowd, 2009).

2.3.2 Proposed methods

2.3.2.1 Earlier works

Many earlier works are rule-based methods. These methods revolve around matching the entity to be normalized with one of the entries in the target dictionary, by comparing the form of the entities using a set of handcrafted rules, and make use of several common techniques.

A popular technique consists in expanding the lexical forms taken by a given entity or synonym (Aubin and Hamon, 2006; D'Souza and Ng, 2015; Golik et al., 2013; Hanisch et al., 2005; Jonnagaddala et al., 2016; Schuemie et al., 2007). For example, a disease with many terms can be transformed into an acronym. Nouns can be made plural, or singular, or lemmatized, i.e. stripped of any grammatical variation as in Schuemie et al. (2007). For example, "painful" can be converted to "pain". These term augmentations can be applied on both entities and dictionary synonyms. There has also been efforts toward automatizing these term augmentations (Ghiasvand and Kate, 2014; Tsuruoka et al., 2007).

Another common technique consists in combining different synonyms from the same or other terminologies to augment the target terminology (Aronson, 2001; Aubin and Hamon, 2006; Hanisch et al., 2005; Jonnagaddala et al., 2016; Kuo et al., 2009). For exemple, the word "eye" can be replaced by "occular" in many synonyms.

Once the entities and dictionary entries have been expanded, the matching step occurs. An entity and a synonym can be matched if they have the same form or only differ by a few words or characters. For example, the MetaMap system (Aronson, 2001) allows a synonym to become a candidate if is within a character distance of two of the entity. In the case where several synonyms can be matched to the entity, several filtering decisions can be made, based for example on the confirmed presence of one of the entities in the document as in D'Souza and Ng (2015); Hanisch et al. (2005) or other features such as the reliability of the entry source (Lee et al., 2015), or the semantic group of the annotated entity. These filtering steps can be cascaded until only one candidate is left as in the work of D'Souza and Ng (2015).

2.3.2.2 Machine learning approaches

Although many rules are still used in modern normalization systems, machine learning approaches have become increasingly important in the design of these models. Most of the proposed solutions generate a set of candidate synonyms (synonyms or concepts), and rank these candidates using a scoring model.

To represent entities and synonyms, some previous systems relied on TFIDF-like approaches at the word level (Leaman et al., 2013; Leaman and Lu, 2016) or by taking a larger number of textual features (Castano et al., 2016). Simple word embedding sum approaches have been used successfully (Castano et al., 2016). Other recent systems use LSTMs (Liu and Xu, 2018; Phan et al., 2020; Tutubalina et al., 2018), CNNs (Arbabi et al., 2019; Deng et al., 2019; Li et al., 2016; Luo et al., 2018; Mondal et al., 2019), or BERT-like pre-trained Transformers (Ji et al., 2020; Sung et al., 2020). The comparison between the produced representations is often either computed from their a scalar product, cosine similarity or the Euclidean distance.

The proposed approaches fall into two categories: learning can be carried out on the similarity between the entity to be normalized and the synonyms in the dictionary, or on the similarity between the entity and the concepts directly.

Synonym similarity methods The training objective of systems comparing entities and synonyms is not trivial. Indeed, multiple correct synonyms may exist for a given entity, and a multi-class classification style approach accepting only one solution is not appropriate. Most systems therefore rely on a ranking mechanism such as pair-wise learning to rank (PLTR) (Huang et al., 2011; Leaman et al., 2013; Liu and Xu, 2018) in which a correct synonym should be given a higher score than a wrong one. Similarly, Mondal et al. (2019) use Triplet Networks (Hoffer and Ailon, 2015) to rank candidate synonyms and Fakhraei et al. (2020) uses Siamese Networks combined with contrastive loss. Tutubalina et al. (2018) propose a method consisting in keeping only the highest similarity score among the synonyms of a concept, and training the model with the cross-entropy classification loss. Finally, Sung et al. (2020) propose to marginalize the positive synonyms, i.e. by maximizing the sum of the probabilities of the correct candidates using a cross-entropy classification loss.

Concept similarity methods Methods in the second category compare entities and concept representations. It is then necessary to generate a representation for each concept, which can be done in a more or less explicit fashion. Tutubalina et al. (2018) suggests representing a concept as a concatenation of its synonyms, then performing a standard classification. Wright et al. (2019) obtains concept embeddings through simple optimization of a classification objective. Hierarchical links between concepts have also been used to improve concept representations (Arbabi et al., 2019; Ferré et al., 2019, 2017). Finally, some efforts toward learning the interactions of concepts in a given sentence have been made by Wright et al. (2019).

However, these learning methods were only evaluated on medium sized terminologies containing between 2000 and 160000 concepts, and to our knowledge no machine learning only method has been applied on larger terminologies.

2.3.2.3 Non English approaches

The normalization of medical entities in languages other than English has so far relied mainly on the translation of English synonyms into the target language (Afzal et al., 2015; Cabot et al., 2016), or conversely, the translation of entities into English (Chiaramello et al., 2016; Perez et al., 2020; Roller et al., 2018). These systems use processing existing rule based indexers like MetaMap (Aronson, 2001) to perform the synonym search, and web-service or local based translation systems (Jiang et al., 2015). In constrast, we chose to design and evaluate an auto-sufficient deep neural network classifier with few to no preprocessing of the input named entities.

2.3.3 A word about person identification

Similarly to how some analogies can be drawn between NER and object detection in images, medical entity normalization can be related to person identification. Indeed, person identification (or face identification) is similar to medical normalization in terms of the very large number of target identities (concepts) and the small number of examples (synonyms) per identity.

Another similarity is the two types of approaches, aimed at either comparing pictures to each other (synonym similarity) (Hermans et al., 2017), or the sample picture and a representation of the person's identity (concept similarity) (Zhai et al., 2019). It is worth noting that since images are less amenable to rule-based processing, these methods cannot benefit from pre-filtering as commonly used in normalization and therefore rely essentially on machine learning models.

For instance, Mondal et al. (2019) used the same triplet networks architecture as Hoffer and Ailon (2015) to learn a distance between the entity (image sample) and possible synonyms (reference images) to match.

2.3.4 Annotated corpora

There exists multiple datasets in medical English and other languages that normalize different types of entities using different terminologies (Dogan and Lu, 2012; Kors et al., 2015; Li et al., 2016). We review the Quaero and Mantra corpus that have been used to evaluate the method proposed in Chapter 4.

Quaero The Quaero FrenchMed corpus (Névéol et al., 2014) consists of two sets of textual documents in French, annotated with concept CUIs from the 2014AB version of the UMLS:

— Titles of research articles indexed in the MEDLINE database
— Information on marketed drugs from the European Medicines Agency (EMEA)

Unlike other normalization corpora such as NCBI Doğan et al. (2014) or BC5CDR Li et al. (2016), the annotated concepts were not limited to vocabularies such as MeSH or MEDIC. However, they were limited to 10 of the 15 UMLS semantic groups. There are two different versions of these corpora. The first version, that we call EMEA 2015 and Medline 2015, was used for the CLEF eHealth evaluation lab in 2015, a challenge for NER and concept normalization. The organizers proposed a training set and a test set for this task. In 2016, a new challenge was organized; the 2015 test set was released as a development set, and a new test set was annotated, leading to a larger corpus containing the previous one.

Mantra The Mantra corpus (Kors et al., 2015) consists of 1450 sentences, annotated with concepts from the Mantra terminology. The annotated documents are in English, Spanish, French, German and Dutch, and consists of

— Titles of research articles indexed in the MEDLINE database
— Information on marketed drugs from the European Medicines Agency (EMEA)
— EPO patents

Many of the texts are translations from each others, so the corpus actually contains 550 unique sentences regardless of the language. Unlike the Quaero corpus, entities were not annotated with their semantic group. Most importantly, there are no training documents as the corpus only contains evaluation samples.

2.3.5 Evaluation metrics

The normalization tasks is commonly evaluated using the standard retrieval metrics, namely precision, recall and F1-score, at the entity level. Some studies (Leaman et al., 2013) also evaluate the performance of the normalization system at the document level: the predicted concepts for all entities are aggregated and evaluated by precision/recall/F1-score for each document, and the resulting scores are finally averaged for all documents to obtain the performance at the corpus level.

These two metrics can be identified by the prefix "micro-averaging" for the entity-level evaluation, and "macro-averaging" for the document-level evaluation. Micro-averaging treats every entity as a unit, regardless of the length of the document in which it occurs. In this work, we will only evaluate our methods using the micro-averaging metrics.

2.4 Structured entities extraction

We define here structured entities as pieces of related information composed of several fields. Each of the fields should, when possible, be justified by a textual mention in order to ensure the transparency of the model and to allow the traceability of predictions in the original document. As we will see, the extraction of information from breast imaging reports lends itself well to this concept. In the rest of this section, we will mostly focus on radiological entities, and relate our task to existing information extraction tasks.

2.4.1 Breast imaging reports: a case study

Breast imaging reports consist of unstructured text written or dictated by a physician. The reports contain multiple measurements, observations, and remarks regarding the patient's condition, including history, potential lesions and their progression, diagnostic procedures performed, such as mammography or ultrasound, and an assessment of the need for further testing in case of suspicious findings. Figure 2.1 shows the English translation of a fictitious but plausible report.

2.4.1.1 Entities

As in other radiology disciplines, the American College of Radiography (ACR) has proposed a set of guidelines to facilitate research and clinical follow-up of patients. The ACR BIRADS (Liberman and Menell, 2002) proposes a standardized lexicon and classification system for breast mammography, ultrasound and MRI. It also recommends a certain organization of reports and the structure of the evaluation. This set of guidelines allows radiologists to communicate results to the referring physician in a clear and consistent manner.

The reported lesions can be described with multiple attributes such as:

— their shape, density and margin
— their laterality: left or right breast
— their relative position in the breast, by a quadrant
— their clock position, e.g. "8 o'clock position"
— a size, indicating either their diameter or their volume (3 dimensions)
— their radial distance to the center of the breast
— the temporality of these lesions, that is their observation was made before or during the exam

The final evaluation grade (or also BIRADS assessment or ACR) ranges from 0 to 6:

— Category 0: incomplete exam
— Category 1: negative
— Category 2: benign

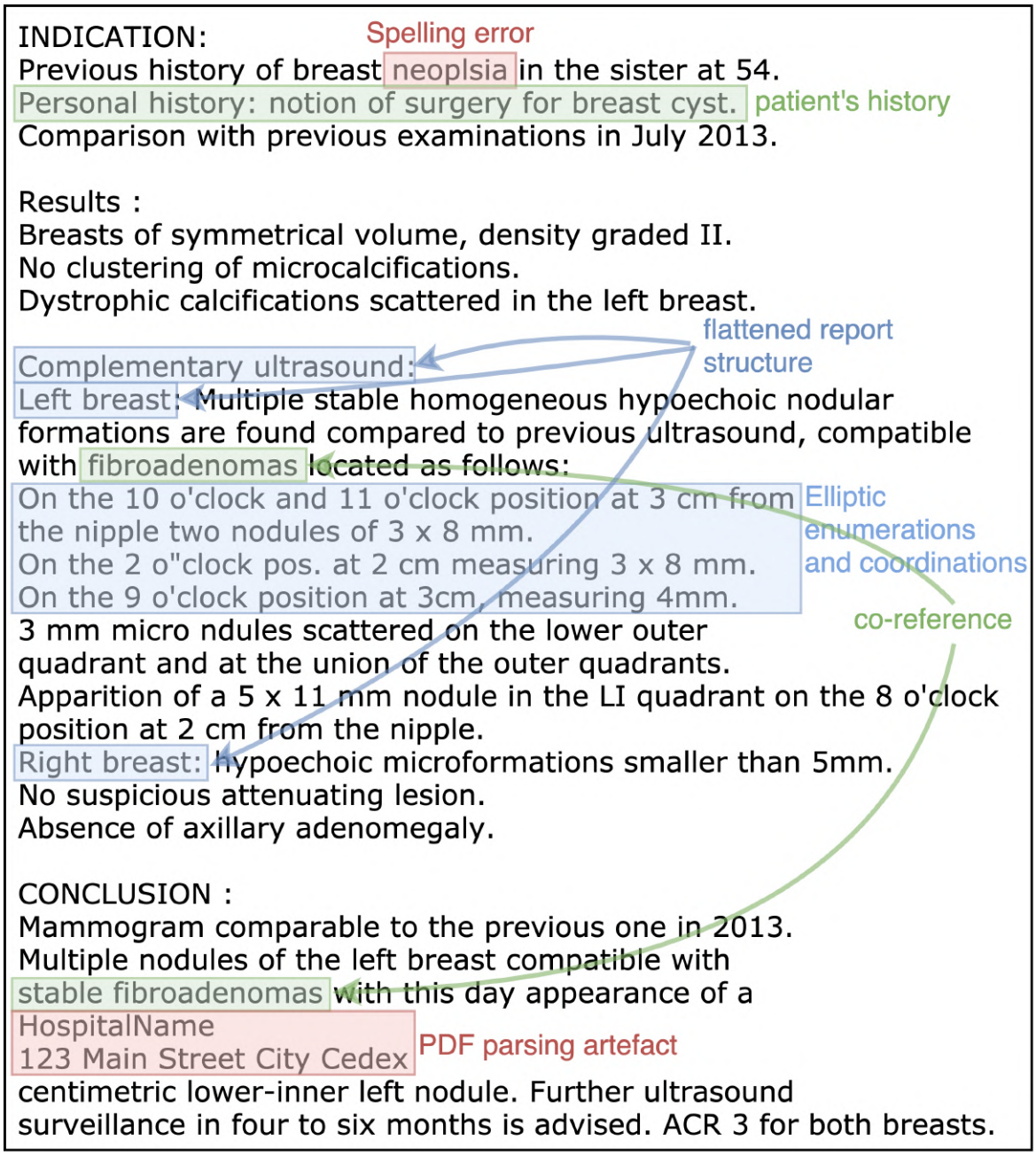


Figure 2.1 Fictitious but plausible mammogram report

— Category 3: probably benign
— Category 4: suspicious mammogram and ultrasound
— Category 4A: low suspicion of malignancy
— Category 4B: Moderate suspicion of malignancy
— Category 4C: strong suspicion of malignancy
— Category 5: strong suspicion of malignancy

— Category 6: known malignancy confirmed by biopsy

The composition of the breast is graded from 1 to 4 according to the percentage of glandular tissue in the breast:

— type 1: the breast is almost entirely fatty
— type 2: there are scattered areas of fibroglandular density
— type 3: the breasts have a heterogeneous density, which can mask small masses
— type 4: the breasts are extremely dense (homogeneous density).

There are also references to diagnostic or therapeutic procedures, which can be past, future or at the time of the visit. These procedures can be characterized by:

— their type: mammography, ultrasound, surgery, chemotherapy, etc
— their anatomical location (breast or other)
— their laterality
— the possible quadrant
— their temporality

2.4.1.2 Report structure

The reports usually include a brief history of the patient's condition, personal or family history of cancer, and previous visits, followed by observations and findings. Noteworthy findings are often summarized in a conclusion. These reports are often organized in a semi-structured manner, with nested sections. However, due in part to conversions between text reports and their PDF edition, this structure is not consistently applied and can be modified throughout the text. This makes the division into sentences and sections far from trivial. Finally, for the sake of brevity, physicians sometimes factorize their findings. These linguistic forms, also known as elliptic coordinations or elliptic enumerations, result in overlapping structured entities:

— *There are small millimeter-sized microcalcifications in the right and left breast*
— *Two lesions are observed in the right breast, measuring 6mm in the UIQ at 3cm from the nipple and 5mm in the LIQ at 2cm.*
— *The left upper inner quadrant contains multiple cysts measuring 6mm and 5mm.*

2.4.2 Structured entities representation

Our objective is to extract different types of entities, as well as attributes qualifying them. These entities should be easily storable and searchable in a database, but also interpretable by locating the zones of a report that mentions them. A certain structure can be found in the elements listed in Section 2.4.1, namely the presence of a mention indicating the existence of a procedure, a lesion or a grade, and different attributes specifying each object, such as its

nature, location or temporality. A useful representation is the one of frames. Frame semantics were introduced by Fillmore (1982), and popularized by the FrameNet project (Baker et al., 1998). A frame is a schematic representation of a situation involving various participants, or conceptual roles. A frame is structured around a "lexical-unit" (or trigger), and composed of "attributes" (or arguments or roles). Each piece of information about a particular frame is held in a slot. As such, the frames are comparable to slices in our representation. As such, we can see frames as key/value tables, on which we add justifications of each field when possible.

However, in the example

"*Right breast: a small nodule of 8mm that was previously measured at 1cm*"

the object ("nodule") is described at several points in its existence and is characterized by a change in its size. A simple key/value list as in the table 2.2 which would list each feature would not be able to properly capture this attribute change over time, and each field would require to be specified (e.g. size → size_now and size_before) to be disambiguated. The process of adding new attributes to better match the representation, known as reification, adds to the complexity of the schema and thus may hinder its generalizability.

field	**value**	**justification**
organ	breast	"breast"
clock position	∅	∅
quadrant	∅	∅
size	8mm	"8cm"
size	10mm	"1cm"
temp	during exam	∅
temp	before exam	"previously"

Table 2.2 Example of a flattened key/value representation of a structured entity

Another ontological formalism has been studied by Burek et al. (2019); Sider (2001) and suggests introducing another dimension to the representation to represent "slices". We could draw inspiration from these works and model objects by a set of slices as in Table 2.3.

	Slice 1		**Slice 2**	
field	**value**	**justification**	**value**	**justification**
organ	breast	"breast"	breast	"breast"
clock position	∅	∅	∅	∅
quadrant	∅	∅	∅	∅
size	8mm	"8mm"	10mm	"1cm"
temporality	before exam	"previously"	during exam	∅

Table 2.3 Example of a temporaly sliced representation of an object

This representation dilemma arises frequently and is intrinsically linked to the granularity of representations. Thus, it can also be found when describing spatial extensions, for example in the case of a tumor covering several quadrants:

"*Breast tumor extending on the upper-outer and lower-outer right quadrants*"

which could be described using two spatial slices as in Table 2.4.

	Slice 1		**Slice 2**	
field	**value**	**justification**	**value**	**justification**
organ	breast	"breast"	breast	"breast"
clock position	∅	∅	∅	∅
quadrant	upper-outer	"upper outer"	lower-inner	"lower inner"
size	∅	∅	∅	∅
temporality	during exam	∅	during exam	∅

Table 2.4 Example of a spatially sliced representation of an object

2.4.3 NLP for cancer and radiology

The extraction of structured information from medical reports has been the subject of many studies. Likewise, many methods have been developed to automatically extract one or more radiological features from clinical reports. Most of these works are not specific to breast imaging reports. Moreover, the extraction objectives vary greatly, in terms of their scope, granularity and form. We will start by focusing on the existing research on radiology reports. Interested readers can refer to existing surveys on the state of NLP in radiology reports (Bitterman et al., 2021; Miwa et al., 2014).

Several works are only concerned with the extraction of a few report-level attributes, and therefore view the task as a classification or term extraction task in EHR for items such as BIRADS scores, histological grade or primary site of lesions (Alawad et al., 2018; Castro et al., 2017; He et al., 2017; Moore et al., 2017; Qiu et al., 2018). Other features have also been the subject of specialized systems such as locations (Datta et al., 2020a). An extensive survey of the different systems proposed for different features was conducted by Datta et al. (2019).

Other works have sought to produce a more detailed and global extraction, and to detect several types of entities at the same time. The earliest work was the one of Taira et al. (2001), who proposed a frame based representation and method for annotating abnormal findings, anatomy, and medical procedures frames in radiology reports. Lacson et al. (2015) used a rule-based system and terminologies to extract abnormal findings and BIRADS scores. The DeepPhe system was proposed by Savova et al. (2017) as a fully integrated software built on cTakes (Savova et al., 2010) to extracts document and patient level cancer summaries (akin to frames) in clinical reports. Steinkamp et al. (2019) proposed a fact-based scheme, in

which each fact is structured around an anchor (such as "cyst") and may contain modifiers (its size, laterality). However, their model is limited by the assumption that all the elements that characterize an entity need to be adjacent inside the fact span. Sugimoto et al. (2021) annotated multiple types of named entities and relations in Japanese chest CT reports but only trained a NER system on their dataset. The facts, anchors and modifiers are then detected by a NER system.

Several methods decompose the problem into two subtasks: named entity detection and relation detection. Unlike (Steinkamp et al., 2019), the relation detection step allows arguments to be distant. Roberts et al. (2019) proposed a frame based scheme for annotating cancer information in clinical reports and a method to perform the prediction (Si and Roberts, 2018). Their method first extracts triggers and modifiers with a NER system, and predicts their relations to form frames. However, their method make the assumption that there is no overlap between the different entities in a text sample, and therefore does not address the problem of factorizations. Recently, a more complex scheme has been proposed by Jain et al. (2021) to annotate nested relationships between different entities. However, these work do not specifically address the case of complex or distant relations between entities.

2.4.4 Related tasks

Our objective of extracting structured entities can be related to four other tasks in different fields of NLP namely slot filling, event extraction, attribute prediction and discontinuous NER.

2.4.4.1 Slot filling

Structured entity extraction can be related to the intent detection and slot filling tasks, also know as semantic role labeling. This task is closely related to the frame semantics formalism. Most often, it is paired with the intent detection task, which consists in detecting the nature of a textual request made by a user. The slot filling task it-self is concerned with detecting the different relevant attributes that compose this request. For example in the query:

"*What are the flights from London to Paris this Saturday?*"

the system must detect that the intent is "*flight information*" if not provided already, and fill the different slots:

— TO: London
— FROM: Paris
— DAY: this Saturday
— TIME: ∅

Most systems turn the task into a named entity extraction, and fill the appropriate slots with the extractions. A comprehensive review of the proposed approaches has been done recently

by Weld et al. (2021). Most often, however, it is assumed that each utterance contains only one intent, which is often not the case (Gangadharaiah and Narayanaswamy, 2019). There has been relatively limited research on slot filling and multiple intent detection Gangadharaiah and Narayanaswamy (2019); Qin et al. (2020). Moreover, among these works, it is assumed that the different intents are of different types. This can be a concern if the user requests information about several flights at the same time for example:

"*What are the flights from London to Paris this Saturday, and from Paris to London the following Saturday?*"

Only one intent type would be detected (flight info), and several slots (London, Paris, Paris, London, Saturday, next Saturday) would conflict in the composition of the entities to extract.

2.4.4.2 Event extraction

Another similar task is the extraction of events in texts. Events in linguistics are most often understood as actions, or situations whose existence is marked by a "trigger" expression (e.g. a verb), and specified by several arguments.

While the ACE event extraction task is focuses on action-like events, the BioNLP shared task datasets are more concerned with interactions between different biomedical entities, where the notion of action is less prominent. As an example:

"*The translocation of the b67 induced by ...*"

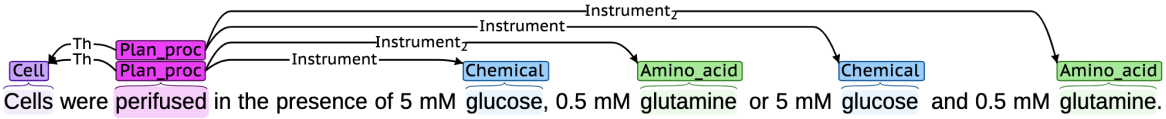
We can then identify:

— the "regulation" trigger: "induced"
— the "cause" argument: ...
— the "theme" argument: b67

Event extraction tasks are divided into two categories: closed world event extraction and open world event extraction. Closed-world event extraction assumes that one already has knowledge of the event pattern (e.g. the "attack" pattern in ACE) while open-world event extraction assumes no predefined pattern as in TDT. Thus, the notion of entity in our problem matches a closed world event extraction problem.

Many efforts have been made to address the problem and there are several reviews of the proposed solutions in the general and biomedical domains (Miwa et al., 2014; Xiang and Wang, 2019). A common approach to this task is to first detect the trigger and argument entities, then to predict the relations between them, and finally to detect event modifiers (e.g. negation) and optionally to filter the predicted events. Most works make the assumption that the named entities are already extracted, but the integration of the entire pipeline into a single architecture combined with multi-task learning has shown promising results in recent works (Nguyen and Nguyen, 2019; Trieu et al., 2020).

Overlapping events occur when the same trigger is associated with several arguments for at least two of these relations, as in this example from the Cancer Genetics (CG) task of BioNLP Shared Task 2013 (Pyysalo et al., 2013).



Techniques to cope with it are mainly based on distributing the arguments over each other and checking that the produced events are legal according to the annotation scheme. However, false positives can be generated with examples such as the one in the figure above. Some methods then use an additional classifier to detect these false positives (Björne and Salakoski, 2011, 2013, 2015; Heimonen et al., 2010; Liu et al., 2015; Miwa et al., 2010; Trieu et al., 2020). To our knowledge, these types of events have not been studied in depth. However, they better illustrate the complexity of structured entities that can be found in clinical radiography reports, and in our opinion require further consideration.

2.4.4.3 Named entity attribute detection

Another related task is the one of named entity attributes detection. This task consists in labelling a named entity with different classes. In our case, if we assume that "triggers" are known ("cyst" for example), the simple prediction of their attributes (type, location, size, temporality) as multi-label problem can be enough for some downstream processing. However, it is desirable that the prediction of these attributes be justified when possible by a zone of the text.

The detection and justification of attributes on named entities has not been the subject of specific works, but most existing models that seek to extract them either use a set of rules , or a classifier for each predicted entity. It is conceivable to imagine a system that would explain its predictions by attention scores on the sentence or the joint prediction of the attribute and the beginning and end bounds of its justification.

Nevertheless, this approach would again face the issue of superposed triggers, as in the examples of Section 2.4.1.2, that would prevent the dissociation between the different combinations of attributes. The prediction and distinction of superposed entities of the same type, but with different attributes has not been addressed in the literature, but Sequence-to-Set systems such as DETR (Carion et al., 2020), or its adaptation to texts (Tan et al., 2021) are promising leads.

2.4.4.4 Discontinous named entity recognition

The task of recognizing discontinuous named entities can be interesting as well, as it aims at extracting named entities composed of several segments (or holes in an equivalent way). Several methods have been proposed. Metke and Karimi (2016); Tang et al. (2018, 2013) propose augmenting sequence tagging techniques with new tags. Lu and Roth (2015); Muis and Lu (2017) construct a complex hypergraph of words. Dai et al. (2020) address the problem using transition model. Wang and Lu (2019) transform the problem into a two stages detection: the first one aims at extracting the fragments (or spans) that will compose the entities, and the second one aims at filtering by a classifier among all the possible combinations of these spans which are valid. More recently Li et al. (2021) has also proposed a two stages approach, but detects the combinations of entities by generating a tree between the different segments.

However, this task makes the assumption that all segments of a discontinuous entity are of the same type, which is not our case, and focuses essentially on segments that are close to each other. Moreover, the number of segments is usually limited, e.g., 3 in SemEval 2014 (Pradhan et al., 2014), which allows enumeration of possible combinations unlike our case.

2.4.5 Public annotated corpora

Several datasets have been developed and made publicly available for information extraction from radiology reports. RadCore (Hassanpour and Langlotz, 2016) is a multi-institutional database of radiology reports that contains named entity annotations. However, it does not relate these named entities together. PadChest (Bustos et al., 2020) contains chest radiographs associated with reports labeled according to different radiographic findings, diagnoses, and anatomical locations. Datta et al. (2020b) annotated 2000 chest radiology reports with named entities of spatial location, observation, and several relationships linking them. Recently Jain et al. (2021) released RadGraph which consists of 600 annotated chest radiology reports with spatial location and observation entities following a finer grained scheme than Datta et al. (2020b).

However, clinical reports in these datasets are relatively short and straightforward, with no deep imbrication in their structure. As a result, relations between named entities are mostly found at the sentence level, and not at the document level. Moreover, to our knowledge, there are no datasets consisting of French radiography reports, let alone breast radiography.

2.5 Conclusion

In this chapter, we have discussed the background regarding the three levels of retrieval that interest us in this thesis, namely NER, entity normalization, and structured entities. The following chapters present our work in these three areas.

Chapter 3

Neural architectures for nested named entity recognition in biomedical texts

Contents

In this chapter, we study the named entity recognition task, and more precisely, the nested named entity recognition task. As we will see in this chapter, tagging-based NER methods, i.e., based on token classification, have attractive properties for pipeline systems and noisy data sets. However, it remains a challenge to adapt these models to overlapping entities. To this end, we propose two supervised approaches using neural networks. The first approach uses an auto-regressive tagging model, which iteratively predicts non-overlapping entities in a sentence. The second method is based on a tagging model combined with an exhaustive scoring model.

We will study the impact of input word features on the model's performance and whether a broader context can improve prediction performance when using pretrained contextualized embeddings. We will also study whether the order of the entities impacts the performance of the auto-regressive model. We study the contribution of tagging prediction for the combined model and the gain over an exhaustive scoring model alone. Finally, we will describe a method to improve the performance of each model by ensembling.

The remainder of this chapter is organized as follows. In Section 3.1, we will describe the datasets that we use in our experiments. In Section 3.2, we will describe the preprocessing of the inputs and the features used by our models. We will present a first model, the autoregressive decoder, in Section 3.3, and a second model, the biaffine tagger decoder, in Section 3.4. We present the experiments n Section 3.6, and the discuss the results in Section 3.7. Finally, we close this chapter by a conclusion 3.8.

The source code for the models described in this Chapter is available at the following URL: https://github. com/percevalw/nlstruct.

3.1 Data

In this chapter, we conduct experiments on the two medical named entity datasets DEFT (Cardon et al., 2020) and GENIA (Kim et al., 2003) and the English subset of the a general named entity dataset CoNLL 2003 (Sang and De Meulder, 2003). These datasets have been presented in more detail in Section 2.2.3.

In each cases, we split the training data into 80% for training the model and 20% for the development (validation) set, and train the final model on both the training and development sets.

We have noticed that different versions of the GENIA dataset have been used to evaluate the NER systems. In particular, one of the versions used by Yu et al. (2020), Shen et al. (2021) and Tan et al. (2021) is pre-tokenized in a way that benefit the performance of NER systems (some words are sometimes merged with neighboring punctuations like "-induced," but this is not consistent across samples).

3.2 Text encoding

We start by describing the model used to generate features for each word of the input sequence. These features will then be used by different decoders to produce named entities.

3.2.1 Preprocessing

Sentence segmentation For long documents, it is common first to perform a sentence segmentation. This step has three objectives. The first is to reduce the size of the samples provided to the model in order to reduce the memory impact and speed up the prediction.These effects are all the more important as the models involve operations of quadratic complexity in the size of the sentences.

The second objective is to improve the gradients computed by the model. Indeed, once the corpus is divided into sentences and mixed, each batch can contain more varied samples and lead to less biased gradients. Finally, the presence or absence of an entity in a sentence is generally considered not to depend on the content of the other sentences, or only to a small extent. This hypothesized invariance suggests that we first segment and shuffle the corpus.

Tokenization Our models use two tokenization methods. The first one is the most intuitive and extracts each word from the sentence. We also consider each punctuation as a token in itself. The second tokenization method is the one used by BERT and splits each previously extracted word into subwords (Wu et al., 2016). In the rest of this chapter, we will refer to these subwords as "wordpieces." For each sample, we align the words and wordpieces to jointly use models that operate with each of these tokenization methods.

3.2.2 Features

In our models, the text is encoded as words embeddings in two steps. In the first step, we gather embeddings from various models that we either learn, finetune or leave intact. These word embeddings are then concatenated and forwarded through a multi-layer highway bidirectional LSTM. We describe the overall architecture of the text encoder in Figure 3.1.

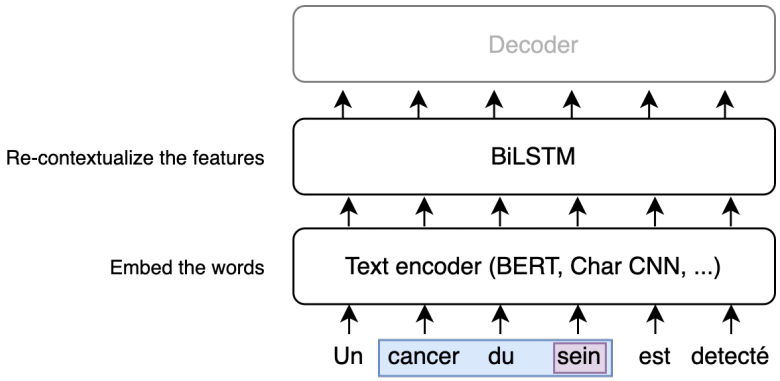


Figure 3.1 Overall architecture of the text encoder. The decoder part of the models is grayed out and will be described in Sections 3.3 and 3.4

.

Pretrained language model features We use the BERT family of pretrained Transformer models. These models use the Wordpiece tokenization algorithm (Wu et al., 2016) and produce one embedding per wordpiece. The parameters of these modules can be frozen or fine-tuned during training. To obtain the embedding of a word, we will evaluate several options: taking the embedding of the first wordpiece, the embedding of the last wordpiece, or the average of all the embedding of each wordpiece. To improve the word embeddings, instead of running the Transformer on each sentence independently, we add neighboring words from the same document until we reach a maximum length. This method of adding context words before running the Transformer is similar to the one of Devlin et al. (2019), Kantor and Globerson (2020), Yu et al. (2020) and further studied by Schweter and Akbik (2020) and Luoma and Pyysalo (2021). Throughout the rest of this chapter, we will call this method "document contextualization." This contextualization should help the model improve its representation of words at the beginning and end of each sequence, especially for short sentences.

Character level features We compute another representation for each word from its characters. Each character in a word is embedded and fed to multiple convolutional neural networks (CNNs) of different kernel sizes. The convolution results of each word are max-pooled and passed through a ReLU. The parameters of these modules are learned from scratch during training.

Context independent word embeddings Finally, we also extract context-free word embeddings with FastText (Bojanowski et al., 2017). These representations are frozen during training in all our experiments. FastText embeddings cannot adapt to their local context. However, they can be pre-computed for every word, regardless of its spelling errors or complexity, because they are computed from character n-grams embeddings.

3.2.3 Recurrent contextualization

All of the previously mentioned representations are concatenated and fed to a bidirectional multi-layer LSTM. The LSTM cell can model local interactions well, which fits our problem since entities often span a few words, and words relevant to the type or boundaries of an entity are often found inside or close to the expression. The output of each layer passes through a sigmoid residual gate, and the output of the last layer composes the word features used by our decoders.

3.3 Auto-regressive decoder

We detail here a first model that handles nested named entity recognition through an auto-regressive mechanism. The prediction occurs in multiple steps. At each step, the bidirectional multi-layer LSTM receives the contextualized embeddings and a list of previously predicted entities (empty list at the first iteration) and produces a list of new entities. The entities predicted at each iteration do not overlap, but all the entities predicted at the end may overlap. This model can be seen as similar to the earlier cascaded model of Alex et al. (2007), but uses a single decoder applied iteratively on the sentence, and is able to recognize overlapping entities of the same label. Figure 3.2 illustrates the architecture of this decoder.

3.3.1 Architecture

The main component of the decoder is a CRF (Lafferty et al., 2001) layer that predicts entities through a multi-type tag scheme (BIOUL or BIO [1]). This multi-type tag scheme can only represent flat entities, which means that the decoder only predicts non overlapping entities at each step. The decoder starts from an empty sequence, in the sense that no entity has already been predicted, and tags each word according to the tag scheme. The sequence of tags is converted into a list of (begin, end, label) entities and added to the set of predicted entities. The decoder repeats this process until no more entities are predicted.

At each step, we need to encode the information about the previously predicted entities to prevent the model from predicting these entities again. We choose to encode each entity as a list of tags on the words that it spans. These tags are embedded into a multidimensional vector space and concatenated with the input features for a given word. In this model, each word is therefore represented by its BERT, FastText and char CNN embeddings, as well as a tag embedding that encodes the entities that were already predicted at the position. This allows the model to reason about what parts of the sentence may still contain other entities.

When multiple previous entities cover the same words, we reduce the tag embeddings at a given position by summing them together. We encode these previous entities in the form of

1. BIO stands for Begin, Inside, Outside and BIOUL for Begin, Inside, Outside, Unary and Last

tags assigned to each token with the BIO, or BIOUL formats (Dai et al., 2015; Ratinov and Roth, 2009). The chosen format will be referred to as the "encoding" tag scheme, in contrast with the "decoding" tag scheme used to decode the entities.

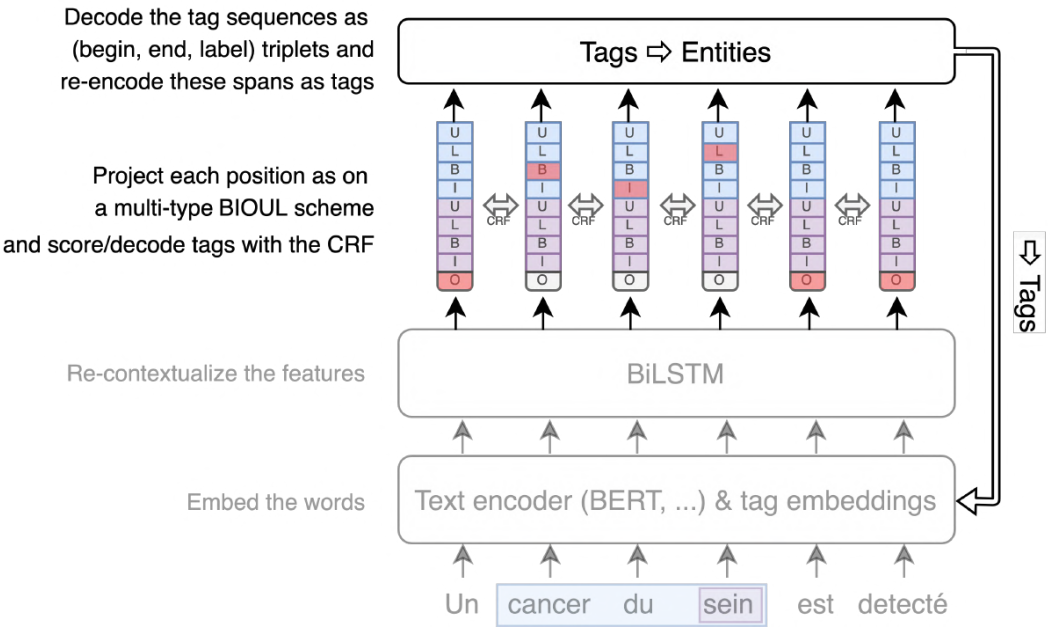


Figure 3.2 Autoregressive decoder. The encoder part of the model is grayed out and was described in Section 3.3. In this example, the model can only predict one of the two nested entities, and chooses the largest one. If the smaller one has not been predicted yet, it will be predicted in a next step.

.

Confidence Additionally, we can compute the confidence given to a (begin, end, label) entity by a BIOUL-CRF model as the product of three probabilities:

— the probability of the first word being a B or U tag of the given label

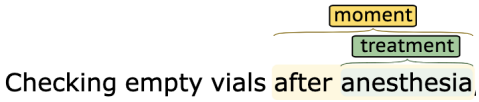
— the probability of the last word being an L or U tag of the given label

— the probability of having no O tag of the label between the first and last words

The first two probabilities can be computed easily by marginalizing the CRF outputs. The last probability can be computed through cumulated sums of log probabilities.

3.3.2 Training

Autoregressive order We proceed in several steps and predict only entities that are not overlapping at each step. However, several permutations, or valid prediction paths, lead to the same list of entities. For two nested statements



(annotations that we call T and H), we can predict T first, then H knowing T, or the opposite, i.e. choose to optimize between two objectives :

— $P(T, H) = P(T, H|T) \times P(T)$

— $P(T, H) = P(T, H|H) \times P(H)$

Two simple strategies are to order the entities by increasing or decreasing length. In the increasing length strategy, the model predicts the small entities first where are overlaps and the larger entities in subsequent steps. This strategy was used by systems like Ju et al. (2018). We will call this order short→large.

In the decreasing length strategy, the model predicts the small entities first where are overlaps and the smaller entities in subsequent steps. We will call this order large→short.

However, the static short→large or large→short strategies may not take advantage of all the inter-dependencies that could make some mentions easier to find when you know the others. An alternative is to rank the entities by decreasing order of confidence. We score each target entity according to the model and greedily build a list of entities starting with the highest confidence scores. For each added entity, we check that it does not overlap with any other entity already in the list, to ensure that the set of entities we will train the model with at this step is flat. By using this ordering strategy, we encourage the model to predict the ones that are easiest first.

Loss We train the autoregressive model by summing the loss for each prediction step. At each step, we run the model and select the entities according to a strategy in those previously described. We convert these entities into tags, and we compute the loss via the linear CRF forward algorithm. The target entities are added to the list of previously predicted entities, and the process is repeated and stopped after no more entities are predicted.

To introduce some noise in the training, we randomly select a subset of the entities in each sentence and mark them as already predicted. This exposes the model to new entities orderings that would not occur if the model had started its prediction from an empty list.

3.3.3 Inference

For each sentence in the corpus, our model starts by predicting the most likely sequence of entities from the input token sequence alone since no mention has already been predicted. Then, we add them to the observed entities list and repeat the prediction until no more entities can be found.

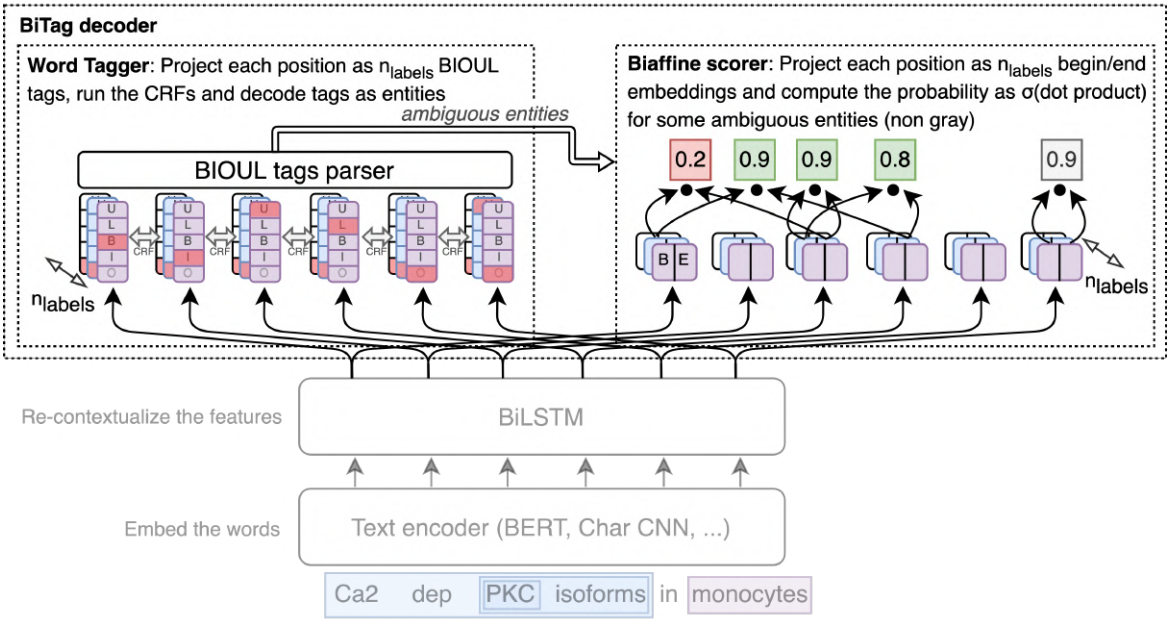


Figure 3.3 Overall architecture of the BiTag decoder. The encoder part of the model is grayed out and was described in Section 3.2.

3.4 Biaffine tagger decoder

3.4.1 Architecture

In this section, we present an other architecture: the Biaffine Tagger decoder (BiTag): a tagging (or sequence-labeling) based-decoder that combines with a biaffine scorer to distinguish between several possible boundaries matching. Unlike the Autoregressive model of Section 3.3, this model does not require multiple prediction steps, and therefore can be easier to integrate into larger architectures. Figure 3.3 illustrates the architecture of this decoder.

Tagger component The main decoder component is a set of CRF (Lafferty et al., 2001) layers that predicts entities through an extension of the BIOUL tag scheme for overlapping entities. Since multiple entity types in a corpus may overlap, we run multiple CRF in parallel, one for each entity label, with five possible tags each. Therefore, each word of the sequence is classified as either a Begin word, Inside word, Outside word, Unary word (a word that is both a Begin word and an End word), or Last word for each label.

Our variant of the BIOUL tag scheme allows overlapping entities of the same label. We generate a sequence of tags for each label according to the following rules: the begin bound of an entity is tagged B, and the end bound of an entity is tagged E. Every word in between is tagged I. If a word is tagged both as a begin and as an end, we tag it as U. If a word is tagged

both as an I and as a B or a E, the bound tag B/E takes precedence. An example of a generated tag sequence for nested entities is shown in Figure 3.4.

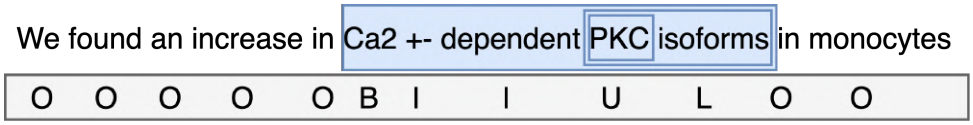


Figure 3.4 BIOUL tagging sequence for the *Protein* label in a GENIA sample

Biaffine scorer In the example of Figure 3.4, the three nested entities are [*Ca2+-dependent PKC isoforms*], [*PKC*] and [*PKC isoforms*]. Multiple combinations of entities can be represented by the same tag sequence. In the above example, we could wrongly predict [*Ca2+-dependent PKC isoforms*] and [*PKC*], or [*Ca2+-dependent PKC isoforms*] and [*Ca2+-dependent PKC*].

We add a biaffine scorer to address this issue and match each begin word with an end word. This decoder is similar to the one of Yu et al. (2020).

The biaffine decoder scores each (begin, end, label) triplet through a biaffine scoring matrix. Each word embedding is projected as n_{label} begin bound embeddings and n_{label} end bound embeddings. Then, each pair of begin and end bounds of a given label is evaluated through a dot product to obtain the score of the (begin, end, label) entity.

3.4.2 Training

We compute the global loss of the model as the sum of the losses of each component. The loss of the tagger module is computed for each label and each sentence via the linear CRF forward algorithm. The target sequence is computed on the fly for each sequence from its list of entities.

The biaffine loss is the binary cross-entropy loss for each (begin, end, label) valid triplet. A triplet is valid when begin $\leq$ end, and the length is below the maximum entity size. In our experiments, we set the maximum size to 40 words.

3.4.3 Inference

Finally, at inference time, we first run the Viterbi decoding algorithm for all labels in a sentence, which gives us a list of tags that we convert into a list of candidate entities. First, for each candidate, we add it to the list of predictions if the biaffine component gives it a score over 0. If a begin or end bound was predicted by the tagger component but was not matched by the biaffine decoder with any other bound, we match it with the end or begin bound that gives the highest scoring entity according to the biaffine module, even if the score if negative. This ensures that each bound predicted by the tagger component is part of an entity. Indeed, we trust the biaffine decoder less on datasets with ambiguous bounds.

3.5 Ensemble models

We propose a method for ensembling multiple instances of each of our models. For each ensemble of models, we average the logits produced by each model before "making a decision," such as running the Viterbi CRF algorithm or classifying a (begin, end, label) triplet. For pipeline models, this generated ensemble output is shared between models at the end of each step. An exemple of the ensembling pipeline before running the Viterbi CRF algorithm is shown in Figure 3.5

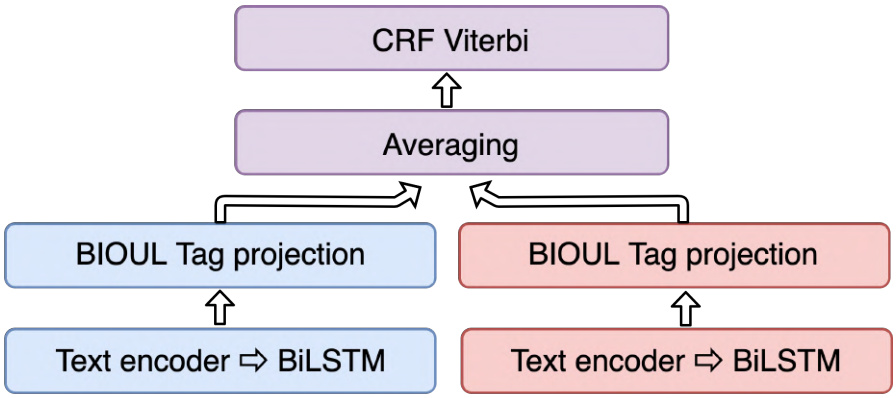


Figure 3.5 Ensemble pipeline before running the Viterbi decoder: each model is run separately until a decision is required. Then all the instances average their logits and the Viterbi CRF algorithm is run. In this Figure, only two models (red and blue) are ensembled.

3.5.1 BiTag model

In the BiTag model, two types of decisions are made. The first decision occurs in the Tagger component and predicts which tag should each word of a sequence be assigned to. All of the instances produce a sequence of BIOUL tag logits for each (word, label) pair in a sentence: we take the mean these logits and run the CRF Viterbi algorithm on these averaged logits. The second decision occurs in the Biaffine component and predicts whether a candidate (begin, end, label) triplet should be predicted or not. We take the mean of the logits and keep the triplets that have an average score over 0.

3.5.2 Autoregressive model

Similarly, for the autoregressive model, we average the logits of each instance before running the Viterbi algorithm. We run the instances in parallel at each prediction step and split the ensemble output between the different models before running the next step. We repeat the process until no more entities are predicted.

3.6 Experiments

We evaluate our models on the DEFT, GENIA and CoNLL datasets. We also perform additional experiments through ablations for the encoder and decoder components on the DEFT and GENIA corpora.

3.6.1 Experimental setup

We run each experiment with 6 different seeds (except for the Ensemble model) and present the average scores. By default, we report the precision, recall and F1 score of the exact metric, and the relaxed half metric corresponds to the retrieval metric where two entities are counted as matching when their word Dice overlap score exceeds 0.5 (see Appendix A). For each model, we optimize the parameters with the Adam optimizer (Kingma and Ba, 2015) without weight decay., over 4000 steps when finetuning BERT and 20000 steps when BERT is frozen. We use two learning rates: the first learning rate, that applies to the pretrained Transformer weights, is initialized at 5×10^{-5} and follows a linear schedule with a 10% warmup, while the second learning rate, for the other parameters, is initialized at 1×10^{-3} and follows a linear decay schedule with no warmup. We selected the hyperparameters by grid search on the development set and trained on both the training and development splits for the GENIA and DEFT datasets. The main hyperparameters are summarized in Table 3.1.

Word features We initialize the Transformer with CamemBERT (Martin et al., 2020) weights for DEFT and BioBERT (Lee et al., 2020) for GENIA and English BERT (Devlin et al., 2019) for CoNLL. We used large (1024) cased versions of these models for our experiments on the test set, and base (768) cased versions of these models for other experiments on the validation set. When finetuning the BERT encoder, we apply Dropout (Srivastava et al., 2014) with a probability of 0.1 in the Transformer layers. Conversely, when the BERT encoder is frozen, no dropout is applied on it. Training is much faster in this setup because the generated embeddings can be cached and reused between epochs. The character embeddings of the character CNN have a size of 50, and are fed to 3 kernels of size 3, 4 and 5. The FastText embeddings are the standard English FastText version for the models trained on CoNLL and GENIA, and the French version for DEFT. The BiLSTM is composed of 3 layers and Dropout is applied on each layer output with a rate of 0.4.

Biaffine tagger The BiTag model bound embeddings size is either 64 for DEFT and 150 for the other models. This distinction was made because of the number of labels that is higher in DEFT (10 labels) than in the other datasets.

Autoregressive model We set the initial observation rate at 0.1. This means that during training, around 10% of the entities are already labelled as predicted, regardless of the autoregressive training order. Following the experiments on the DEFT and GENIA validation sets, we used the short-to-large strategy for the DEFT dataset and the large-to-short strategy for the GENIA dataset. For datasets, such as CoNLL, that does not contain any overlapping features, these three strategies are equivalent.

Ensemble models We evaluate the performance of our ensembling method for each decoder, by training 3 instance of the same model with different seeds, and ensembling these three models using the procedure of Section 3.5.

Maximum sentence/context size	256 wordpieces
Char CNN kernel size	(3, 4, 5)
Char embedding size	8
Char CNN output size	50
FastText size	300
Decoders dropout	0.1
BiLSTM layers	3
BiLSTM dropout	0.4
Biaffine hidden size	64 or 150
Autoregressive pre-observation rate	0.1
Number of steps	20000 if frozen BERT else 4000

Table 3.1 Hyperparameters of the autoregressive and BiTag models

3.6.2 Baselines and ablations

We provide the reported results for several published models in each dataset. Some recent models (Shen et al., 2021; Tan et al., 2021; Yu et al., 2020) were not included in the GENIA comparison (Table 3.2) as they use a non-standard version of the dataset. We compare our methods against the reported results of these works in Table 3.3, on the modified GENIA dataset. We also provide a close re-implementation of the method of Yu et al. (2020) by removing the sequence labelling component from our model, under the name "Biaffine-only". The main differences between the two implementations are that we finetune the pretrained BERT while they freeze it, and that they perform "document contextualization" by re-running a full BERT model for each word of the sentence with a sliding window of size 512, instead of running BERT once for each sentence in our case.

Finally, we provide the performance of the Hugging Face NER re-implementation of (Devlin et al., 2019) (BERT followed by a softmax layer) under the name "BERT + softmax" for each dataset. This model was trained for 4000 steps with the same pretrained weights as our models on a CoNLL formatted (one label per word with the BIO tag scheme) version of each

dataset, as preprocessed by the `ann2conll.py` script of the Github BRAT tools, and the results were exported to the BRAT standoff format for the evaluation.

Input features We perform several feature ablations on the BiTag model.

We study the effect of the BERT document context. More specifically, we only compute the BERT embeddings by running the Transformer on the tokens in the sentence. In contrast, when using Document Context, the neighboring words of a sentence are added as context to the input sequence when running the Transformer.

We also ablate the character CNN representations and the FastText embeddings to estimate the contribution of these features.

Finally, we change the word pooling strategy with the BiTag decoder. Specifically, we evaluate three modes: the "first" mode uses the embedding of the first wordpiece of a word as the word embedding, the "last" mode uses the embedding of the last wordpiece, and the "mean" mode computes the unweighted average of the wordpiece embeddings for each word.

Autoregressive model We study the effect of the autoregressive order on the model performance. We compare three modes: top to bottom, bottom to top, and greedy decoding. These modes differ when choosing between two overlapping entities as to which one the model should first predict.

In the top to bottom mode, we always choose the larger entity first. After learning with this mode, the model should first focus on the large entities and detect smaller ones later.

In bottom to top mode, between two overlapping entities, we always choose the smaller entity first. After learning with this mode, the model should output the small entities first and detect larger entities later.

Finally, we let the model choose the mentions in greedy decoding mode by first selecting the mention with the highest model confidence score. In this setup, the model should output the easiest entities first and the more complex entities later.

Biaffine tagger model We remove the tagger decoder while keeping the biaffine module, such that our decoder is only composed of the biaffine module. This model should be equivalent to the one of Yu et al. (2020).

3.7 Results and discussion

3.7.1 Main results

The results of our systems and the baselines are presented in Tables 3.2, 3.4 and 3.5.

On the GENIA dataset (see Table 3.2), the proposed BiTag model (with finetuning) obtains the exact F1 score of 78.4 and the ensemble model obtains the F1 score of 79.1. The

Autoregressive model obtains a score of 78.3 and its ensemble version reaches 79.0. This is slightly below the reported state-of-the-art results.

An interesting finding comes from the relaxed half metrics, as we observe that exact metric is not always adequate to discriminate between two models. Indeed, for the GENIA dataset, the biaffine tagger and Biaffine-only models obtain very close exact F1 scores (78.4 vs 78.5). However, the BiTag model performs better the biaffine model by +0.5 pt on the relaxed half F1 score. We will expand further on this aspect in Section 3.7.3. Finetuning also shows a greater effect on the half F1 metric (+1.1 pt) than on the exact metric (+0.3 pt).

	P	R	F1	Half F1
Katiyar and Cardie (2018)	78.6	68.2	73.6	
Ju et al. (2018)	78.5	71.3	74.7	
Wang et al. (2018a)			73.9	
Wang and Lu (2018)	77.0	73.3	75.1	
Sohrab and Miwa (2018)	93.2	64.0	77.1	
Lin et al. (2019)	75.8	73.9	74.8	
Shibuya and Hovy (2020)	76.3	74.7	75.5	
Luan et al. (2019)			76.2	
Straková et al. (2019)*			78.3	
Wang et al. (2020)*	80.3	78.3	79.3	
BERT + softmax	77.5	70.4	73.8 (±0.3)	81.7 (±0.1)
Autoregressive large→short	78.9	77.8	78.3 (±0.1)	84.3 (±0.1)
BiTag w/o finetuning	79.3	76.9	78.1 (±0.1)	83.4 (±0.1)
Biaffine-only	78.0	79.0	78.5 (±0.2)	83.8 (±0.1)
BiTag	78.9	77.9	78.4 (±0.1)	84.3 (±0.1)
Autoregressive large→short (ensemble)	80.0	78.0	79.0	85.1
BiTag (ensemble)	80.3	77.9	79.1	85.1

Table 3.2 GENIA test performance. * indicates that the method also uses Flair embeddings (Akbik et al., 2018). Some recent models (Shen et al., 2021; Tan et al., 2021; Yu et al., 2020) were not included in this table, as they use a non-standard version of the dataset.

On the CoNLL English dataset (see Table 3.4), the BiTag model obtains a F1 score of 93.1, and the Autoregressive model obtains a score of 93.0, slightly below the reported state of the art models with the same features. The ensemble versions of each model obtain 93.6 and 93.4 F1 respectively, gaining respectively +0.5 pt for the Autoregressive model, and +0.3 pt for the BiTag model in comparison. The differences between half and exact metrics are much smaller, and all of our models perform broadly on par with each other.

The results of our systems and the baselines are presented in Tables 3.2, 3.4 and 3.5.

On the GENIA dataset (see Table 3.2), the proposed BiTag model (with finetuning) obtains the exact F1 score of 78.4 and the ensemble model obtains the F1 score of 79.1. The Autoregressive model obtains a score of 78.3 and its ensemble version reaches 79.0. This is slightly below the reported state-of-the-art results.

An interesting finding comes from the relaxed half metrics, as we observe that exact metric is not always adequate to discriminate between two models. Indeed, for the GENIA dataset, the biaffine tagger and Biaffine-only models obtain very close exact F1 scores (78.4 vs 78.5). However, the BiTag model performs better the biaffine model by +0.5 pt on the relaxed half F1 score. We will expand further on this aspect in Section 3.7.3. Finetuning also shows a greater effect on the half F1 metric (+1.1 pt) than on the exact metric (+0.3 pt).

	P	R	F1	Half F1
Tan et al. (2021)	82.3	78.7	80.4	
Yu et al. (2020)	81.8	79.3	80.5	
Shen et al. (2021)	80.2	80.9	80.5	
BERT + softmax	79.2	71.1	74.9 (±0.3)	81.8 (±0.3)
Autoregressive large→short	81.4	79.3	80.3 (±0.1)	85.6 (±0.1)
BiTag w/o finetuning	81.6	79.6	80.6 (±0.2)	85.5 (±0.2)
Biaffine-only	80.1	80.5	80.3 (±0.3)	84.8 (±0.3)
BiTag	81.0	79.8	80.4 (±0.3)	85.5 (±0.1)

Table 3.3 Non-standard GENIA test performance, as used by Shen et al. (2021); Tan et al. (2021); Yu et al. (2020)

For reference, we also provide the results of our model on the modified GENIA dataset in Table 3.3. Regarding the Exact F1 performance, the models seem to perform on par with each other and the recent models of Shen et al. (2021); Tan et al. (2021); Yu et al. (2020). However, regarding the relaxed Half F1 measure, both BiTag models (with and without finetuning) outperform the Biaffine-only model by an average of 0.7 pt.

	P	R	F1	Half F1
Klein et al. (2003)	91.4	91.9	91.6	
Lample et al. (2016)			90.9	
Strubell et al. (2017)			90.7	
Devlin et al. (2019)			92.8	
Straková et al. (2019)			93.4	
Yu et al. (2020)	93.7	93.3	93.5	
BERT + softmax	90.2	92.0	91.1 (±0.2)	92.8 (±0.2)
Autoregressive	92.9	93.1	93.0 (±0.2)	94.2 (±0.2)
BiTag w/o finetuning	92.6	93.1	92.8 (±0.1)	94.1 (±0.1)
Biaffine-only	92.9	92.8	92.8 (±0.2)	94.0 (±0.1)
BiTag	93.0	93.2	93.1 (±0.2)	94.3 (±0.2)
Autoregressive (ensemble)	93.7	93.5	93.6	94.5
BiTag (ensemble)	93.3	93.6	93.4	94.7

Table 3.4 CoNLL English test performance

On the DEFT task 3.1 (see Table 3.5), the BiTag model obtains the best F1 result of 77.2 (with the exact delimitation of mentions), and a F1 measure of 67.6 on the DEFT task 3.2.

	DEFT 3.1 (exact)			DEFT 3.2 (exact)			Overall (F1)	
	P	R	F1	P	R	F1	Exact	Half
Copara et al. (2020)			74.4			62.3	70.7	
Copara et al. (2020) (ensemble)			75.5			66.0	72.6	
BERT + softmax	67.8	31.2	42.7 (±0.6)	62.2	63.9	63.0 (±0.9)	50.4 (±0.5)	60.5 (±0.3)
Autoregressive short→large	78.7	75.9	77.3 (±0.2)	66.8	67.1	66.9 (±0.8)	74.1 (±0.3)	84.5 (±0.1)
BiTag w/o finetuning	78.8	75.7	77.2 (±0.5)	66.7	66.5	66.6 (±0.6)	73.9 (±0.3)	83.6 (±0.2)
Biaffine only	76.2	76.4	76.3 (±0.4)	66.6	67.7	67.1 (±1.4)	73.5 (±0.6)	82.1 (±0.3)
BiTag	78.7	75.9	77.2 (±0.4)	67.5	67.6	67.6 (±1.2)	74.3 (±0.4)	84.3 (±0.1)
Autoregressive short→large (ensemble)	80.3	75.9	78.5	70.0	68.9	69.4	75.4	85.2
BiTag (ensemble)	80.0	76.7	78.3	68.5	68.7	68.6	75.3	85.4

Table 3.5 DEFT test performance

The ensemble BiTag model reaches 78.5 on the 3.1 task and 68.6 on the 3.2 task. We observe that the different models have a large variance on the strict F1 score, and a lower one on the relaxed half F1 score, and that the relaxed score is almost 10 pt higher than the strict one. This could be explained by a high noise in the entities boundaries annotation. The discrepancy between the exact F1 score and the half F1 score is even stronger in this case: the BiTag model gains "only" $+0.8$ pt on the exact metric, but $+2.2$ pt on the half metric in comparison to the Biaffine-only model.

For each data set, we observe that the BERT + softmax model performs worse. On the CoNLL dataset, it reaches a score of 91.1 Exact F1, below the reported results of (Devlin et al., 2019) (92.8). This difference might be caused by a difference between the hyperparameters selections (we used the default hyperparameters of the Hugging Face `run_ner.py` script), or the fact that we did not set a maximum sequence size that can affect the outputs of the commonly used `seqeval` tool. The performance gap with this baseline much larger on the other datasets containing nested entities, between 15 and 25pt on GENIA and DEFT, which is due to the impossibility of predicting overlapping entities using a multi-class BIO tag scheme.

The better results of the ensemble models on each dataset confirm the common idea that ensembling is an effective way to boost the performance of a model. Similarly, finetuning the BERT model seems to improve the performance of the models to varying degrees depending on the domain and language. Overall, the gaps in performance between the two proposed models (BiTag and Autoregressive) are slim, despite the differences in design between each. This could be explained by the fact that each model is based on a sequence-labelling mechanism, and this suggests that features have a more important role, since they are the same in both our proposed models.

3.7.2 Auto-regressive model ablations

3.7.2.1 Tag scheme

We analyze the performance of two common tag schemes: BIO (Begin, Inside, Outside) and BIOUL (BIO with Unary and Last tags) to encode observed (i.e. previously predicted) entities. Results can be found in Table 3.6. As a decoding scheme, the BIOUL tag scheme shows better overall results than the BIO scheme. This conclusion is similar to what others (Lample et al., 2016; Ratinov and Roth, 2009) have observed for flat named entity recognition. Moreover, as an encoding scheme, that is to encode previously predicted entities as features to the subsquent prediction steps, the BIOUL "encoding" tag scheme's also shows better results. Overall, we conclude that a linear representation of entities as a tag sequence benefits from the added expressiveness of the BIOUL scheme.

	BIO encoding	BIOUL encoding
BIO decoding	70.1	71.3
BIOUL decoding	70.5	**71.6**

Table 3.6 Performance of the BIO and BIOUL reading and writing tag schemes on the DEFT validation dataset.

3.7.2.2 Autoregressive learning order

From Table 3.7 we can observe that the short-to-large training order obtains the highest performance on the DEFT validation splits, but the large-to-short depth training order obtains the highest performance on the GENIA dataset. We did not reach the same conclusion in a previous work (Wajsbürt et al., 2021b) using a variant of the model architecture for which we observed that the short-to-large strategy obtained the best result on both datasets. On the DEFT dataset, where entities can be quite long, we hypothesize that learning to detect the smallest, and often easier, entities first leads the model to learn how to compose new entities from small entities. On the other hand, learning to predict large, and often more difficult, mentions first, must lead the model to overfit on these large mentions and fail to recover smaller nested mentions when the largest ones are wrongly predicted. On the GENIA dataset, the large-to-short strategy might perform better due to the different average size of entities. These inconsistent observations between the two datasets could therefore indicate differences in entities distribution between each of them and/or highlight an excessive sensitivity of the autoregressive model to these differences

	DEFT		GENIA	
	Exact	Half	Exact	Half
large $\rightarrow$ short	70.5	79.7	**79.5**	**85.2**
greedy	71.1	80.3	79.2	**85.2**
short $\rightarrow$ large	**71.6**	**80.6**	78.7	85.0

Table 3.7 F1 score of the autoregressive ordering strategies experiments on the DEFT and GENIA validation datasets

3.7.3 Biaffine-tagger model ablation

We remove the tagger component of the biaffine tagger model and only rely on the biaffine scorer to extract spans. We evaluate the effect of this ablation on the DEFT and GENIA validation datasets. In this setup, the model is similar to the one of Yu et al. (2020), with the exception of the BERT embedding computation, for which we did not replicate their expensive sliding window mechanism with a stride of 1. On the DEFT validation dataset, the strict performance is not significantly affected and increases by $+0.2$ pt and decreases by -0.1 pt

	DEFT		GENIA	
	Exact	Half	Exact	Half
base	71.4	80.9	78.9	84.5
− Tagging	71.2 (+0.2)	79.2 (−1.7)	78.8 (−0.1)	83.5 (−1.0)
− Doc context	70.6 (−0.8)	80.2 (−0.7)	78.6 (−0.3)	85.0 (−0.2)
− Char CNN	71.0 (−0.4)	80.2 (−0.7)	78.8 (−0.1)	84.4 (−0.1)
− FastText	71.8 (+0.4)	81.1 (+0.2)	78.8 (−0.1)	84.4 (−0.1)
+ Finetuning	73.3 (+1.9)	82.4 (+1.5)	78.9 (+0.0)	84.5 (+0.0)

Table 3.8 F1 score of the ablation experiments on the DEFT and GENIA validation datasets for the Biaffine Tagger. Every experiment was averaged on 6 different seeds

on the GENIA validation dataset. However, the effect on half performance is significant as the model loses -1.7 pt on the DEFT dataset and -1.0 pt on the GENIA dataset.

This type of discrepancy can be explained by the presence of entities with ill-defined bounds. The tagger model confidently labels words inside an entity where there is little ambiguity and hesitates on entity boundaries for such entities. On the contrary, the Biaffine-only model is likely to give too low a score to each pair of start/end bounds and predict no entity. Both models fail to predict the entity exactly, but the tagger model predicts some of its words. It may be more valuable for downstream tasks (like the model of Chapter 5) to predict imperfect entities sometimes than perfect entities or nothing.

3.7.4 Features ablations

Document context We remove the "Document Context" described in Section 3.2.2 and evaluate the model on the GENIA and DEFT validation sets. From Table 3.8, we can see that the document context contributes a lot to the performance of the model and removing it leads to -0.7 pt loss on the DEFT exact metric and -0.3 pt loss on the GENIA exact metric. In this setup, each sentence is contextualized on both sides. We hypothesize that this contextualization benefits the model because BERT has been pre-trained with large sentences (between 128 and 512 tokens), and therefore should have a better representation power for tokens in long sentences.

Character embeddings We ablate the character embeddings features and observe from Table 3.8 that these features have a positive effect on the performance for exact and half metrics, and contribute to up to $+0.4$ pt of the exact performance on the DEFT dataset and $+0.1$ pt on the GENIA dataset. Sub-word embeddings have been shown to perform poorly on tasks that require a reasoning on the character level (Wallace et al., 2019). Such ability can be necessary for tasks that involve accurate number representations or acronym detection, and could therefore benefit named entity recognition. Likewise, the pretrained FastText embeddings for English and French were trained with a n-gram size of 5 and are fixed during

training. Thus, they may not offer a representation that enables the model to reason on shorter n-grams. GENIA contains a lot of DNA and RNA related acronyms, which could explain that it gains more from the character embeddings than the DEFT dataset. We conclude that character embeddings offer a useful representation for named entity recognition.

FastText embeddings We remove English FastText embeddings for the GENIA dataset and French FastText embeddings for the DEFT dataset. On the GENIA dataset, these features have a positive contribution of 0.1 pt of the model exact performance, and on the DEFT dataset, these features have a negative contribution of -0.4 pt of the model exact performance. Overall, these differences are slim, and this mixed effect could be explained by the differences of language, domain or size between the two corpora.

Wordpiece pooling Table 3.9 shows that the "mean" wordpiece pooling obtains a better performance than the "first" and "last" pooling strategies. This suggests that every wordpiece of a word contains information that is relevant to the NER task, rather than only a specific word such as the first or the last one. This superiority of the mean pooling also holds when BERT is fine-tuned. However, it is less significant, which suggests that BERT is able to learn to gather the required information of all the wordpieces of a word in the embedding of the first or the last.

		DEFT	GENIA
Frozen BERT	first	71.7	80.0
	last	72.0	80.1
	mean	**73.0**	**80.5**
Finetuned BERT	first	74.4	80.6
	last	74.2	80.6
	mean	**74.6**	**80.7**

Table 3.9 Wordpiece pooling ablation

3.8 Conclusion

In this chapter, we addressed the task of nested named entity recognition and proposed two approaches. We have compared these models with each other and with state-of-the-art models. We have highlighted a divergence between the strict and relaxed metrics, which should not be overlooked when choosing NER models. Indeed, this difference seems to be all the more important when the dataset is small and contains entities with ambiguous start/end bounds. We have also provided insight into the behavior of the decoders and the contribution of the input features. We have shown that finetuning BERT improves model performance, but more importantly, preserving the context of sentences before running BERT improves

the performance significantly. We also observed that the autoregressive order impacts the performance of the layered named entity recognition model and that predicting short entities first and large ones later gives the best results. Finally, we show that the simpler biaffine tagger model achieves the best overall results and that its Biaffine-only counterpart performs worse on relaxed metrics.

In the next chapter, we will focus on the task of normalizing named medical entities.

Chapter 4

A large scale neuronal classification approach for multilingual medical entity normalization

Contents

In this chapter, we focus on the normalization of medical named entities. More specifically, we address the task of medical entity normalization in non-English languages for large medical lexicons containing hundreds of thousands of concepts, with few to no annotated samples.

Our objective is to match a given named entity with a concept in a terminology, as illustrated in Figure 4.1. We assume that the named entities have already been extracted and may have been labelled with a semantic group. Possible methods for medical named entity recognition have been detailed in Chapter 3.

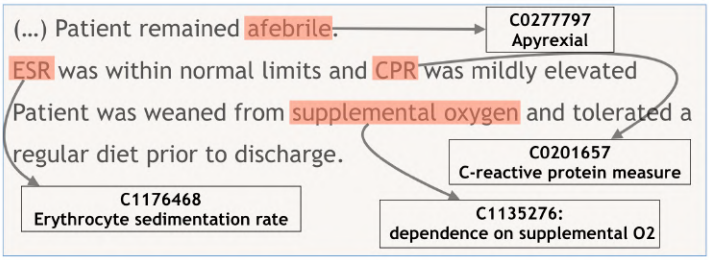


Figure 4.1 Example of medical entity normalization

We take advantage of the multilingual nature of available terminologies and embedding models to improve concept normalization in non-English languages without translation nor direct supervision. We chose to consider the task as a standard classification task amongst concepts, meaning that we only encode concepts of the target terminology, rather than their synonyms, into fixed-length representations that can be stored and even indexed to accelerate lookup at inference time.

This chapter is organized as follows. In Section 4.1 we describe the corpora and terminologies used to train and evaluate our model. In Section 4.2 we describe the neural network model architecture. In Section 4.3 we describe the method used to perform the training and the inference on new data. We present the experiments n Section 4.4, and the discuss the results in Section 4.5. Finally, we close this chapter by a conclusion.

The source code for the model described in this chapter is available at the following URL: https://github.com/percevalw/deep_multilingual_normalization.

4.1 Data

In this our experiments, we focus on normalizing terms in the Quaero FrenchMed corpus (Névéol et al., 2014) and the Mantra corpus (Kors et al., 2015). We have described both datasets, and the corresponding UMLS and Mantra vocabularies in Section 2.3.1, and will briefly review some key aspects of these resources.

Corpus		Mentions	Unique mentions	CUIs	French CUI %
EMEA 2015	train*	2695	923	650	67
	test**	2260	756	525	70
Medline 2015	train*	2994	2296	1860	77
	test**	2977	2288	1847	76
EMEA 2016	train*	2695	923	650	67
	dev**	2260	756	525	70
	test	2204	658	474	62
Medline 2016	train*	2994	2296	1860	77
	dev**	2977	2288	1847	76
	test	3103	2390	1909	79

Table 4.1 Statistics of the Quaero corpus. In each EMEA and Medline split, * and ** denote identical sets of documents between the 2015 and 2016 versions of the corpus

4.1.1 Quaero

The Quaero FrenchMed corpus contains two sets of documents, Medline article search titles and EMEA drug records, annotated with concepts from the 2014AB version of UMLS in 10 semantic types. Since two versions of this dataset were proposed in 2015 and then 2016 (the latter version proposing a new test set), we will evaluate our method on each version. Also, in order to ensure a fair comparison with the other systems published on this benchmark, we use the 2014AB version of UMLS, unless otherwise mentioned. Each annotated entity has an associated semantic type that can be used to improve normalization predictions. Table 4.1 presents general corpus statistics including the number of annotated mentions (i.e., text spans linked to UMLS concepts within the documents), the number of unique mentions, the number of unique concept CUIs, as well as the rate of mentions in each corpus that are linked to a concept with at least one synonym in French in the terminology. Note that very few mentions are annotated with more than one CUI in the corpora.

We have described the UMLS in Section 2.3.1. We will call the UMLS Bilingual subset the set of concepts that have a synonym in both French and English. We built a subset, that we will call "English 5 sources", of the UMLS with terms from five CHV, SNOMEDCT_US, MTH, NCI, or MeSH terminologies. We chose these terminologies because they cover 96% of the labels in the annotated training corpus, without exceeding a million labels. Table 4.3 shows statistics on the number of concepts and synonyms in English and French, for the versions 2014AB and 2019AB, both used in this work.

4.1.2 Mantra

As mentioned in Section 2.3.1, the Mantra corpus (Kors et al., 2015) consists of 1450 annotated with concepts of the Mantra terminology, in five different languages: English, Spanish, French, German and Dutch. While the English language has synonyms for every

Language	Docs	Mentions	Unique mentions	Unique CUI	Language coverage %
English	550	1963	1366	1301	100.0
Spanish	200	756	522	550	93.9
French	250	1052	729	710	68.5
German	250	1082	751	729	68.4
Dutch	200	677	481	490	64.3

Table 4.2 Statistics of the Mantra corpus

Terminology	Subset	#synonyms	#concepts	#synonyms/#concept
UMLS 2014AB	English	5,772,518	2,528,878	2.28
	English 5 sources	2,298,600	766,548	3.00
	French	179,992	88,985	2.02
	Bilingual	544,383	88,911	6.12
UMLS 2019AB	English	9,187,793	4,258,236	2.16
	English 5 sources	3,055,453	968,467	3.15
	French	374,144	154,362	2.42
	Bilingual	903,098	154,307	5.85
Mantra	English	2,030,891	591,665	3.43
	Spanish	750,740	309,600	2.42
	French	138,990	67,743	2.05
	German	116,338	65,974	1.76
	Dutch	127,951	60,241	2.12
	Overall	3,164,910	591,918	5.35

Table 4.3 UMLS and Mantra terminologies statistics. The UMLS Bilingual subset is the set of concepts having synonyms in both English and French.

concept that appears in the terminology, other languages do not and coverage drops as low as 64.3% for the Dutch entities, as illustrated in Table 4.2. Most importantly, being much smaller than the Quaero corpus, the Mantra corpus does not contain a training set and only consists of test samples. It is therefore not possible to perform any supervised learning on this dataset. Moreover, unlike the Quaero dataset, the entities are not labeled with a semantic group: only the text can be used to identify the concept of an entity. Table 4.2 shows statistics on the number of concepts and entities, as well as the percentage of concepts in each language split that have at least one synonym of the same language in the Mantra terminology.

4.2 Model overview

We cast the normalization problem as a classification task. $C = \{c\}$ is the set of all concepts c (i.e., concepts to predict) identified by their CUI. Each concept is associated with one semantic group G_c, with very few exceptions (Bodenreider, 2004). We denote the set of all concepts in a semantic group g as C_g. An entity m is a phrase in a textual document referring to a concept.

In this work, we consider these mentions to be already available and each labeled with a semantic group g_m. The set of synonyms that share a same concept c is called a synset. For example, the concept C0678222 contains the synonyms "breast cancer", "breast carcinoma", "carcinoma of the breast", is associated with the semantic type "Neoplastic Process" and is therefore in the semantic group "DISO" (Disorders). Given a french term "cancer du sein" extracted from a document and pre-labelled with the "DISO" semantic group, our goal will be to correctly map it to the C0678222 concept. Given a dataset, $D = [m_1, m_2, ..., m_n]$ we build a CUI classifier, i.e to learn a probability distribution P to predict the concept of each mention $m \in D$:

$$c^* = \text{argmax} P(c|m; \theta, H_g) \tag{4.1}$$

where θ represents the parameters of the encoder (detailed below), which goal is to map a mention to a dense vector space, and H_g represents the embeddings of the concepts in this space, that have the same semantic group g_m as the mention.

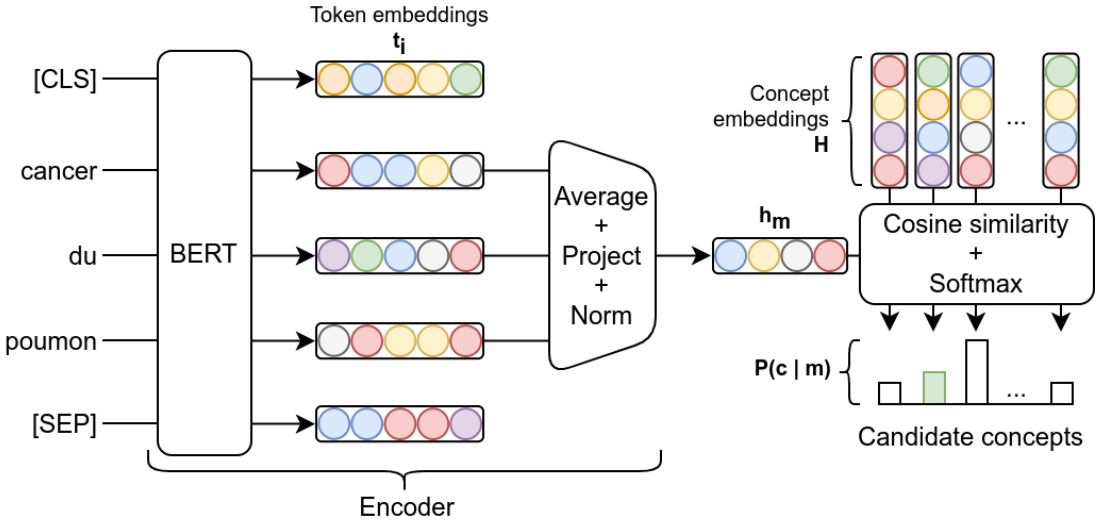


Figure 4.2 Model overview. In the two step training (see Section 4.3.2), candidate concepts (bottom right of the figure) of step 1 are those of the terminology subset; during step 2, candidate concepts are the top candidates

Our model is a classification model built on top of a pretrained Transformer (Vaswani et al., 2017). We call this model MLNorm (for MultiLingual Normalization). The model is described in this section and illustrated in Figure 4.2. The mentions are first tokenized into wordpieces (Wu et al., 2016) and fed into a pretrained BERT encoder to obtain contextualized representations t_i for each token.

$$(t_i) = \text{BERT}(m) \tag{4.2}$$

These contextualized token representations are then averaged across each mention as t_m, without the first [CLS] and last [SEP] special tokens

$$t_m = \frac{1}{l-2} \sum_{i \in [1, l-1]} t_i \tag{4.3}$$

We then perform a projection into a lower dimension embedding to reduce the model size, apply a ReLU function and normalize the result with batch normalization. This leads to a mention embedding h_m.

$$h_m = \text{BN}_{\mu,\sigma} \left[\text{ReLU}(W \cdot t_m + b)\right] \tag{4.4}$$

where $\text{BN}_{\mu,\sigma}$ is the batch normalization layer with mean μ and variance σ, and W and b are the projection weights and bias respectively. Finally, we classify each mention by computing the cosine similarity between its representation and the embedding of the concepts in the semantic group of the mention. Following Wang et al. (2018b) we multiply the similarity by a hyperparameter s. We obtain concept probabilities by applying the softmax function on these scores.

$$P(c|m;\theta;H) = \frac{e^{s \cdot \text{cosine}(h_m, H_c)}}{\sum_{k \in C_g} e^{s \cdot \text{cosine}(h_m, H_k)}} \tag{4.5}$$

where

$$\text{cosine}(h_m, H_c) = \frac{h_m}{||h_m||} \frac{H_c}{||H_c||}$$

H_c is the embedding of the gold concept

H_k is the embedding of a concept in the semantic group C_g of c

$\theta = \{\mu, \sigma, W, b, \text{BERT}\}$

4.3 Model training and inference

We now describe the procedure to train this model and perform predictions with it.

Training our model can be done by learning/finetuning the parameters of the encoder, and all the concepts at the same time, as a standard classification model. In this setup, we iterate through mini-batches of synonyms and classify each synonym against the set of all possible concepts. However, the number of concepts can be very large (up to almost a million in our experiments), and affect both the required computational time required and available memory for training. We discuss two procedures to reduce this computational burden.

4.3.1 Top candidate sampling

A first method, illustrated in Figure 4.3 consists in filtering the set of concepts that are going to be included in the softmax computation, by only keeping those that are the most probable, and the gold concept.

For each synonym, we will refer to its k most likely concepts as its k top candidates.

$$\text{top\_candidates} = \text{top}_c^k P(c|m, \theta, H) \tag{4.6}$$

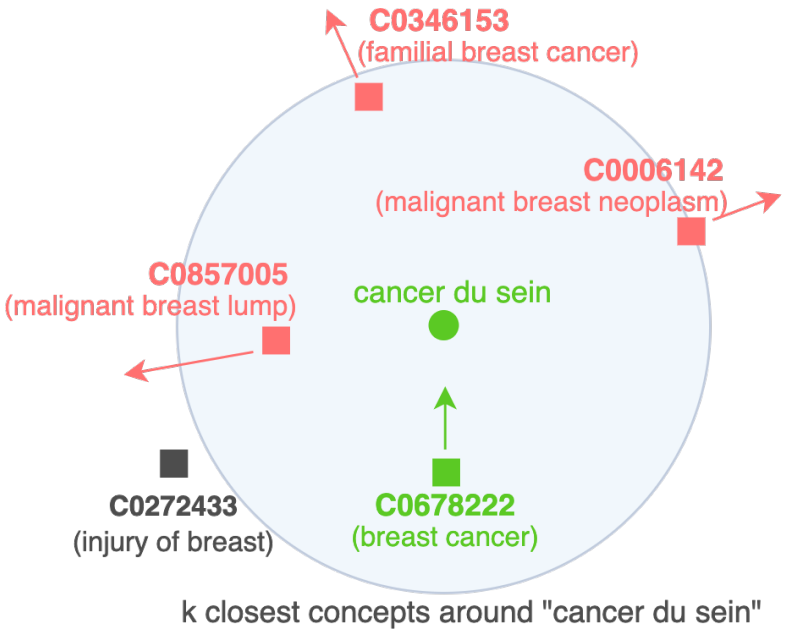


Figure 4.3 Overview of the local concept embedding learning. For each synonym (green dot), we compute its k closest concepts neighbors (squares in the blue disk). Only these concept neighbors will be updated (arrows)

Indeed, for each mention, most of the concepts have a near-zero probability and are therefore not updated during the optimization. Using this method, we only have to compute gradients for a relevant subset of the concept embeddings, thus enabling a faster and more memory-efficient training. Since a batch consists of multiple samples, each sample lists its own most likely concepts, and the set of concepts that will be optimized in a given batch is the union of all. This relates to softmax sampling methods (Jean et al., 2015) with synonym dependent hard negatives (Schroff et al., 2015).

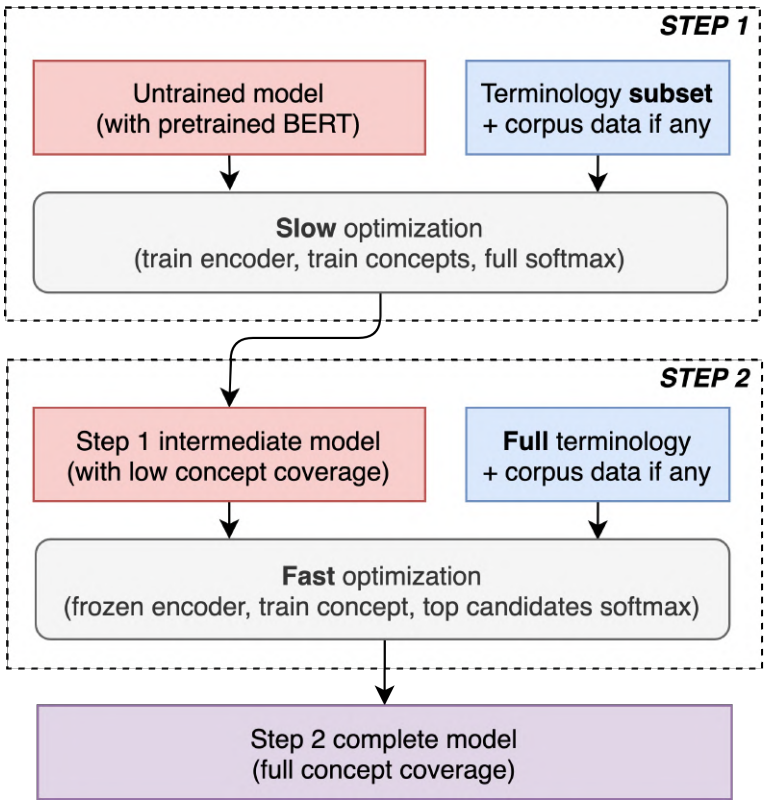


Figure 4.4 Two steps training procedure

4.3.2 Two steps training

We propose to include the previous mechanism in a two step training procedure, illustrated in Figure 4.4. Intuitively, the role of the encoder is to group synonyms such that multiple synonyms of a same concept, maybe in different languages, are projected to the same location in the embedding space. The finetuning of BERT and learning of concept representations is time-consuming and computationally expensive. Moreover, the inclusion of all concepts might not be required to learn this behavior. Therefore, we suggest to start by training the encoder on a subset of synonyms and concepts, with full softmax computation. To choose an adequate subset, we keep bilingual concepts, that have a synonym in at least English and another language (French in the case of Quaero), to focus our training on the multilingual capacity of the model. This leads to a system called S_1, limited to predicting only concepts having French synonyms.

In a second step, we freeze the encoder (Transformer + projection parameters) and train the representation of all the concepts (not only those of the initial subset) with the top-k softmax computation. The idea is that enough synonyms were seen during the first step to ensure that the encoder produces adequate medical entities representations. We now only need to add the missing concepts to the model. The encoder being frozen, the embeddings of

the synonyms stay the same during the second step, and we can efficiently compute the indices of the top candidates before starting the optimization. For each concept, we re-initialize its embedding as the sum of its synset's representations:

$$H_c^{\text{mean}} = \sum m \in \text{synset}(c) h_m \tag{4.7}$$

and use these representations to compute and store the k top candidates for each synonym.

4.3.3 Prediction

At inference time, a mention is tokenized and passed into the encoder and the classifier. We subset the candidate concepts to only keep those that are in the semantic group of the mention if given (only for Quaero entities in our experiments). Finally, we apply a threshold to remove all the predictions that have a low probability. This filtering is required because not all mentions can be mapped to a concept: in our experiments, 4% of the concepts that occurred in the training set where dropped as described in Section 4.1.1. We expect that the filtering will leave out the entities with missing concepts.

4.4 Experiments

We now present and discuss the results obtained by the system on the Quaero and Mantra datasets.

We report our main results on the test datasets from the Quaero FrenchMed 2015 and 2016 challenges, on the Mantra dataset and the results of our additional experiments, using the traditional metrics precision, recall and F1-measure. We also give some predictions of the distantly supervised (trained without the Quaero training set) model in Table 4.6.

The method was evaluated through two main sets of experiments that we call "distantly supervised" and "supervised." In the "distantly supervised" setup, we used only distant supervision from the UMLS, and no direct concept supervision from the available, labeled samples from Quaero. Since the Mantra corpus does not contain a training set, the models that we evaluate on this dataset also fall in the "distantly supervised" category and were only trained with the (synonym, concept) pairs from the Mantra terminology. These systems do not suffer from any potential bias related to the specificities of the corpus and do not benefit from the redundancy of mentions in labeled data sets.

In the "supervised" setup, we augment the training (synonym, concept) pairs with mentions and labels from the Quaero Medline and Quaero EMEA training sets, thus enabling comparison with state-of-the-art supervised approaches on the Quaero dataset. Despite being annotated with concepts from all the UMLS 2014 AB version, we restricted the concepts used in our

Quaero experiments to the EN5 subset (see Table 4.3), because of its good coverage of the corpus and reasonable size.

4.4.1 Experimental setup

We chose the hyperparameters by selecting the best-performing values on the training set of Quaero in the distant supervision setting. We kept the same hyperparameters to train the Mantra models. We run our models on a 20 Go Tesla P40 GPU, except the 1-step experiment which required a 30Go Tesla V100 GPU.

As a result from the hyperparameter search described above, the token embeddings space of size 768 is projected into a space of size 350, the cosine similarity scaling parameter s is of 20, both dropout rates for the transformer and the projection layer are set to 0.2. We set the batch size to 128 and the maximum synonym wordpiece count to 100. We used two different learning rates, lr_{BERT} for the pretrained transformer, lr_{main} for the concept embeddings and projection layer. During the training, we vary the learning rates using two schedules. Following Sun et al. (2019), we used a slanted triangular learning rate lr_{BERT} for BERT with a warm-up phase of 10% of the total number of training steps. We keep the learning rate lr_{main} constant during the warm-up phase and linearly decay it for the rest of the training. We set the maximum learning rates to lr_{BERT}=2e-5 and lr_{main} = 8e-3. We used Adam with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. During the second step of the two-steps training, we preselect the k = 100 highest scoring concepts for each synonym. Unless mentionned otherwise, we perform the step 1 training for 15 epochs and the step 2 for 5 epochs in the 2-step setup, the probability threshold is set to 0.1 and the pretrained Transformer is the multilingual BERT (bert-base-multilingual-uncased in the Huggingface library) (Devlin et al., 2019).

4.4.2 Baselines and ablations

Baselines We compare our system against the following baselines:

— the top ranked systems of respectively CLEF 2015 (Afzal et al., 2015) and CLEF 2016 (Cabot et al., 2016), on the same exact task of normalization from gold-standard mentions. The CLEF 2015 winning team (Afzal et al., 2015) first augments the French UMLS by translating a subset of the English UMLS concepts encountered in Medline abstracts, using Google Translate. This terminology is then queried by a rule-based text indexer. The CLEF 2016 winning team (Cabot et al., 2016) relies on their ECMT indexer which performs bag of words concept matching at the sentence level and integrates up to 13 terminologies partially or totally translated into French.

— the best-performing system, to the best of our knowledge (Roller et al., 2018) on Quaero and Mantra. In this work, the authors first train a local LSTM-based French to English translator on synonym pairs from the UMLS and other general domain sources. The

French and English terminologies are then indexed and searched using Apache Solr through exact and fuzzy matching rules.

We also performed a range of ablation studies and additional experiments on the distantly supervised setup, in order to estimate the impact of our different choices.

Impact of the two-step procedure We trained our system in one step with full softmax instead of two steps, using all the synonyms (French and English from EN5), and evaluated the model on the Quaero 2015 dataset. This is a much more time- and memory-consuming experiment that will allow us to estimate the trade-off between cost and quality.

Impact of the pretrained embeddings We compare the performance on Quaero, using different BERT embeddings either trained on French data only (CamemBERT (Martin et al., 2020), model camembert-base-uncased) or English-only (model bert-base-uncased), or multiple languages (bert-base-multilingual-uncased), in order to evaluate the contribution of the multilingual embeddings.

Impact of translating entities Since the system from Roller et al. (2018), based on machine translation + English-only normalization, is quite different from our own system, we also experimented on the Quaero dataset with a machine translation approach combined with our classifier. This allows a fairer comparison between our multilingual learning approach and a translation-based approach. For this purpose, we translated all UMLS French terms with a state-of-the-art pretrained (opus-mt-fr-en) translation system (Tiedemann and Thottingal, 2020) built with MarianMT (Junczys-Dowmunt et al., 2015) and trained on the OPUS bitext repository corpus (Tiedemann, 2012). We then trained our model with all original-English and translated-English terms. We called this strong baseline BERT-MT (using the English BERT) and mBERT-MT (using the multilingual BERT).

Impact of more French terms (UMLS2014AB vs. UMLS2019AB) We present an experiment using the 2019AB version of the UMLS, containing 154k concepts with French synonyms instead of 89k in the 2014AB version. With this system (UMLS2019), we aim at showing the impact of adding new French synonyms to the terminology used for distant supervision.

Impact of the training language We evaluated the impact that the training data language has on the performance of our system. To do so, we only trained our distantly supervised system on the bilingual UMLS concepts (French and English) and evaluated it on the Quaero 2015 corpus. This filtering was done to train our experiments with the same number of concepts, and mitigate the errors that occur due to missing concepts in the training data. Because the number of synonyms is lower, we trained FR-only model, EN-only model and

FR+EN model longer for 30, 20 and 20 epochs respectively. We use a probability threshold of 0.5 since more entities have concepts that are not seen during training. Moreover, due to the small number of concepts in this configuration, the two-step training was not necessary.

— FR/EN: trained with synonyms of bilingual UMLS concepts

— FR-only: only the French synonyms of these bilingual concepts

— EN-only: only the English synonyms of these bilingual concepts

On the Mantra dataset, we evaluated the effect of training the system with various languages combinations. More specifically, we trained 6 systems with subsets of the Mantra terminology, containing either:

— the synonyms in all languages (Multilingual)

— only the English synonyms (ENG)

— the English and the French synonyms (ENG + FRE)

— the English and the Spanish synonyms (ENG + SPA)

— the English and the German synonyms (ENG + GER)

— the English and the Dutch synonyms (ENG + DUT)

We report the performance on each language in the Mantra dataset for all of these models.

4.5 Results and discussion

4.5.1 Main results

On the Quaero corpus, our distantly supervised system obtains very good results without concept-labeled training data (Table 4.4). It even reaches a slightly higher performance than the best system published so far (Roller et al., 2018) on the corpus MEDLINE 2015 (F1=73.7 vs. 73.6) that used the Quaero training set. It also outperforms all participants of the 2016 edition. Note that CLEF campaigns provide scores on both end-to-end task (named entity + normalization) and normalization-only task; similarly to Roller et al. (2018), we compare to the latter. The much higher term redundancy can explain the better score of supervised systems on EMEA corpus (e.g., F1=83.5 and 73.4 on 2015 and 2016 for Roller et al. (2018) vs. resp. 76.5 and 72.7 for our system) between training and test set (see Table 4.1), which gives a free boost to supervised systems but is not very representative of a real world scenario where no annotated document is available. Training our system with corpus data leads to an F1 improvement of +5.3 pt, +8.6 pt, +4.1 pt and +1.6 pt on resp. MEDLINE 2015 and EMEA 2015, MEDLINE 2016 and EMEA 2016. It outperforms other systems by a large margin on MEDLINE. It also outperforms (Roller et al., 2018) on EMEA 2015 and 2016, but not (Afzal et al., 2015) that obtained a perfect precision on EMEA 2015, at the cost of many handcrafted rules and extra labeled data.

		MEDLINE 2015			EMEA 2015		
		Prec.	Rec.	F1	Prec.	Rec.	F1
Others 2015	(Afzal et al., 2015)	80.5	57.5	67.1	**100**	77.4	**87.2**
	(Roller et al., 2018)	**83.1**	66.1	73.6	90.9	77.2	83.5
MLNorm	dist. supervised	75.6	71.9	73.7	79.7	73.6	76.5
	supervised	80.6	**77.5**	**79.0**	87.5	**82.7**	85.1

		MEDLINE 2016			EMEA 2016		
		Prec.	Rec.	F1	Prec.	Rec.	F1
Others 2016	(Cabot et al., 2016)	59.4	51.5	55.2	60.4	46.3	52.4
	(Roller et al., 2018)	77.1	66.3	71.3	78.1	**69.2**	73.4
MLNorm	dist. supervised	77.5	73.4	75.4	74.6	70.9	72.7
	supervised	**86.0**	**74.0**	**79.5**	**83.2**	67.0	**74.3**

Table 4.4 Main results for our system on the 2015 and 2016 Quaero datasets, and comparison with existing systems.

	English			Spanish			French			Dutch			German		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
(Roller et al., 2018) (Medline)	—	—	—	**79.0**	60.7	68.7	**79.4**	60.4	68.6	**76.7**	56.0	64.8	80.4	58.8	67.9
MLNorm (Medline)	82.7	80.7	81.7	76.5	**72.5**	**74.5**	75.6	**67.7**	**71.5**	75.4	**65.4**	**70.0**	**80.7**	**71.8**	**76.0**
MLNorm (all)	82.5	79.6	81.0	75.7	71.3	73.4	78.2	70.4	74.1	74.7	64.0	68.9	77.9	68.6	73.0

Table 4.5 Comparison between our system and Roller et al. (2018) on the Mantra dataset. Roller et al. (2018) only evaluate their method on Medline titles. We also provide the results for all documents in the Mantra corpus (all).

System	Example mention	Expected concept + synonyms	Predicted concept + synonyms
MLNorm(S2)	greffon renal	**C1261317** — [EN] transplanted kidney — [EN] kidney transplant — [EN] structure of transplanted kidney	✓
	cinquième métacarpien	**C0730166** — [EN] bone structure of fifth metacarpal — [EN] fifth metacarpal bone — [EN] fifth metacarpal	✓
	vaccination par le b.c.g	**C0199804** — [FR] immunisation contre la tuberculose — [EN] bcg vaccination — [EN] tuberculosis vaccination — [EN] tuberculosis immunization — [EN] administration of bcg vaccine… — (other similar English synonyms)	✓
	in vitro	**C0681828** — [EN] in vitro study — [EN] studies vitro — [EN] study vitro	**C3850137** — [EN] in vitro techniques — [EN] technique in vitro — [EN] in vitro as topic
	coffea robusta	**C0678439** — [EN] coffea rubusta (food)	**C1138610** — [EN] coffea arabica
mBERT-MT	cellar (translated from the French “cave”)	**C0042460** — [EN] vena cava structure — [EN] venae cavae — [EN] vena cava — [EN] vein — [MT] veins cellars (from “veines caves” — [MT] vein cellar (from “veine cave”)	**C0007634** — [EN] cell — [EN] cell structure — [EN] cells set — [EN] cellular — [EN] normal cell — [MT] cells (from “cellules”)
	be careful (translated from the French “attention”)	**C0004268** — [EN] attention — [EN] attentions	**C3257858** — [EN] my thinking is usually careful and purposeful

Table 4.6 Some predictions from our system. The last two columns contain the synonyms seen during training for the target concept and the predicted one, if different. Some long or similar synonyms have been removed to improve readability.

We can also see that the system using only French synonyms (FR-only in Table 4.11) performs much worse, with almost 20 points less in recall than the model trained with all the terminology, which we can attribute to the missing concepts in the French UMLS.

On the Mantra Medline titles (Table 4.5), the F1 score of our system outperforms the reported results of Roller et al. (2018) by a large margin in all languages, namely Spanish (+ 5.8 pt), French (+ 2.9 pt), Dutch (+ 5.2 pt) and German (+ 8.1 pt), as illustated in Table 4.5. However, it is worth mentioning that their method obtains a higher precision in all languages except German. Besides our use of a pretrained Transformer to compute rich representations of entities, we believe that this gap is also explained by their only bilingual translation, since they train a system for each language to translate entities into English. In contrast, we take advantage of all the languages to train a single multi-lingual system. We will expand further on this aspect in Section 4.5.6.

We will now discuss the experiments described in Section 4.4.2

4.5.2 Impact of the two steps training

Our experiment with one-step training procedure showed no improvement over the two-step training (Table 4.7, "1-step"), and took approximately 15 hours instead of 7 hours (5 hours for the first step and 2 hours for the second step with one million synonyms). Our two-step method can therefore effectively reduce training time without reducing accuracy by choosing an appropriate partition of the training data. Our results even show a slight loss in performance for the one-step model. This could be explained by the regularization that occurs in the two-step training when we freeze the encoder during Step 2. Indeed, since most of the data seen during Step 2 is English, unfreezing the encoder may encourage the model to forget its inner translation abilities.

	MEDLINE 2015			EMEA 2015		
	Prec.	Rec.	F1	Prec.	Rec.	F1
2 steps	75.6	**71.9**	**73.7**	79.7	**73.6**	**76.5**
1 step	**78.5**	69.2	73.6	**81.6**	71.4	76.2

Table 4.7 Comparison on Quaero 2015 of two models trained with the one step procedure or the two steps procedure

4.5.3 Impact of translating entities

Our experiments with translated French terms (Table 4.8) show that even a good machine translation model can lower the accuracy of the final model. We experimented with both English BERT and multilingual BERT to account for the impact of the transformer pre-training language. We could argue that the off-the-shelf translation model could be improved by

	MEDLINE 2015			EMEA 2015		
	Prec.	Rec.	F1	Prec.	Rec.	F1
MLNorm (dist. supervision)	**75.6**	**71.9**	**73.7**	**79.7**	73.6	**76.5**
w/ mBERT-MT	73.5	70.2	71.8	78.4	**74.6**	76.5
w/ BERT-MT	75.1	69.8	72.4	77.4	73.7	75.5

Table 4.8 Comparison of our system with a comparable machine translation approach, using our classifier.

fine-tuning on UMLS synonyms like Roller et al. (2018). However, we think that those results hint at the fact that translation and indexer pipeline may suffer from error cascade: being trained in an end-to-end fashion, our system does not suffer from this behavior. Table 4.6 shows that the ambiguity of some terms ("cave" can mean both "cellar" and "cava" in English) is lost during translation.

4.5.4 Impact of the pretrained embeddings

	MEDLINE 2015			EMEA 2015		
	Prec.	Rec.	F1	Prec.	Rec.	F1
mBERT (mulilingual)	75.6	**71.9**	**73.7**	79.7	**73.6**	76.5
camemBERT (FR)	**76.9**	70.4	73.5	**82.1**	69.9	75.5
BERT (FR)	75.9	71.6	**73.7**	80.5	73.4	**76.8**

Table 4.9 Comparison on Quaero 2015 of two models using differently pretrained BERT models

Our experiments with French-only embeddings CamemBERT and English-only embeddings BERT, reported in Table 4.9, show that our hypothesis that multilingual embeddings improve the system's performance is not verified, with almost no difference between these three embeddings. French wordpieces and embeddings can handle medical terms in English, and vice versa. Even if this can be again explained in part by the proximity of the two languages concerned, the low results of EN-only in Table 4.11, yet benefiting from much more training data, suggest that it is not that obvious; besides, other papers in the literature suggest that multilingual embeddings are helpful even for such pairs of languages (Pires et al., 2020; Wu and Dredze, 2020). This observation may also be due to the fact that medical synonym normalization data (short word sequences) is quite different from BERT pretraining data (full sentences), so it is harder for the model to re-use its multilingual knowledge. This aspect deserves more experiments, notably on other, non-European languages. Note that biomedical-specific embeddings such as Clinical BERT (Alsentzer et al., 2019) are not yet available in French, which is why we did not consider them. Moreover, as illustrated in Table 4.6, we can see that the model correctly predicts concepts, even when no common wordpieces exist between the entity and the training synonyms of the target concept. Therefore, the proximity

between French and English cannot be the only explanation to the model performance. To correctly classify the mention "cinquième métacarpien" (fifth metacarpal bone) to its concept, without having the numeral "cinquième" in any of the training synonyms, the model must have learned to generalize from other concepts that contained both French "cinquième" and English "fifth" in their training synonyms. We can also note that despite addressing out-of-vocabulary errors with wordpiece vocabularies, such errors still exist. For example in Table 4.6, "robusta" (single wordpiece "##robusta") and "rubusta" (two wordpieces, "##rubus" and "ta") are tokenized differently despite having almost identical characters. Recent models (El Boukkouri et al., 2020) that compute wordpieces embeddings from their characters are a promising approach to reduce such errors.

4.5.5 Impact of more French data

	MEDLINE 2015			EMEA 2015		
	Prec.	Rec.	F1	Prec.	Rec.	F1
UMLS 2014AB	**75.6**	**71.9**	**73.7**	**79.7**	**73.6**	**76.5**
UMLS 2019AB	75.3	71.0	73.1	79.5	72.8	76.0

Table 4.10 Comparison on Quaero 2015 of two models trained with the synonyms of 2014AB UMLS or those of the 2019AB UMLS

Our experiment with UMLS 2019AB (Table 4.10) leads to slightly lower results than the model trained with the 2014AB version, despite the much higher number of concepts with French synonyms. The system has more French terms to train on, but the coverage in the Quaero corpora is not much better. In addition, this could be explained by the higher number of concepts, i.e. choices, for each model prediction. Since Quaero annotations are based on a different version of UMLS, it is possible that some entities would have been annotated differently if the 2019 version of UMLS had been used, possibly leading to some prediction errors.

4.5.6 Impact of the training languages

	MEDLINE 2015			EMEA 2015		
	Prec.	Rec.	F1	Prec.	Rec.	F1
FR synonyms only	73.8	52.8	61.5	82.4	52.8	64.4
EN synonyms only	**79.7**	45.1	57.5	**84.3**	41.0	55.1
FR + EN synonyms	78.3	**62.1**	**69.3**	82.7	**57.4**	**67.8**

Table 4.11 Comparisons between monolingual training setups and bilingual training evaluated on the Quaero dataset. Only concepts that have both French and English synonyms were kept.

In Table 4.11, we compare the same model trained with either only the French synonyms of bilingual concepts, only the English synonyms, or with both (FR + EN synonyms). FR + EN achieves an 7.8 pt improvement over FR-only, despite having the same concepts coverage and the same pretrained embeddings. This indicates that a larger training set, even in a different language can help improve the system's performance by a significant margin. This improvement could be attributed to the lexical similarities between French and English languages. For example in Table 4.6, the only training French synonym of "vaccination par le b.c.g" is "immunisation contre la tuberculose" and shares no common word. The system can therefore benefit from the addition of similar terms, such as "bcg vaccination" even though they are in English.

	English	Spanish	French	German	Dutch	Overall
ENG	81.1	52.2	53.0	45.9	38.7	62.2
ENG+SPA	**81.9**	72.8	60.8	49.9	40.0	67.4
ENG+FRE	81.4	56.9	73.7	48.8	40.9	67.4
ENG+GER	81.8	55.5	56.6	70.9	45.2	68.3
ENG+DUT	81.4	55.7	55.1	51.7	66.1	66.6
Multilingual	81.0	**73.4**	**74.1**	**72.9**	**68.8**	**75.7**

Table 4.12 F1-score of the system on the Mantra corpus when trained with different language combinations

The F1 scores from our experiment with different language combinations on the Mantra dataset are presented in Table 4.12. Not surprisingly, looking at the diagonal of the table, the model performs better for a given language when that language was part of the training combination, and conversely performs worse when that language was not seen during training. However, we also observe that the multilingual training configuration improves the performance for all non-English languages compared to the bilingual training. In particular, the Dutch and German scores increase by more than 2 points with the multilingual model compared to the bilingual model. The different models seem to achieve similar scores for English, but we note that the the Spanish-English combination seems slightly better.

Another interesting way to read the results is to look at the interactions between different languages (other than English). We can see that the Dutch language benefits the most from training in German, and vice versa, and that the French language benefits the most from training in Spanish, and vice versa. This can be explained by the etymological similarity between the languages in these two pairs. Both these experiments on Quaero and Mantra demonstrate the transfer that operates between languages, and the importance of training on multiple languages when possible.

4.6 Conclusion

In this chapter, we have presented a method for medical entity normalization. Our method is able to handle a large number of concepts and predict entities in French, despite the low number of French synonyms in international terminologies. We obtained state of the art results on the Quaero and Mantra corpora. We demonstrated the importance of training with French and English data jointly, and even the benefit of training a single multilingual model, instead of several bilingual models.

Our system can therefore be used to normalize simple entities on medical documents, and does not require manually annotated concepts to obtain good results. These structured predictions can then be used directly to query reports, or as inputs to more complex systems. In the next chapter we will focus on the task of extracting structured entities from breast imaging reports.

Chapter 5

Structured entity extraction from breast imaging reports

Contents

In this chapter, we focus on the problem of extracting structured entities from breast radiology reports, as described in the introduction to this thesis. These reports contain rich and useful information about a patient's physical condition, clinical history, and physician assessments and recommendations.

As discussed in the Chapter 2 Section 2.4.1, the task of structured entity extraction can be approached from a frame semantic perspective. We describe a frame-based annotation scheme for extracting radiological entities, procedures, and assessments from these reports. Using this scheme, we describe a new corpus of 120 annotated documents from the APHP clinical data warehouse. Next, we consider the task of automatically generating these annotations. While many methods exist for related topics such as event extraction, slot filling, or discontinuous entity recognition, a challenge in our study resides in the fact that clinical reports typically contain overlapping frames that span multiple sentences or paragraphs. We propose a new method that addresses these difficulties and evaluate it on the new annotated corpus. Despite the small number of annotated documents, we will see that the hybridization between 1/ a system of constraints on the outputs of the system, 2/ a terminology and a 3/ learning-based system allows us to quickly obtain proper results. We will also introduce the concept of scope relations and show that it both improves the performance of our system, and provides a visual explanation of the predicted relations. In this study, we will focus only on the extraction and classification of frames, and leave the task of object extraction, i.e. frame coreferences, for future work.

In order to avoid confusion, we will call simple entities "mentions", the conjunction of several mentions and labels "frames" and the union of several frames "objects". Examples of mentions will be denoted by the form [the mention].

This chapter is organized as follows. We first describe the annotation scheme and the resulting corpus in Section 5.1. In Section 5.2, we describe the architecture of the proposed model. We will detail the different components that will allows us to extract and normalize the named entities and compose them as frames. We present several experiments in order to study the contribution of the various components of the model and the choices regarding its training in Section 5.3, and the discuss the results in Section 5.4. Finally, we close this chapter with a conclusion.

This study was approved by the institutional review board at APHP (CSE 190022) as part of the EZMammo project. Only previously pseudonimized documents were used in this study (Paris et al., 2019). The source code for the model described in this Chapter is available at the following URL: https://github.com/percevalw/breast-imaging-frame-extraction.

5.1 Annotation scheme

We first detail the annotation scheme and the resulting dataset. We focus on entities related to therapeutic (e.g. surgery) or diagnostic (e.g. mammography) procedures, radiological observations (e.g. cysts or masses), and breast density or BIRADS scores. The relevant entities to extract were the result of discussions with a physician expert in the field. The annotation scheme itself was the result of many iterations between annotations and scheme revision. The document-level statistics are detailed in Table 5.1. The corpus consists of 120 annotated clinical documents, 80 for the training set and 40 for the evaluation set.

	train	**test**
count	80	40
average words	361.0750	362.175
average lines	45.7375	45.475
average frames	19.4750	18.425
average objects	10.8125	10.475

Table 5.1 Document level statistics for the EZMammo NLP corpus

Our annotations focus on three types of elements: mentions, frames and objects. Mentions are simple named entities that consist of a begin, an end, a type and optionally a value. We have seen in Chapter 3 how to extract entities and in Chapter 4 how to label them using a fine-grained terminology. Frames are conjunction of mentions, that is entities in which every mention applies its meaning. Finally, objects are unions of frames that define the same real world elements.

As an example, we seek to structure the following report excerpt. The extracted mentions, frames and objects are presented in Figure 5.2.

```
1  Breast ultrasound:
2  Left:
3  Two cysts located on the 8 o'clock radius at
4  3 cm, and at 2cm on the 6 o'clock radius.
5  These nodules are millimetric.
6
7  Right:
8  No abnormal masses to report.
9
10 CONCLUSION :
11 Multiple cysts on the left.
```

Figure 5.1 Fictitious radiology report except

Diag. procedure object 1	Frame 1		Frame 2	
field	**value**	**justification**	**value**	**justification**
trigger		[Ultrasound]		[Ultrasound]
type	ultrasound	[Ultrasound]	ultrasound	[Ultrasound]
organ	breast	[Breast]	breast	
laterality	left	[Left]:	right	[Right]:
temporality	overlap		overlap	

Finding object 1	Frame 3		Frame 5	
field	**value**	**justification**	**value**	**justification**
trigger		[cysts], [nodules]		Multiple [cysts]
organ	breast	[Breast]	breast	
laterality	left	[Left]:	left	on the [left]
temporality	overlap		overlap	
quadrant				
size		[millimetric]		
distance	30mm	[3 cm]		
angle	8	[8 o'clock radius]		

Finding object 2	Frame 4		Frame 5	
field	**value**	**justification**	**value**	**justification**
trigger		[cysts], [nodules]		Multiple [cysts]
organ	breast	[Breast]	breast	
laterality	left	[Left]:	left	on the [left]
temporality	overlap		overlap	
quadrant				
size		[millimetric]		
distance	20mm	[2 cm]		
angle	6	[6 o'clock radius]		

Table 5.2 Mention, frames and objects extracted from the example 5.1

5.1.1 Mention annotation

First, we annotate several types of mentions, each justifying the value of a field in a frame. In our scheme, each mention has an *effect* that can be combined with other effects to describe an entity. Some mentions have the effect of justifying the existence of a frame: we will refer to these mentions as "triggers". Other mentions have the effect of specifying an attribute of an object: we will refer to them as "attribute" mentions. No frame is created if there is no trigger, even if several attributes are present. In the example 5.1, the trigger [Ultrasound] mention has the effect of creating at least one "Diagnostic procedure" frame, whereas the [millimetric] attribute has the effect of giving a size to the frames that it is part of.

The trigger mention types are *BIRADS score*, *Breast density*, *Diagnostic procedure*, *Therapeutic procedure* and *Radiological lesion*. The additional attribute mention types are *Diagnostic procedure type*, *Therapeutic procedure type*, *Breast density type*, *BIRADS score type*, *Organ*, *Laterality*, *Temporality*, *Size*, *Distance*, *Angle* and *Breast quadrant*.

We have chosen to annotate mentions describing attributes (such as laterality or size) even if they are not part of any frame. On the other hand, trigger mentions are not annotated if they do not justify the presence of an object. In the sentence "*No suspicious mass on the right*", only [right] is annotated as potentially justifying the laterality of an object, but not [mass] since it is preceded by a negation, and therefore does not justify the creation of any radiological lesion object.

Finally, each mention is classified, or normalized, according to a predetermined set of values. For example, a trigger mention "Breast density" may be labeled exclusively "type 1", "type 2", "type 3", "type 4". A laterality can take the values "left", "right", or "left + right".

The annotation statistics for mentions and their type are described in Table 5.3.

5.1.2 Frame annotation

Frames describe semantic slices of an object, or conjunction of triggers and attributes that share their effect (or concept) on a given entity. In the above example, [8 o'clock radius] (applying an angle), [3cm] (applying a distance), [Left] (applying a laterality), [Breast] (applying an organ) and the trigger [cysts] (applying the effect of existing) share their respective effect on a same slice of an object. These mentions may be located in different sentences or paragraphs, and a field in a given frame may be justified by several mentions. On the other hand, if an object is described in several places in the text, we annotate it with several distinct frames. The notion of "several places" and the choice to split a same object into multiple frames is sometimes ambiguous. We choose to annotate a single frame for an object if it is described on several juxtaposed sentences, and split it into multiple frames otherwise. For instance, the [cysts] trigger is combined with the [nodules] trigger because they are found in juxtaposed sentences, and [nodules] is clearly referring to the previously mentioned [cysts].

All frames follow a specific scheme that constraints the set of labels and mentions (or effects) combinations. A summary of the frame schemes is shown in Figure 5.4. In practice, these constraints take the form of a list of 2502 label tuples that enumerates every possible mention / label combination. For example, a Cancer Risk type 0 on the right breast at the time of the exam is described by the following tuple:

(score_trigger, score_type_0, temp_overlap, organ_breast, lat_right)

As shown in the structured output 5.2 of example 5.1, five frames are annotated:

— the ultrasound "Diagnostic procedure" frame for its left location, composed of the [Breast], [ultrasound] and [left] mentions on lines 1 and 2

Mention type	Mention value (if any)	Train	Test	Examples
Trigger mentions				
Finding		491	228	[nodules], [mass]
Diagnostic proc.		468	227	[mammography]
Therapeutic proc.		80	32	[surgery], [chemo]
BIRADS score		132	64	[ACR 3]
Breast Density		55	27	[type 2 density]
Attribute mentions				
Diagnostic proc. type	biopsy procedure	123	49	[micro-biopsy]
	ultra sound procedure	128	81	[ultrasound]
	MRI procedure	32	17	[MRI]
	mammography	137	74	[mammography]
	other	17	11	[PET scan]
	palpation	19	4	[palpation]
Therapeutic proc. type	surgery	32	13	[tumorectomy]
	other	19	7	[radiotherapy]
BIRADS score type	Type 0 BIRADS score	1	0	[ACR0]
	Type 1 BIRADS score	20	7	[ACR 1]
	Type 2 BIRADS score	48	30	[score BIRADS 2]
	Type 3 BIRADS score	19	9	[ACR 3]
	Type 4 BIRADS score	18	8	[ACR 4]
	Type 4a BIRADS score	6	1	[ACR4a]
	Type 4b BIRADS score	2	1	[ACR 4 b]
	Type 4c BIRADS score	2	5	[ACR 4c]
	Type 5 BIRADS score	8	4	[ACR 5]
	Type 6 BIRADS score	2	1	[ACR 6]
Density type	Type 1 breast density	10	3	[type A density]
	Type 2 breast density	24	15	[type 2 density]
	Type 3 breast density	18	7	[type III density]
	Type 4 breast density	2	2	[type D density]
Angle		51	19	[8 o'clock position]
Radial distance		63	30	at [1cm from the nipple]
Size		130	70	measured at [1cm]
Temporality	future temporality	35	20	[in 6 months]
	current temporality	101	47	exam date if any
	passed temporality	212	106	[last time]
Organ	breast organ	461	235	[breast]
	other organ	78	8	[kidney], [hepatic], ...
Laterality	left laterality	345	188	[left]
	right laterality	349	183	[right]
Breast quadrant	areolar region	42	28	[para areolar region]
	axillary region	110	55	[axillary areas]
	lower outer quadrant	21	7	[lower outer quadrant]
	lower inner quadrant	12	6	[lower inner quadrant]
	upper outer quadrant	60	37	[upper outer quadrant]
	upper inner quadrant	10	9	[upper inner quadrant]
	outer quad. junction	27	12	[outer quadrants junction]
	lower quad. junction	19	7	[lower quadrants junction]
	inner quad. junction	6	5	[inner quadrants junction]
	upper quad. junction	25	3	[upper quadrants junction]

Table 5.3 Mention annotation statistics

— the ultrasound "Diagnostic procedure" frame for its right location, composed of the [Breast], [ultrasound] and [right] mentions on lines 1 and 7

— the first "Finding" frame of the first nodule, with two trigger mentions: [cysts] and [nodules] and attribute mentions [8 o'clock position], [3cm] and [millimetric] on lines 1, 2, 3, 4 and 5

— the first "Finding" frame of the second nodule, with two trigger mentions: [cysts] and [nodules] and attribute mentions [6 o'clock position], [2cm] and [millimetric] on lines 1, 2, 4 and 5

— the second "Finding" frame of both nodules in the conclusion: composed of the trigger [cysts] and the laterality [left] on line 11

Since the mass negation on line 8 is not an indication of the presence of an object, we do not annotate it. The temporality of each frame overlaps the exam, although no explicit mention can support this fact, so we fill the temporality field of the frames with the value "overlap" and leave the justification empty.

5.1.3 Object annotation

Finally, the different frames are grouped into objects. Objects are union of frames. For a given set of concepts, multiple frames might be required to describe a same object. In the context of of growing lesions, a union of multiple (temporality, size) conjunctions can represent the evolution. In an other setting with moving objects, a union of (temporality, localisation) labels could be used. In our case, as we represent lateralities with two exclusive "left" and "right" concepts, bilateral objects are described with two co-referent frames.

In the previous example, three objects are annotated, grouping two frames for the ultrasound procedure and two frames for each cyst. The last nodule frame in the conclusion is a case of plural coreference, since it its attributes apply to both objects. In this case, the frame describing several objects is added to each one. The statistics of objects in the annotated documents are described in Table 5.5. This step amounts to annotating coreferences between frames.

5.1.4 Annotation process

Clinical documents were de-identified automatically beforehand and the manual annotation was performed with Brat (Stenetorp et al., 2012) by two annotators. 120 clinical reports were sampled from a from of query the APHP clinical data warehouse that combined the substrings "mamm" (to obtain breast related reports), "ACR" and "BI-?RADS" (to obtain BIRADS scores). Some sampled reports were not breast radiology reports, yet we kept them as negative samples. Since Brat was not originally designed to annotate long multi-line relations, using the "Event" or "Relation" annotations turned out to be impractical and made the annotated documents

Frame type	Field	Field value
Cancer Risk	score trigger	
	score type	type 0 / type 1 / ... type 6
	laterality	left / right
	temporality	overlap / before doc time
Breast density	density trigger	
	density type	type 1 / type 2 / type 3 / type 4
	laterality	left / right
	temporality	overlap / before doc time
Diagnostic procedure	diag. trigger	
	diag. type	mammography / ultrasound / ...
	organ	breast / other
	laterality	left / right
	temporality	overlap / before / after doc time
Therapeutic procedure	ther. trigger	
	ther. type	surgery / other
	organ	breast / other
	laterality	left / right
	temporality	overlap / before / after doc time
Radiological lesion	lesion trigger	
	organ	breast / other
	laterality	left / right
	temporality	overlap / before doc time
	quadrant	lower inner / axillary region / ...
	size	
	distance	
	angle	

Table 5.4 Schemes of the extracted frames. Each frame is composed of multiple fields that can take a value.

	train		**val**	
	object	frame	object	frame
radiological lesion	279	449	122	210
diagnostic procedure	285	795	141	379
therapeutic procedure	51	83	22	29
BIRADS score	152	152	82	82
breast density	98	98	52	52

Table 5.5 Frame and object statistics in the annotated corpus

hard to read. We choose instead to annotate frames using a mix of identifier attributes (frame1, frame2, ...) on mentions, and relations on close-by mentions. Coreferences, i.e. object annotation, were annotated using identifier attributes (objectA, objectB, ...) for the same reason. The BRAT annotations of Example 5.1 are shown in Figure 5.2. The direction of

the annotated relations is only used to extract the paths along which the frames are clustered, but is not used as directed relation in our model, since it is not consistent.

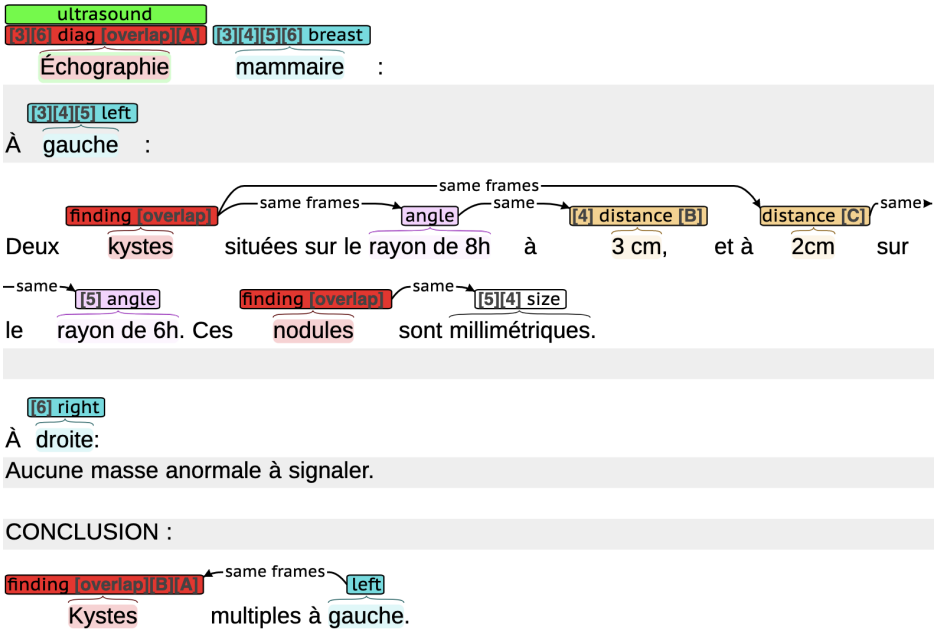


Figure 5.2 BRAT annotation of Example 5.1

5.1.5 Metrics

We propose three metrics to evaluate the predictions at the mention and frame level. A more detailed explanation of the procedures used to compute these relaxed metrics is given in Appendix A.

5.1.5.1 Mention

The mention metrics scores the retrieved mentions using the standard NER metrics. Every value of a multi-label mention produces its own triplet (begin, end, value), and we run the half NER metric described in Chapter 3, in which a mention is only counted as correct if its Dice overlap measure with a gold mention is at least 0.5 (i.e. at least half of the predicted and expected words match) and its label/value is correct.

5.1.5.2 Frame support

We design a specific metric to score frame support in a document. When comparing two frames, each supported field of the predicted frame is counted as correct if it overlaps any mention of the same field in the other frame. The score is 1 if all fields are correct, 0 if no

field is correct and the Dice overlap of predicted and gold fields otherwise. In the following example:

Predicted: "*The small [nodules](lesion) on the [left](lat:left) measure [2cm](distance)*"

Expected : "*The [small nodules](lesion) on the [left](lat:left) measure [2cm](size)*"

Only **2** of the **3** predicted and **3** expected fields [nodules], [left] have the correct label and overlap, so the final mention field overlap score is $2 \cdot \frac{\mathbf{2}}{\mathbf{3+3}} = 0.66$

5.1.5.3 Frame label

Finally, we evaluate the final multi-label classification of frames using the Frame label metric. Two frames are matched only if their triggers overlap, and all predicted labels are correct. We add an exception to this rule: if a label is predicted in a frame, and the target frame does not contain it but another frame of the parent object does, the label is not counted as false. For instance, if an object is described in two places in the text, the first time only with a left laterality, and the second time only with its size, we will not penalize the model if it predicts a left laterality on the second frame.

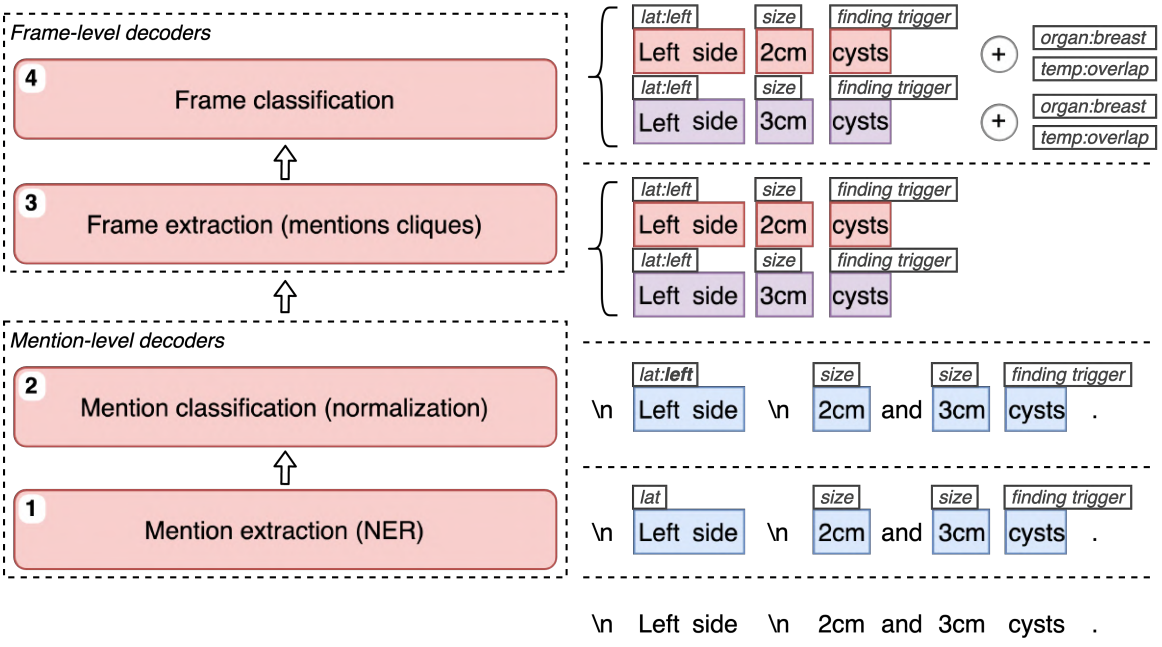


Figure 5.3 Overview of the decoder

5.2 Proposed method

We now detail a neural network based method to automatically extract the previously described structured entities from clinical reports.

We encode each documents as word embeddings and share them with the downstream decoding components. Like most relation and event extraction models, our model operates as a pipeline. As illustrated by Figure 5.3, the first two mention-level decoders extract the named entities (step ①), or mentions, that are likely to be used in the composition of structured entities, and normalize them (step ②) to obtain the value of the field they apply to. The next two decoders focus on frame-level extractions. The frame extraction decoder (step ③) detects the relations between these mentions, or more specifically, extracts groups of mentions to form frames. The last frame classification decoder (step ④) predicts for each frame the values of the fields for which no mention was found, such as the temporality.

5.2.1 Text encoder

5.2.1.1 Word embeddings

Like the models of the previous chapters 3 and 4, we use a pre-trained BERT Transformer. Our documents are written in French, therefore a good candidate is the CamemBERT model (Martin et al., 2020) pre-trained on a general French corpus. A specifically pre-trained clinical French BERT would most likely perform better. However no such model has been trained to our knowledge. Following our experiments in Chapter 3 Section 3.7.4, we also average the wordpieces embeddings of a word to obtain its embedding, and add the left and right contexts (document context) of a sentence before running it through BERT.

5.2.1.2 Document-wide contextualization

As in the models of Chapter 3, we apply an LSTM layer on BERT embeddings. A notable difference with the previously addressed tasks is the longer size of the documents: BERT can only encode sequences of up to 512 wordpieces and more than half of our reports exceed 512 wordpieces. Several works on encoding long sequences by Transformers have emerged since 2019 (Beltagy et al., 2020b; Zaheer et al., 2020) but no pre-training has been applied to French to our knowledge. One strategy is to split these reports into sentences, apply BERT on each sentence and then re-contextualize these sentences by applying the LSTM on the concatenated sentence word embeddings. Additionally, the preliminary sentence splitting reduces the length of the processed sequences and thereby makes the processing of each document faster. This process is illustrated in Figure 5.4.

Moreover, because BERT models focus on sentences, the "line break" character is missing from their vocabulary and replaced by a single space during preprocessing. However, clinical documents typically contain multiple line breaks and this separation information would normally be lost. To prevent this, we replace all line breaks with the rarely used "_" character so that this information is kept in the generated embedding sequences.

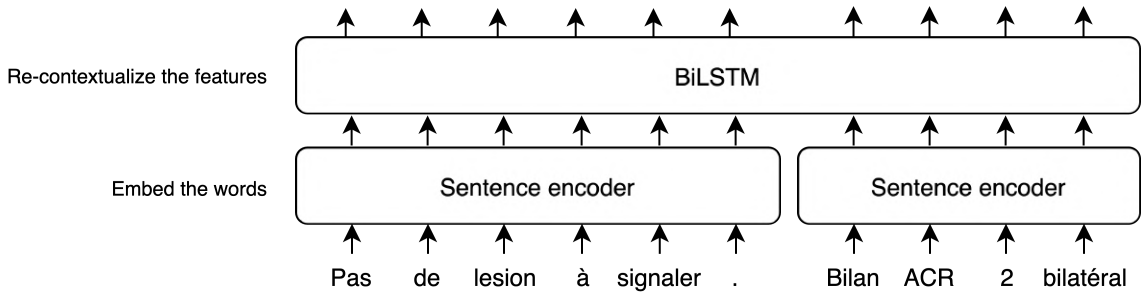
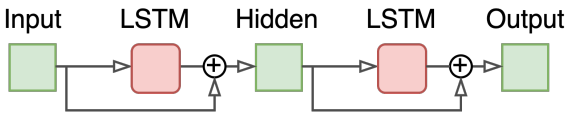


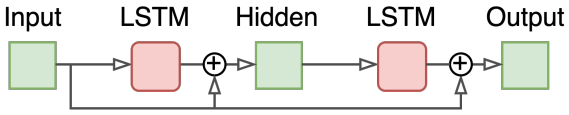
Figure 5.4 Overview of the document encoder

5.2.1.3 LSTM gating mechanism

We observed that the standard gating mechanism of the multi-layer LSTM converged poorly. We propose an alternative formulation in which the outputs of each LSTM layer are not mixed with the outputs of the previous layer, but with the input features of the contextualizer. We also drop the sigmoid weight and use simple addition instead. While in the original formulation, each LSTM layer must produce an update to the output of the previous layer, the role of each layer is now to produce an update to the first input. We will call the proposed gating mechanism "input-residual", as opposed to the standard "last-residual" mechanism. The difference between these two mechanisms is detailed in Figure 5.5.



(a) Standard "last-residual" gating mechanism



(b) Proposed "input-residual" gating mechanism

Figure 5.5 Difference between the gating mechanisms, shown on a two layer LSTM network. The top figure (a) shows the standard "last-residual" gating, while the bottom figure (b) shows the "input-residual" variant.

5.2.2 Mention recognition and normalization

5.2.2.1 Mention detection

We reuse the architecture of the BIOUL decoder described in Chapter 3. Mentions may overlap, but our annotations do not contain overlaps of the same type. For this reason, the

matching of start and end boundaries is unambiguous from the tags predicted by the BIOUL decoder and the biaffine module is not needed.

5.2.2.2 Mention normalization

Each mention is then classified, or normalized to obtain the values of the fields to which it applies. Unlike the system proposed in Chapter 4 which allowed only one concept per mention, each mention can accept several values. For example, "bilateral" is normalized as both "left" and "right". However, most mentions map to only one value. The mapping between NER labels and the legal multi-label combinations is part of the annotation scheme. Another difference is that we run the normalizer on embeddings of mentions in which the words carry contextual information from outside the mention. In contrast, the model of Chapter 4 processed each mention as a text sample on its own.

We compute a max-pooled representation for each mention m and project it against to obtain one score per label

$$\text{score}^{\text{label}}(m) = V^{\text{mention label}} \cdot \underset{w \in \text{words}(m)}{\text{maxpool}} \; E(w)$$

Finally the score of each possible legal label combination L_{mention} is computed as the score of the labels present in the combination. The probability of a combination is computed by normalizing over all legal combinations L_{mention}

$$\text{score}^{L_{\text{mention}}}(m) = \sum_{\text{label} \in L_{\text{mention}}} \text{score}^{\text{label}}(m) \tag{5.1}$$

$$\text{P}(L_{\text{mention}}|m) = \frac{1}{Z}\text{score}^{L_{\text{mention}}}(m) \tag{5.2}$$

$$\text{with } Z = \sum_{\text{legal } L_{\text{mention}}} \text{score}^{L_{\text{mention}}}(m) \tag{5.3}$$

5.2.2.3 Mention embedding

Each mention is represented by a single embedding in order to be processed by the next decoders. This embedding $E(m)$ is computed as the average embedding of the words of the mention.

5.2.3 Frame extraction

We now seek to extract the frames. Given that we have extracted entities in a previous step, we need a strategy to group mentions of a same frame together. The approach of most Event Extraction models is to extract one frame (event) per trigger mention, and look for related mentions that might be part of the same frame (event). However, many trigger mentions

belong to several distinct frames that can only be distinguished by considering interactions between their attribute mentions. Indeed, in a sentence containing an elliptic conjunction: "*Nodules of 2cm on the right and 3cm on the left*", the trigger mention [Nodules] belongs to two different frames, and the knowledge alone of trigger-attribute relations [2cm]$\Longleftrightarrow$[Nodule], [3cm]$\Longleftrightarrow$[Nodule], [right]$\Longleftrightarrow$[Nodule] and [left]$\Longleftrightarrow$[Nodule] is not sufficient to reconstruct the two frames.

To address this issue, an approach consists in listing all the possible combinations of mentions, then filtering them with a classifier (Björne and Salakoski, 2011, 2013, 2015; Heimonen et al., 2010; Liu et al., 2015; Miwa et al., 2010; Trieu et al., 2020). However, this solution does not seem satisfactory from a computational point of view. Indeed, a frame can contain up to 8 mentions (and more if there are several mentions for the same field), which quickly leads to a combinatorial explosion of possible frames.

We will now describe a method to overcome the previously discussed issues. The overall frame extraction component and its training are described in Figure 5.6.

5.2.3.1 Clique extraction

Our approach consists in examining relations between every mention of a document. The binary relation between two mentions answers the question: "are these two mentions part of the same frames ?". We can then extract maximal groups of entities such that in each group, all the mentions agree with each other on belonging to the same entity. In graph theory, this type of subgraph is known as a *clique*. To extract maximal cliques, i.e. cliques that cannot be extended by including one more mention, we use the NetworkX implementation based on the works of Bron and Kerbosch (1973) and Tomita et al. (2006), and only keep the cliques that contain at least one trigger mention.

Each mention u computes its agreement scores $r(u, v)$ with the other mentions v of a document. For two mentions u and v, we obtain two scores: the one computed by u on its agreement with v ($r(u, v)$), and the one computed by v on its agreement with u ($r(v, u)$). We define the final agreement score between the two mentions as the maximum score

$$R(u, v) = \max{}^{T} r = \max(r(u, v), r(v, u))$$

Intuitively, this means that one of the two mentions can be uncertain about the relationship.

5.2.3.2 Biaffine relation scores

A simple approach to compute $r(u, v)$ is to use a biaffine model. In our case, we compute this score as an attention score between the mentions. Additionally, we inject the relative distances between mentions inside the attention mechanism using a similar mechanism to He et al. (2020). This attention is the sum of a content-content attention (the original dot

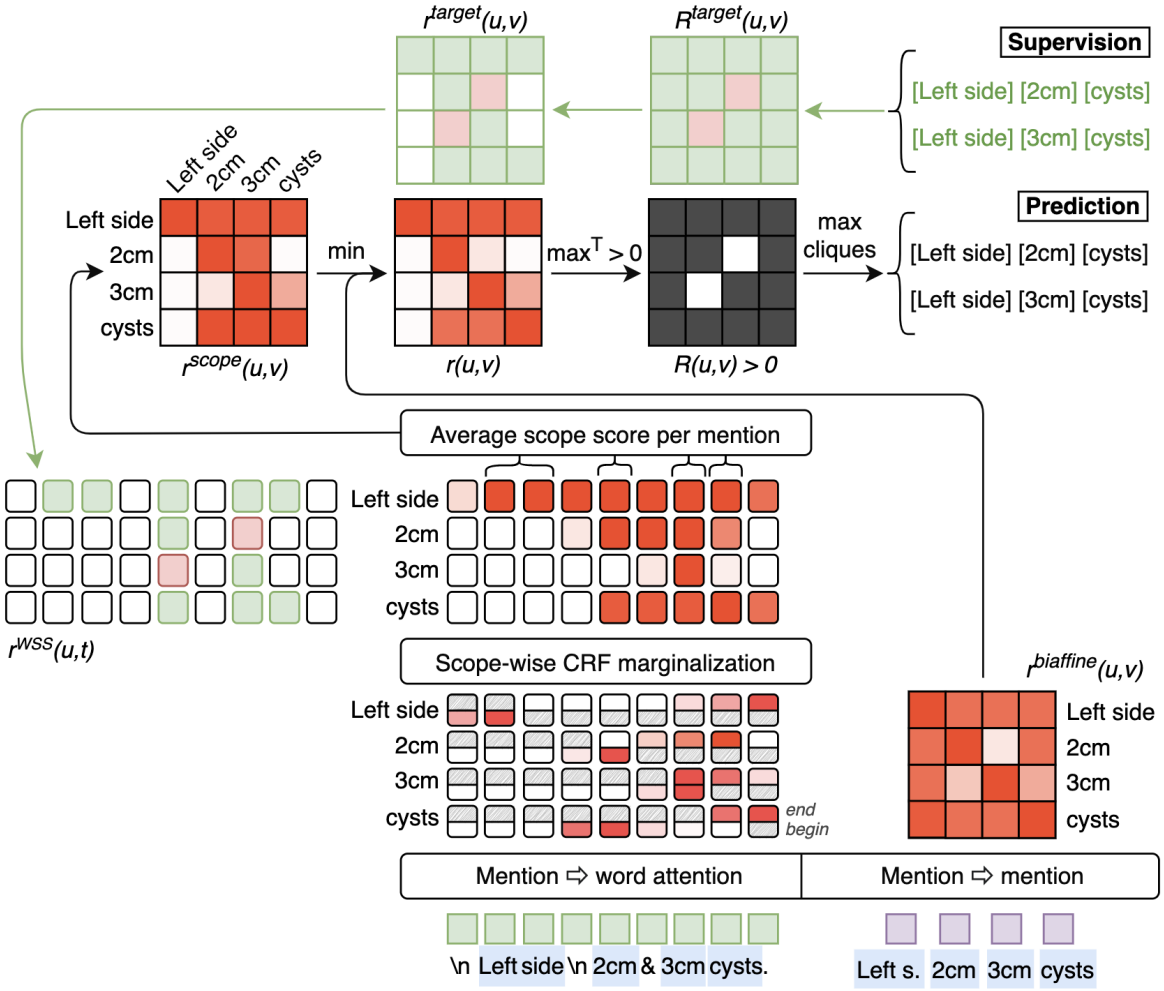


Figure 5.6 Overview of the frame extraction process and its supervision. Forbidden scope begin and end locations (because they are located after or before the mention) are grayed out. Green matrices and arrows at the left and top of the Figure show the possible supervision signals: red forces a logit to be negative, green forces a logit to be positive, and white means no supervision for the associated logit.

product attention of Vaswani et al. (2017)), a content-position attention and a position-content attention.

$$
\begin{aligned}
r^{\text{biaffine}}(u,v) = & \frac{(W_1^c u)\cdot(W_2^c v)^T}{\sqrt{3d}} && \text{content to content} \\
& + \frac{(W_1^p p_{u\to v})\cdot(W_2^c v)^T}{\sqrt{3d}} && \text{content to position} \\
& + \frac{(W_1^c u)\cdot(W_2^p p_{v\to u})^T}{\sqrt{3d}} && \text{position to content}
\end{aligned} \tag{5.4}
$$

with $W_1^c, W_1^c, W_1^p, W_2^p$ four projection matrices

and $p_{x\to y}$ the embedding of the relative position of y w.r.t. x

To get a better intuition about these different types of attention, we formulate them as fictitious search samples from a given mention in the document:

— *content-content* : "my content is 'ultrasound' so I'm looking for other mentions whose content contains information about temporality"

— *content-position*: "my content is 'ultrasound' so I'm looking for mentions that are 3 positions after of me"

— *position-content* : "regardless of my content, I will attend to the mention one word away from me if it contains information about temporality, two words way next word if it contains information about laterality, etc."

5.2.3.3 Scope relation scores

We propose another approach for the same relation extraction task, based on the concept of scopes. Scopes are annotations of contiguous text zones on which a named entity referred to as a "cue" applies its meaning. Scopes have been mostly studied in the context of negation and uncertainty detection (Dalloux et al., 2020; Khandelwal and Sawant, 2020; Li and Lu, 2018; Vincze et al., 2008). For example in the sentence: "*There is no sign of cancer*", the scope of the negation entity [no] is "sign of cancer". We propose to extend this concept to all types of named entities and make it the primary mode of relation extraction in our problem. Indeed, it may be simpler for the model to detect where the scope of a mention starts and stops, and to retrieve all entities between these boundaries, rather than inferring the value of the relation for each pair of mentions. In the example of Section 5.1.2, the scope of laterality [Left] covers all the section and therefore applies its effect to all frames composed of these mentions, and the scope of one of the two mentions [2cm] and [8'oclock position] contains the other mention.

For the mathematical details of our formulation, we will call u and v two mentions, and t a token (or word) of the document. Each scope is represented with the BIOUL format. We compute two attention matrices $A^B(u,t)$ and $A^L(u,t)$ between the mentions and words, using the relative attention mechanism described in Section 5.2.3.2, to obtain start (B) and end (L)

scope scores for each word. We prevent the start of a scope from being after the first word of a mention, and the end of a scope from being before the last word of a mention, which means that we impose that mention is contained within its scope. The score of the tag U (scope that only contains one word) can be computed as the sum of the start and end scores.

$$S_{\text{B}}(u,t) = \begin{cases} -\infty & \text{if } t \text{ is after the first word of the mention} \\ A^B(u,t) & \text{otherwise} \end{cases} \tag{5.5}$$

$$S_{\text{L}}(u,t) = \begin{cases} -\infty & \text{if } t \text{ is before the last word of the mention} \\ A^L(u,t) & \text{otherwise} \end{cases} \tag{5.6}$$

$$S_{\text{U}}(u,t) = S_{\text{B}} + S_{\text{L}} \tag{5.7}$$

$$S_{\text{I}}(u,t) = 0 \tag{5.8}$$

$$S_{\text{O}}(u,t) = 0 \tag{5.9}$$

To know if a word is in the scope of a mention, we compute the marginalized probabilities of a CRF (hereafter referred to as Scope CRF) with the forward-backward algorithm that we apply to the scope of each mention. The Scope CRF is parameter-less but illicit transitions (such as I $\longrightarrow$ B or L $\longrightarrow$ I) between tags are prevented (i.e. the transition is set to $-\infty$). A word is in a scope if it is labeled I, B, L or U but not O. The score $r^{\text{scope}}(u,t)$ of each word t being in the scope of u is therefore:

$$r^{\text{scope}}(u,t) = S_{\text{B}}^{\text{marg}} + S_{\text{L}}^{\text{marg}} + S_{\text{U}}^{\text{marg}} + S_{\text{I}}^{\text{marg}} - S_{\text{O}}^{\text{marg}} \tag{5.10}$$

$$\text{with } S_{\text{BIOUL}}^{\text{marg}}(u) = \text{ForwardBackwardCRF}(S_{\text{BIOUL}}(u)) \tag{5.11}$$

and, the relation score between two mentions u and v is computed as the score of v being in the scope of u, i.e. the average of the scores of each word of v of being in the scope of u:

$$r^{\text{scope}}(u,v) = \frac{1}{|\text{words}(v)|} \sum_{t \in \text{words}(v)} r^{\text{scope}}(u,t) \tag{5.12}$$

Using a CRF allow us to never explicitly compute the score for a word to be in the scope of a given mention. Instead, we let the network predict the start and end of scope for each mention via the mention-word attention matrices, and use the CRF Scope to "paint" the inside and outside of the scopes in a differentiable way.

5.2.3.4 Score combination

The scope relation and biaffine relation scores are combined together. Because we defined scopes as continuous spans of text, it is possible that a mention falls in the scope of another mention and yet does not belong to its frame. In the following example

"*Mammography: we find the left mass biopsied in 2010. Nothing else in the right breast.*"

the scope of [Mammography] contains the temporality [2010] but the two mentions are not part of a same frame. Therefore, a relationship between two mentions is only predicted if both components (biaffine-based and scope-based) predict this relation. A mathematical formulation reflecting this constraint consists in returning the minimum of the two scores.

$$r(u, v) = \min(r^{\text{scope}}(u, v), r^{\text{biaffine}}(u, v))$$

5.2.3.5 Frame relation supervision

Asymmetric supervision Training the frame extraction module raises several difficulties. For two compatible mentions u and v, we require that $R(u, v)$ is positive if u and v are part of the same frames, and negative otherwise. The symmetric matrix $R(u, v)$ is the result of the maximum of a matrix $r(u, v)$ and its transpose, which, from a scope perspective, means that one mention can be within the scope of another without the reverse being true. One problem with supervising this non-differentiable maximum alone, is that the network might initially choose the wrong direction (e.g., decide that [Breast] belongs to the scope of [2cm], when it is the opposite), and get stuck in this wrong configuration for the rest of the training.

We propose instead to supervise one of the two direction scores specifically, instead of the maximum, through the asymmetric matrix $r(u, v)$. The difference between these two supervision modes is illustrated at the top of the figure 5.6. If two mentions u and v are not part of the same frames, then both direction should have a negative score. However, if the two mentions share the same frames, the question becomes: what do we ask the model to learn? We do not know a priori the direction of the relation u—v, only that one of the directions must have a positive score. One solution is to "explore" the different possibilities. To do this, we perform stochastic sampling of the supervised direction $r^{\text{target}}(u, v)$ by weighting each direction with its probability as estimated by the model:

$$[r^{\text{target}}(u, v), r^{\text{target}}(v, u)] \sim Cat(\text{softmax}([r(u, v), r(v, u)]))$$

The idea is that the model explores a few configurations at the beginning of the training when the probabilities are close to 0.5, and sticks to a given strategy that leads to stable solutions as learning progresses and its confidence in either direction increases.

Relation supervision heuristics We also propose to incorporate heuristics in the supervision matrix $r^{\text{target}}(u, v)$. If u belongs to strictly more frames than v, we maximize $r(u, v)$. If both belong to the same number of frames, we choose the direction that leads to the smallest number of wrong scope memberships. For example, in the example section 5.1.2, if we chose [Breast] to be in the scope of [2cm], then [8 o'clock radius] would also be in the scope of [2cm] due to the continuity of the scope. Conversely, if we choose [2cm] to be in the scope of [Breast], no erroneous scope assignment is generated. Finally, if no heuristic can be applied, we sample a direction as previously described.

Word-level scope supervision (WSS) We also propose to supervise the scopes at the word-level using partial word-level annotation generated from the r^{target} matrix, as illustated on the left side of Figure 5.6. Using this supervision matrix, for a given mention u, we can determine which words t of other mentions should be contained in its scope, which words of other mentions should not, and which words are not supervised. Because scopes are contiguous, if a mention v that is not part of the frame of u is contained within its partially supervised scope, i.e. if it is between two mentions that belong to the scope of u, we do not supervise its words and leave the biaffine component handle the non-relation detection. Thus, we generate a partial supervision matrix r^{WSS} with which we supervise the Scope CRF outputs. An example of this matrix is shown on the left of Figure 5.6.

5.2.4 Frame classification

Some labels of a frame such as its temporality or laterality may not be explicitly supported by the text. Each frame is therefore fed through a multi-label classifier. The possible field-value combinations and incompatibilities in a frame are known in advance. For example, a mammogram is necessarily located on the breasts. The "legal" label combinations are the same 2502 label tuples mentioned in Section 5.1.2.

We represent each frame by an embedding computed as a projection of the max-pooling output of its mentions' embeddings.

$$E(f) = W_{\text{frame}} \cdot \underset{m \in \text{mentions}(f)}{\text{maxpool}} E(m)$$

This embedding is then projected to give a score per label.

$$\text{score}^{\text{label}}(f) = V^{\text{label}} \cdot E(f)$$

Finally the score of each possible legal combination L_{frame} is computed as the score of the labels present in the combination. The probability of a combination is computed by

normalizing over all legal combinations:

$$\begin{aligned} \text{score}^{L_{\text{frame}}}(f) &= \sum_{\text{label} \in L_{\text{frame}}} \text{score}^{\text{label}}(f) \\ \text{P}(L_{\text{frame}}|f) &= \frac{1}{Z}\text{score}^{L_{\text{frame}}}(f) \\ \text{with } Z &= \sum_{\text{legal } L_{\text{frame}}} \text{score}^{L_{\text{frame}}}(f) \end{aligned} \tag{5.13}$$

During prediction, the label combinations are filtered to keep only those that contain at least all the supported labels predicted by the frame extraction decoder.

5.2.5 Optimization

Every component, namely the named entity recognition and normalization modules (5.2.2), the frame extraction module (5.2.3) and the frame classification module (5.2.4) are trained jointly. The encoder is shared and each decoder receives the prediction of the previous decoders.

The NER model uses the CRF Forward algorithm to compute the NER loss $\mathcal{L}_{\text{NER}}$, the normalization loss $\mathcal{L}_{\text{norm}}$ is the cross entropy loss. The frame extraction decoder relation loss $\mathcal{L}_{\text{relation}}$ is the sum of binary cross entropy for every valid supervised mention-mention pair and the CRF Forward algorithm to compute the Scope CRF loss $\mathcal{L}_{\text{scope}}$. Finally, the frame classification decoder loss is $\mathcal{L}_{\text{frame\_classification}}$ the cross entropy loss for every extracted frame.

The losses are combined through a weighted average:

$$\begin{aligned} \mathcal{L} = &\alpha_{\text{NER}}\mathcal{L}_{\text{NER}} \\ &+\alpha_{\text{normalization}}\mathcal{L}_{\text{normalization}} \\ &+\alpha_{\text{relation}}\mathcal{L}_{\text{relation}} \\ &+\alpha_{\text{WSS}}\mathcal{L}_{\text{WSS}} \\ &+\alpha_{\text{frame\_classification}}\mathcal{L}_{\text{frame\_classification}} \end{aligned} \tag{5.14}$$

5.2.6 Knowledge injection via data augmentation and constraints

We now discuss several techniques to inject knowledge via data augmentation and output constraints.

5.2.6.1 Data augmentation

Given the small number of annotated documents, we augment our training data in two ways. First, we randomly extract parts of documents such that no frame is cut, and add them as new documents to the dataset. This augmentation assumes that there is little dependence

between distant frames, and since we do not address the task of coreference in this work, splitting documents is not an issue. This augmentation also has the effect of reducing the training time (by around half in our experiments), as the average size of the training samples becomes smaller.

Second, we build synthetic sentences from a manually pre-defined lexicon of mentions, and add these sentences to the dataset. Because these sentences contain no frame annotations, the frame related losses are masked for these samples. The sentence creation process is the following: we randomly pick a synonym from the lexicon such as [ACR 6] and insert it in a randomly picked context from a predefined list such as "*There is* {} ." to generate "*There is [ACR 6].*" This sentence is then added to the list of training samples. Although this may seem very simple, we will see that this allows us to easily inject knowledge into our model and improve its performance. This method is closely related to the training of the Chapter 4, in which we built a training set from a terminology. As we mentioned in Section 5.2.2.2, however, our model deals with mentions that are part of a context, which is why we add an artificial context around each of our synonyms to avoid having too large a distribution gap between our real and synthetic samples.

The documents generated from these augmentations are mixed with the original documents such that every batch approximately contains $\frac{1}{3}$ of each (original, doc parts and lexicon sentences).

5.2.6.2 Output constraints

As stated in the section 5.1.2, the set of "legal" frame label combinations is known in advance. These label tuples supplement the manual annotations. Some background knowledge can be injected this way by constructing rules such as the fact that "left" and "right" are exclusive, or the fact that a mammogram is always performed on the breasts.

During the frame extraction step, relations between mentions that cannot be part of the same frame are filtered out during learning and prediction. We derive the allowed and forbidden relations from the list of label tuples mentioned earlier. For example, due to the spatial division of objects, two mentions [left] and [right] are incompatible and the relation $r([\text{left}], [\text{right}])$ is set to $-\infty$. This filtering reduces the number of possibilities that the model must evaluate. In addition, sometimes a procedure is explicitly located on a quadrant in the text. However, we chose not to extract the "quadrant" field for diagnostic or therapeutic procedures during annotation in order to simplify the schema. Preventing the model from learning that "procedure" and "quadrant" are incompatible in our schema improves the consistency of the supervision information.

During the frame classification step, instead of classifying each label independently, we score each combination of allowed labels, as described in equation 5.13. Conversely, scoring each label independently is equivalent to allowing all label combinations.

Maximum BERT sequence size	192 wordpieces
Post-encoder dropout	0.5
Decoder dropout	0.2
BiLSTM layers	3
BiLSTM hidden size	200
Number of steps	2000
Batch size	16
lr_{BERT}	$5e^{-5}$
lr_{main}	$4e^{-5}$
α_{NER}	2
$\alpha_{\text{normalization}}$	1
α_{relation}	1
α_{WSS}	1
$\alpha_{\text{frame\_classification}}$	0.5

Table 5.6 Hyperparameters

5.3 Experiments

We evaluate our proposed approach on the test set of the new annotated dataset, using the mention metric, the Frame Support metric and Frame Label metrics described in Section 5.1.5.

We also evaluate different document-level queries on the predicted frames. Each query extracts a deduplicated list of tuples for each document, and standard precision and recall metrics are computed on the predictions. As an example, the query "Lateralized current breast density" extracts tuples (laterality, density score) from frames with document overlap temporality, while the query "Current breast density" does not extract laterality.

5.3.1 Experimental setup

Hyperparameters were manually selected by trial and error on 20 documents from the training dataset. Many of them are the same as the model from Chapter 3. We optimize the parameters with the Adam optimizer (Kingma and Ba, 2015) without weight decay and use two learning rates: the first learning rate lr_{BERT}, that applies to the pretrained CamemBERT (Martin et al., 2020) base weights, is initialized at 5×10^{-5} and follows a linear schedule with a 10% warmup, while the second learning rate lr_{main}, for the other parameters, is initialized at 5×10^{-4} and follows a linear decay schedule with no warmup. The models were trained with a batch size of 16 samples. Due to the large size of model and documents, we used the gradient accumulation method to fit the available GPU memory (32Go). All experiments were averaged by training 3 differently seeded models. The main hyper-parameters are described in Table 5.6.

5.3.2 Ablations

Additionally, we perform several ablation experiments to investigate the design choices of our model:

— we look at the effect of the gating mechanism and the relative positional attention mechanism on our model

— we evaluate the contribution of scope relations and the effect of different types of supervision, i.e., we drop the word-level scope supervision and also change the supervision of the relation mechanism from asymmetric supervision of $r(u, v)$ to symmetric supervision of $R(u, v)$.

We also perform experiments on the training data. In particular, we investigate the contribution of the augmented samples, and the evolution of the performance with the amount of annotated data.

5.4 Results and discussion

5.4.1 Main results

Table 5.7 shows the performance on the different types of frames. The model performs better for frames with fewer fields such as Cancer or Breast densities. It is worth mentioning that matching all frame of a document is not necessary to answer most queries, since multiple frames can be co-referent.

The query metrics are shown in Table 5.8. Similarly the model performs better for queries that require less frame fields. The model low performance on the passed surgery query can be explained by the few number of annotated therapeutic procedures, and the difficulty to extract the temporality, that sometimes requires complex contextual and global reasoning.

We visualize the predicted scopes of the proposed model on the right side of Figure 5.7. We observe that the scopes coarsely follow the structure of the document, i.e. that the predicted boundaries are located at the beginning or the end of the different sections. It is worth keeping in mind that these scopes have only been supervised with the requirement that they contain or exclude certain mentions, and that no information regarding the precise location of their boundaries has been given.

Moreover we notice that the reading of these scopes gives a partial explanation of why some relations were predicted or not, whereas the outputs of relation prediction model are usually hardly explainable.

	Frame support			Frame label		
Frame type	P	R	F1	P	R	F1
BIRADS score	89.6	95.7	92.5	80.6	86.1	83.3
Breast density	84.9	96.9	90.5	82.6	94.3	88.1
Diagnostic procedure	82.1	91.7	86.6	74.0	82.7	78.1
Therapeutic procedure	86.2	87.1	86.6	68.3	69.0	68.6
Finding	74.0	82.4	78.0	59.6	66.5	62.9
Overall	81.1	90.0	85.3	68.7	76.2	72.2

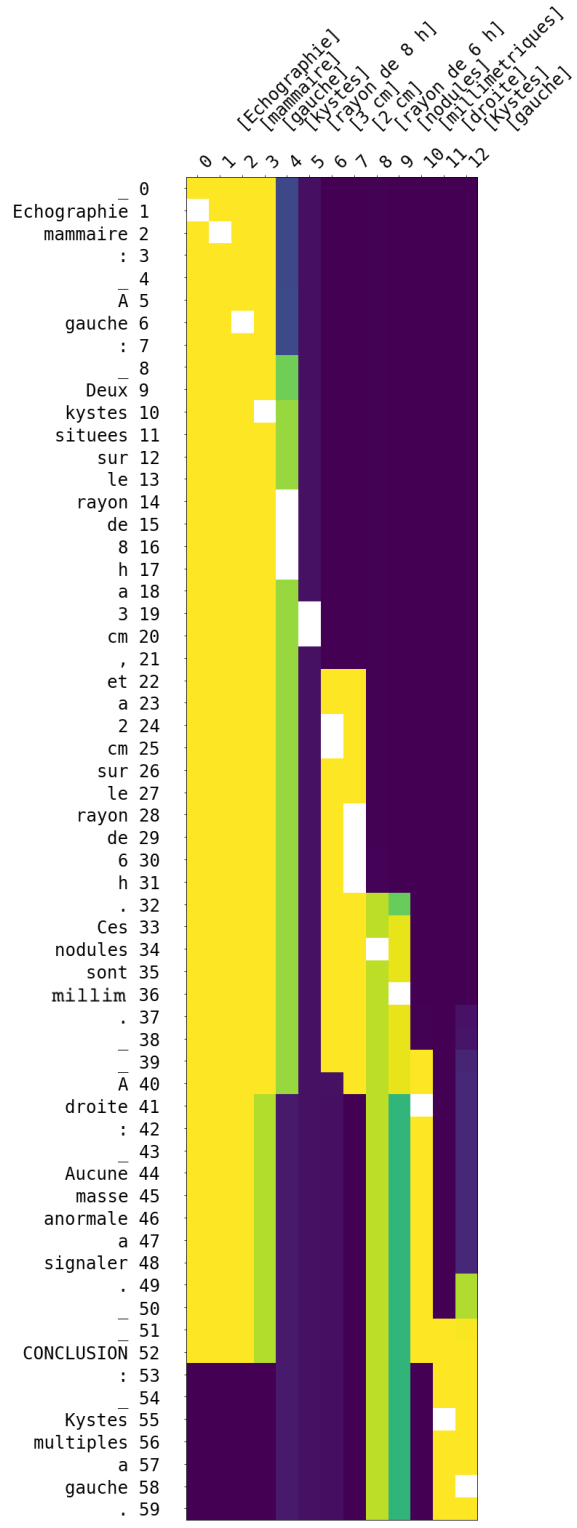
Table 5.7 Performance of the model at the frame level

	P	R	F1
Is mammography ?	88.5	100.0	93.9
Has passed surgery ?	63.6	87.5	73.7
Current BIRADS score	94.3	100.0	97.1
Current lateralized BIRADS score	92.0	92.0	92.0
Current breast density	89.3	96.2	92.6
Current lateralized breast density	86.4	95.0	90.5
Current lesion with quadrant	85.2	81.2	83.2
Current lesion with quadrant or radial position	77.9	77.9	77.9
Current lesion with quadrant or radial position & size	76.7	78.4	77.5

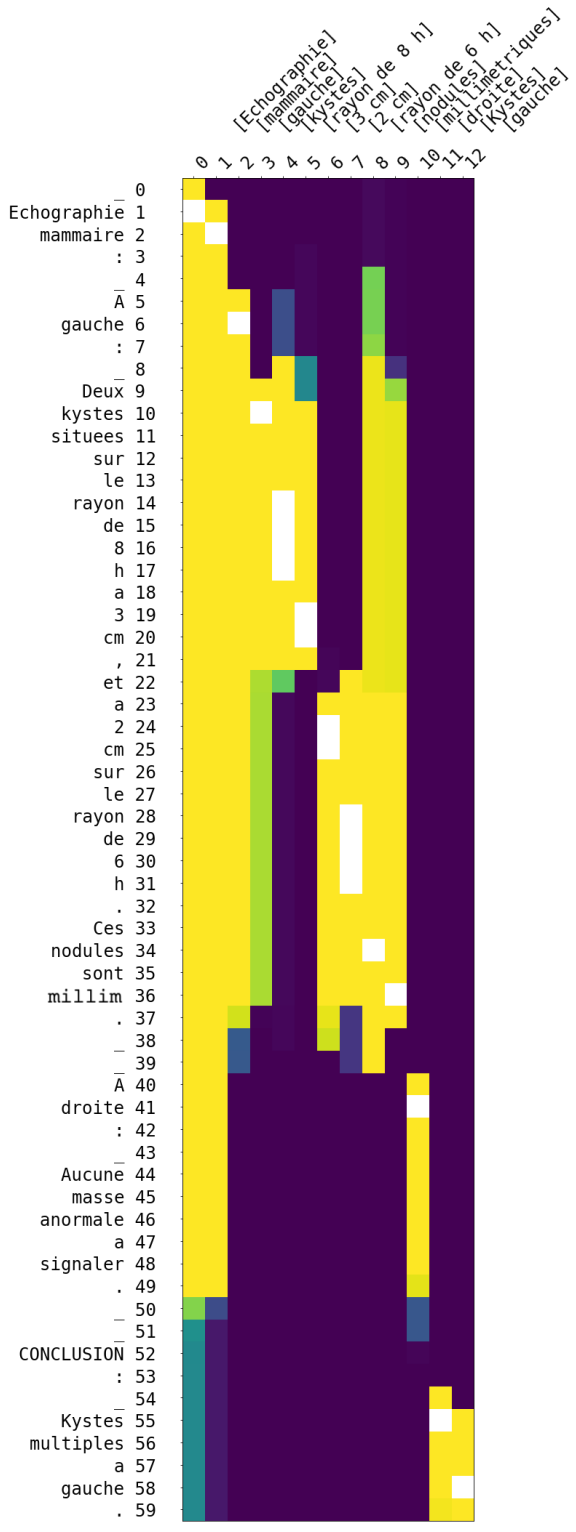
Table 5.8 Performance of the model against various queries

		Mention	Frame support	Frame label
Base		96.2	85.3	72.2
Neural net tricks	− input-residual	95.2 (−0.9)	83.9 (−1.4)	69.3 (−2.9)
	− relative attention	95.6 (−0.5)	84.0 (−1.3)	70.5 (−1.8)
Frame extraction	− relation heuristics	96.1 (−0.1)	85.4 (+0.1)	71.8 (−0.4)
	− WSS	96.1 (−0.1)	82.1 (−3.2)	69.5 (−2.7)
	− WSS − asymmetric	95.9 (−0.3)	74.4 (−10.9)	57.5 (−14.8)
	− scopes (only biaffine)	96.2 (+0.0)	80.4 (−4.9)	66.9 (−5.3)
Knowledge injection	− doc splitting (1)	96.1 (−0.0)	85.3 (+0.1)	71.5 (−0.7)
	− lexicon sentences (2)	95.4 (−0.8)	85.0 (−0.3)	70.8 (−1.5)
	− data augmentations (1+2)	95.4 (−0.8)	85.0 (−0.3)	69.9 (−2.3)
	− constraints	96.2 (−0.0)	84.0 (−1.3)	69.4 (−2.8)

Table 5.9 Ablation experiments on the model and training data. WSS stands for Word-level Scope Supervision. All reported metrics are F1-scores.



(a) Scopes inferred by a model trained with symmetric $R(u, v)$ supervision only



(b) Scopes inferred by the proposed model with asymmetric $r(u, v)$ supervision and word level supervision

Figure 5.7 Visualization of the predicted mentions and scopes on the example of Section 5.1.2. The vertical axis represents the words, and the horizontal axis represents the mentions. For each scope, the words contained in the corresponding mention are marked in white.

5.4.2 Model ablations

5.4.2.1 Impact of scopes

Table 5.9 shows the effect of ablating the model scopes. In this configuration, the model can only predict the relations through the biaffine model. We can observe that ablating scopes results in an overall loss of 5.3 pt for the Frame Label metric and 4.9 pt for the Frame Support metric. We believe that this is due to the inability of other neural components to reason with intervals, i.e., to answer queries such as "what word is between these two words". Scopes allow the model to focus on section or enumeration boundaries and leave interval logic reasoning to the scope CRF.

Given that scopes improve the quality of predictions, the question arises as to what kind of supervision is needed for to learn them. As shown in Table 5.9, when the scopes are learned directly using word-level partial annotations, the model performs better than with distant supervision on the $r(u, v)$ matrix. This suggests that finer manual annotation of scopes may benefit the system. If we directly supervise the symmetric matrix $R(u, v)$ instead of the asymmetric matrix $r(u, v)$, the performance collapses and we lose between 10 and 15 pt for the Frame metrics. This can be seen in the visualization of Figure 5.7: the scopes overlap several unrelated sections, which leads to the prediction of erroneous frames. The learning of scopes must be hindered by the uncertainty related to the supervision of this matrix alone and the small amount of data.

Interestingly, if we remove the relation supervision heuristic and let the model explore different configurations on its own, the performance remains on par with the proposed approach. Since these heuristics aim at injecting information about the hierarchy of mentions and the structure of the text, this suggests that the model is able to infer this information itself from "flat" annotations. This is a valuable finding because it suggests that complex, hierarchical, directed annotations for other tasks could be alleviated when it is easier to annotate groups of mentions than directed graph structures between mentions.

5.4.2.2 Impact of the gating mechanism

Table 5.4.2 shows the effect of the different gating mechanisms on the performance of the model. We can observe that the "input-residual" gating mode leads to a performance gain of 1.4 pt in Frame Support and 2.9 pt in Frame labels. Although this variant performed well in our experiments, more research is required to evaluate the reason behind this apparent better performance, and we did not investigate this mechanism further in this work.

5.4.2.3 Impact of the relative attention mechanism

We evaluated the effect of the added information on the relative position of the word-mention and mention-mention attention mechanisms. From the table 5.4.2, we can observe

that this added information leads to a performance gain of 1.3 pt of F1 frame support and 1.8 pt of F1 frame label. Without it, a mention is "positionally blind" and must rely on the inductive bias of the LSTM to find its neighboring words or mentions. Therefore, we expected a larger drop in performance, especially in the context of long documents. This suggests that the chain structure of the LSTM is capable of encoding relative position information at both the word and mention level. Nevertheless, relative attention proves to be an effective way to improve retrieval performance.

5.4.3 Data ablations

5.4.3.1 Impact of the size of the training data

Figure 5.8 shows the overall performance of the model when trained with different numbers of annotated samples. We can observe that the first 10 documents are critical, and expectedly that the added value of additional documents becomes lower as their number increases. On one hand, we can note that our system requires only a small amount of documents to achieve "correct" accuracy, i.e., it can be used to pre-annotate more documents. This "data efficiency" is important when tackling new domains in order to allow quick feedback and possible changes regarding the annotation scheme. However, given the complexity of the task and the evolution of performance with the training set size, we also note that a larger number of annotated documents might be needed to approach a perfect score.

5.4.3.2 Impact of the augmented samples

We remove the augmented samples from the training data and show the effect on performance in Table 5.9 and Figure 5.8. We observe that adding synthetic sentences only slightly helps improving the model mention detection performance (+0.3 pt). However, this improved performance has a larger effect of 1.5 pt on the Frame Label metric. This is typical of the phenomenon of error propagation. Indeed, a missing or mislabelled mention can have an effect on multiple frames. This shows the importance of focusing efforts on the first steps of pipeline models such as ours.

As we reduce the number of annotated documents in the training set, the effect of augmentation becomes more important, and with only 4 annotated documents we obtain an average performance of 89.4 F1 in mention extraction versus 81.1 F1 without, and an average performance of 45.7 F1 in Frame Label F1 versus 34.7 without. Finally, we can see that a model trained with synthetic sentences only retrieves most of the annotated mentions, which is valuable when tackling a new domain. The non-zero Frame metrics can be explained by the presence of singleton frames that contain only one mention, and by the Frame classification constraints that prevent the system from predicting impossible label combinations.

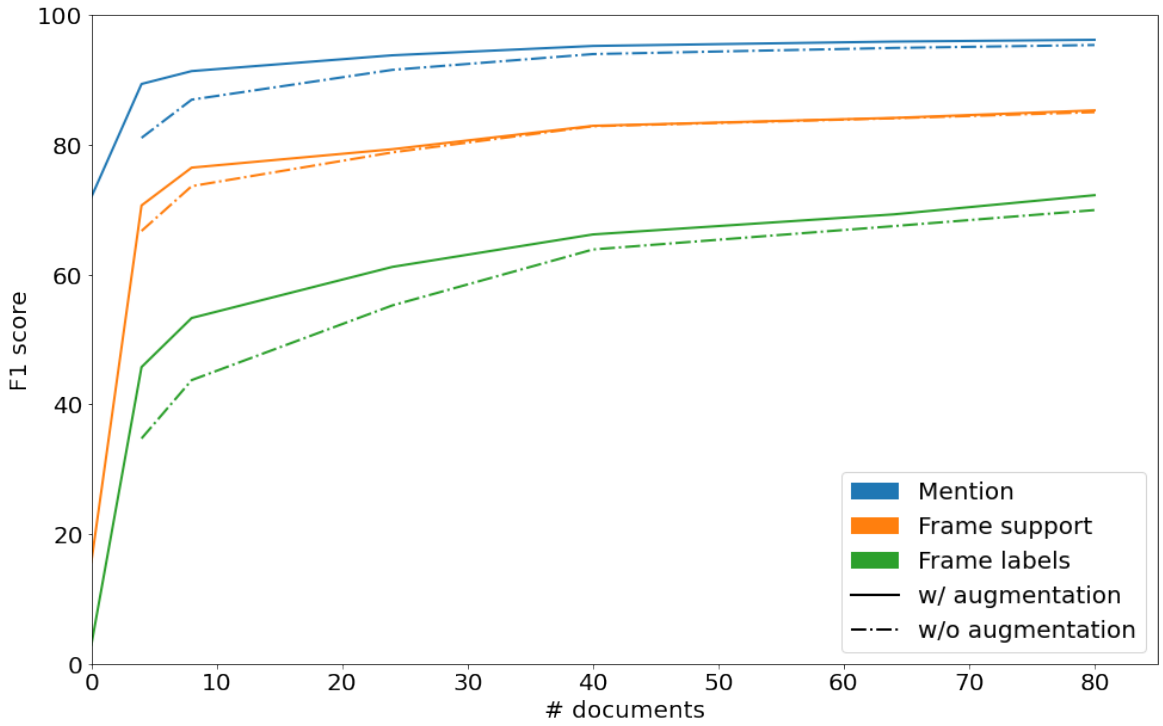


Figure 5.8 Plotted evolution of the F1 scores with the number of annotated documents. The plain lines show the performance with data augmentation (synthetic sentences and document splitting), while the dashed lines show the performance without augmentation.

5.4.3.3 Impact of the constraints

We train the model without the constraints described in section 5.2.6. In this configuration, the model learns that each pair of mentions is legal. However, we still apply these constraints during the evaluation phase to avoid illicit predictions. This amounts essentially to applying only post-processing on the predictions. We observe in Table 5.9 that removing these rules leads to a loss of 2.3 pt in the Frame label F1-score and 1.3 pt in the Frame support F1-score. This can be explained by the fact that the model has to "learn" the annotation scheme and its inevitable imperfect representations of the reports. These constraints can also help the model focus on the actual uncertainties of the task, and leave what is already known to the modeled constraints.

5.5 Limitations

In this section, we review some of the limitations of our method.

Annotation bias First, the model was developed in parallel to the data annotation. While this approach allowed us to better fit our initial objective, this biases the performance of

the model such that it obtains a correct performance on the annotated data, but might not generalize as well on other tasks.

Scheme granularity Second, we did not annotate the radiological lesions using a fine-grained scheme and left that disambiguation task for future works. This again might overestimate the performance of the model on these radiological lesions, since we do not distinguish between tumors and cancer diagnosis. Moreover, we did not annotate other relevant aspect of tumors such as their size trend, morphology, or margin.

Coreferences Third, we did not address the issue of coreferences, which are nevertheless important in the extraction of information from clinical documents, since they allow us to obtain a deduplicated list of entities, to fill in possible missing fields, and to perform a final evaluation of the extraction independently of the intermediate annotation choices. This step will be the focus of future work, together with the problem of cross-document coreference to link objects across multiple reports.

5.6 Conclusion

In this chapter, we proposed an annotation scheme and a system for extracting structured entities from clinical breast radiology reports. We trained and evaluated our method on a new dataset of 120 annotated documents. Although these documents are not made public for medical privacy concerns, this dataset can be used to evaluate the performance of future systems and developments in the field of clinical NLP. In particular, the pre-training of specific encoders for the French clinical domain and for long documents should greatly benefit our system. We have shown that the addition of synthetic sentences can improve the performance in the context of a small amount of data. This information is valuable for the annotation and development of new information retrieval systems in other domains, where key words or phrases are known in advance. The method we described introduces the notion of frame extraction in the form of mention cliques, and we have shown that a formulation of the relation extraction task via scopes improves the performance of our system. Future work will evaluate this approach on other structured entity extraction tasks such as event extraction.

Chapter 6

Conclusion and perspectives

Contents

Structuring medical documents is a complex task that is related to several NLP research topics. This thesis presented several contributions to the extraction and normalization of simple and structured entities. This chapter makes a brief summary of the thesis (Section 6.1) and discusses future research directions (Section 6.2).

6.1 Summary

In our work on nested named entity recognition, we introduced two methods to handle the extraction of overlapping entities. In particular, we showed that sequence labelling methods are better suited for the extraction of long and ambiguously annotated entities when exact boundaries are not required, and we discussed several aspects of the design of these systems. We also showed how ensembling can improve the performance of a NER model.

We also addressed the issue of training models in languages other than English. More specifically, in Chapter 4, we demonstrated the importance of training French and English jointly in the case of medical concept normalization, and even the benefit of training a single multilingual model, instead of several bilingual models. We evaluated all the models proposed in this thesis on French datasets, and annotated a new corpus of French clinical radiology reports in Chapter 5.

In the case of structured entities in Chapter 5, we proposed a new frame-based annotation scheme, and designed a method to automatically extract these entities from unlabelled reports. We also introduced the concept of mention cliques to overcome the issue overlapping structured entities, as well as a new mechanism of relation prediction with mention scopes. We showed how these "scope-relations" both improve the performance of our system on clinical documents, and provide partial explanation of the predicted relations between mentions.

Finally, we also developed multiple techniques to inject external medical knowledge into the training of learning algorithms, while alleviating the need for language or domain specific pre-processing methods and reducing the requirement for annotated data. In Chapter 4, our proposed model obtained good results without any annotated normalization sample. In the context of radiological entity extraction in Chapter 5, we showed that the hybridization of a set of output constraints, a terminology and a learning based method enabled our method to be effective with a small number of training documents.

6.2 Future research directions

Starting from the work presented in this thesis, several research directions arise.

6.2.1 Deeper hybridization between learning and symbolic models

In Chapter 5, we saw how a structured entity extraction task could be represented by an enumeration of compatible concepts. However, the number of legal combinations (2502) remained tractable and could be baked into the model without becoming an issue. More complex schemes could lead to a larger number of combinations, making their enumeration infeasible. One solution to overcome this problem is to directly represent the allowed outputs by logical propositional formulas and model them with a CRF (Lafferty et al., 2001). For example, Deng et al. (2014) used a CRF to model subsumption and exclusion relations between labels to improve image classification.

Further along this path, the integration of first-order logic into retrieval models is an exciting perspective. Indeed, when relations are added to the retrieval scheme, modeling the logical interactions between objects could improve performance. Markov logic networks (Domingos and Lowd, 2009) unify symbolic and learning-based methods, and are a promising avenue for integrating symbolic reasoning into information extraction models. For example, we saw how implicit attributes could be inferred from other attributes, such as the organ in the case of a mammogram. However, all these entities require the presence of a trigger and implicit entities are out of scope of the proposed model. One could therefore imagine conditioning the existence of a current lesion on the presence of a (possibly implicit) current

diagnostic procedure by modeling the following formula:

$$\begin{aligned} &\exists \text{Lesion s.t. lesion\_frame}(\text{Lesion}, \text{temp:overlap\_exam}, ...) \\ &\implies \exists \text{Diag s.t. diag\_proc\_frame}(Diag, \text{temp:overlap\_exam}, ...) \end{aligned} \tag{6.1}$$

6.2.2 Multilingual and multitask training

We saw in chapter 4 how joint training on multiple languages benefits a normalization system. Recent work has shown how a unified training on multiple named entity datasets improves the performance of a NER system. Since resources for medical entity normalization are scarce, a promising approach is to train a normalization system on multiple datasets and multiple languages to achieve a robust, multilingual normalization system. To go further, we can also consider a multi-task training of normalization, NER and structured entity extraction systems. Moreover, we have shown how a pre-trained system can handle more concepts by being partially re-trained in a second phase. The reverse direction can also be considered, i.e., pre-training a normalization model on a large amount of concepts, and fine-tuning it on a smaller number to better fit the target domain.

6.2.3 Interactively programmable annotation software

As mentioned in Chapter 5 Section 5.1.4, the choice of annotation software must be taken into account in the design of the annotation scheme. For example, it is difficult to annotate implicit entities in Brat or to annotate relations on multiple lines, and impossible to handle multiple documents at once. There are many annotation tools available (Neves and Ševa, 2021), but most of them are either proprietary, poorly adapted to document or patient annotation, require a complex installation that is not compatible with existing remote work environments, or are difficult to customize. Finally, the standardization of annotation levels (mention / relation / event) is an obstacle to the development of new tasks. Given these limitations, we started to develop *Metanno* (illustrated in Figure 6.1), a dynamically programmable annotation software integrated to the popular Jupyter IDE.

We list here some of its features:

— ease of installation as a Jupyter extension
— joint annotation of multiple reports (cross-document co-referencing)
— visualization of annotations at the level of a document, patient or corpus in Excel-style dynamic tables
— bidirectional communication between the Python kernel and the front-end to facilitate the integration of active learning algorithms
— simple Python API to modify the behavior of the software when clicking a button, selecting annotations, highlighting table rows

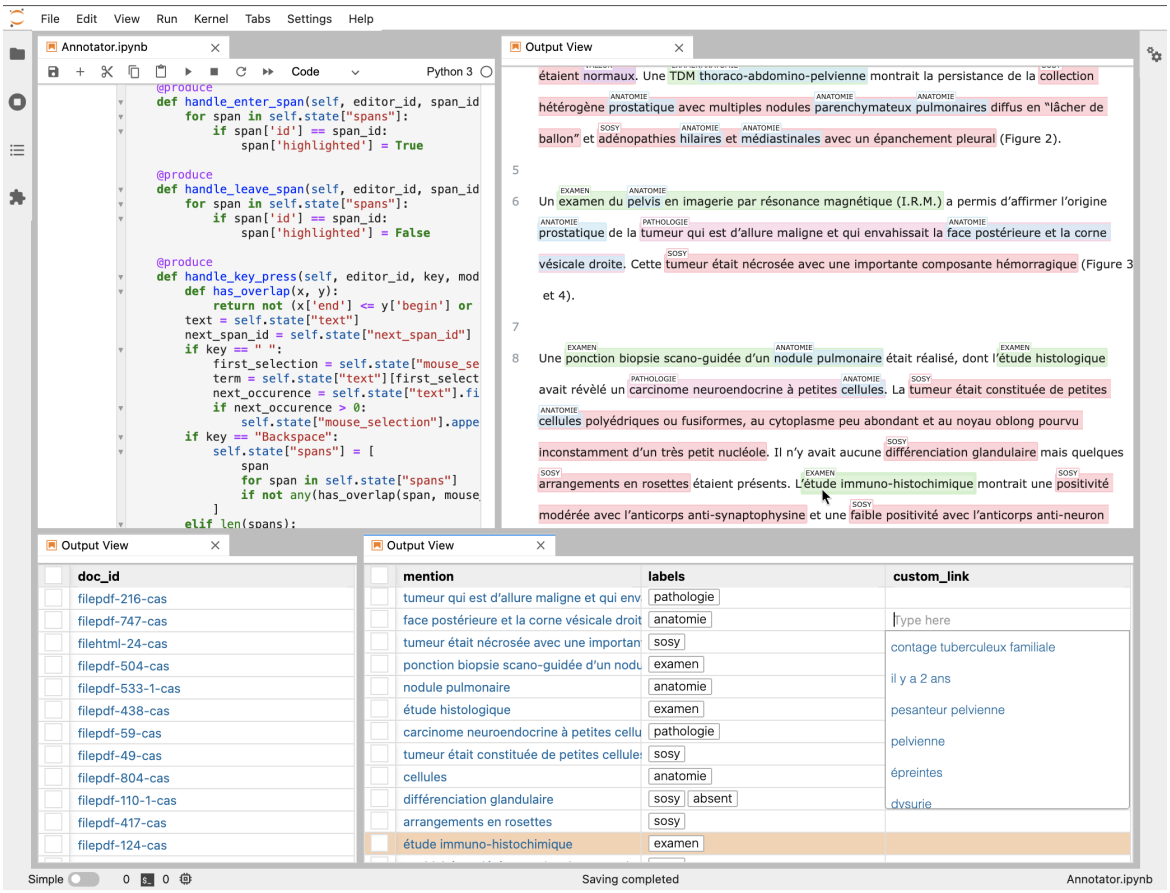


Figure 6.1 Metanno annotation software

— large range of possible actions such as batch modification or selection of annotations
— annotation of any type of entity that can be represented as tables.

This project did not come to fruition in time for the annotation of the Chapter 5 corpus, but we plan to open-source a first version in the upcoming months. We believe that this project could have significant repercussions on future research. A more detailed explanation of the software can be found in Appendix B.

6.2.4 Structured entity centric pre-training

In addition to the data augmentation and logic modeling discussed earlier, incorporating inductive biases and biomedical structuring task-specific goals into text encoding models could be beneficial to the development of information extraction models.

We saw in Section 5.1 that the concept of scopes was a useful and intuitive mechanism for improving detection of relations between mentions. We also saw how incorporating mechanisms such as relative attention improves the detection of relations between mentions. One avenue for improvement is the integration of scopes into model attention to allow for a

broader range of query types that the model can rely on to compute its representations. To the best of our knowledge, no current attentional mechanism can formulate an attentional query of the type "what are the words/mentions after me and before the next line break?"

Another avenue for improvement is the development of pre-training that promotes the representation of simple entities. Several studies have been conducted on the improvement of pre-training objectives to better take into account the entities in the text representation models (Joshi et al., 2019; Lin et al., 2021; Yamada et al., 2020) and most pre-trained models focus on whole sentence representations. In particular, to the best of our knowledge, entity-centered pre-trainers that handle both contextualized and context-free entity representations have not yet been studied. As an example, a key aspect of taking context into account in medical entity normalization is the distributional shift between contextualized entity representations and non-contextualized entity representations such as those present in terminologies. To overcome this problem, our approach in Chapter 4 was to "cut out" the entities in the medical texts, while we chose to augment the synonyms with an artificial context in Chapter 5. Nevertheless, these are not elegant solutions. Just as the pre-training of models like BERT or ELMO have improved the performance of many NLP tasks, the pre-training of a model that also takes entity representations into account should benefit the improvement of information extraction systems.

Appendix A

Relaxed retrieval metrics

Unlike the exact match NER metric for which a true positive is unambiguously counted when two elements of the predicted and gold entities match, defining and computing relevant metrics between more complex sets of objects becomes more difficult as the number of element attributes increases. One option is to lower the minimum similarity threshold required between predicted and gold features to account for small errors such as mismatch between mention boundaries. However, this leads to ambiguities in the metric computation, since several predicted elements may match a single gold element, and vice versa. We explicitly formulate a greedy matching procedure to compute a maximum bipartite greedy match between the elements of two sets, in the algorithm 1 to avoid double counting true positives.

For reference, the exact match metric NER is written using this matching procedure in the Algorithm 2.

The NER metric for the section 3 uses a score function that returns 1 if the Dice overlap of words in two mentions is higher than 0.5. The procedure is described in the Algorithm 3.

The matching procedure is used in the computation of the frame support metric in Chapter 5 (Algorithm 4), where two frames have a non-zero match score if some of their mentions overlap, and a perfect score if all their mentions overlap, and 0 otherwise. This score between 0 and 1 is the Dice/F1 overlap between the mentions of the two frames. It is used as a "relaxed" true positive when computing the retrieval metrics.

The matching procedure is used in the calculation of the frame label metric in Chapter 5 (Algorithm 5), where two frames have a matching score of 1 if their labels match and their trigger mentions overlap, and 0 otherwise. This score is used as a true positive when computing retrieval metrics.

Algorithm 1 Procedure to compute the maximum sum of greedily matched items between two sets of predicted and gold items

```
▷ greedily matches elements between two sets P and G to maximize the sum of the bipartite matching according to the MATCH_SCORE function
function MATCH_SUM(P, G, MATCH_SCORE)
    scores ← empty matrix                                   ▷ match scores between P and G
    matched ← {}                                            ▷ matched predicted and gold entities
    result ← 0                                              ▷ the aggregated score
    for each predicted item p ∈ P do
        for each gold item g ∈ G do
            scores[p, g] ← MATCH_SCORE(p, g)
    while there remains both gold and predicted entities not matched do
        Take the first remaining predicted entity p ∈ P\matched
        g ← argmax(scores[p])                               ▷ find the best matching g ∈ G
        if scores[p, g] > 0 then
            result ← result + scores[p, g]
            matched ← matched ⋃ {p, g}
    return result
```

Algorithm 2 Procedure to compute the Exact NER metric

```
function EXACT_NER_MATCH_SCORE(p, g)
    ▷ return 1 if p and g have the same boundaries and label, 0 otherwise
    return p.begin = g.begin and p.end = g.end and p.label = g.label

function EXACT_NER(P, G)
    ▷ return the retrieval metrics, where true positives between P and G are computed with EXACT_NER_MATCH_SCORE
    tp ← MATCH_SUM(P, G, EXACT_NER_SCORE)
    precision ← tp/|P|
    recall ← tp/|G|
    f1 ← 2 · tp/(|G|+|P|)
    return (precision, recall, f1)
```

Algorithm 3 Procedure to compute the Half NER metric

function HALF_NER_MATCH_SCORE(p, g)
    ▷ *return 1 if p and g have a word dice overlap $\geq$ 0.5 and the same label, 0 otherwise*
    return $2 \cdot |\text{p.words} \bigcap \text{g.words}| \;/\; (|\text{p.words}| + |\text{g.words}|) > 0.5$ and p.label = g.label

function HALF_NER(P, G)
    ▷ *return the retrieval metrics, where true positives between P and G are computed with HALF_NER_MATCH_SCORE*
    tp $\leftarrow$ MATCH_SUM(P, G, HALF_NER_SCORE)
    precision $\leftarrow$ tp/|P|
    recall $\leftarrow$ tp/|G|
    f1 $\leftarrow 2 \cdot$ tp/(|G|+|P|)
    return (precision, recall, f1)

Algorithm 4 Procedure to compute the Frame Support retrieval metrics

function SAME_TYPE_OVERLAP(a, b)
    ▷ *return 1 if a and b share $\geq$ 1 word and have the same label, 0 otherwise*
    return $|\text{a.words} \bigcap \text{b.words}| > 0$ and a.label = b.label

function FRAME_SUPPORT_MATCH_SCORE(p, g)
    ▷ *return the Dice overlap between p mentions and g mentions which is 0 if there is no overlap and 1 if all mentions of p and g match*
    tp $\leftarrow$ MATCH_SUM(p.mentions, g.mentions, SAME_TYPE_OVERLAP)
    return $2 \cdot \text{tp}/(|\text{g.mentions}| + |\text{p.mentions}|)$

function FRAME_SUPPORT(P, G)
    ▷ *return the retrieval metrics, where (relaxed) true positives between P and G are computed with FRAME_SUPPORT_MATCH_SCORE*
    relaxed_tp $\leftarrow$ MATCH_SUM(P, G, FRAME_SUPPORT_MATCH_SCORE)
    precision $\leftarrow$ relaxed_tp/|P|
    recall $\leftarrow$ relaxed_tp/|G|
    f1 $\leftarrow 2 \cdot$relaxed_tp/(|G|+|P|)
    return (precision, recall, f1)

Algorithm 5 Procedure to compute the Frame Label retrieval metrics

function FRAME_LABEL_MATCH_SCORE(p, g)
    ▷ *return 1 if all labels of g are in p, all labels of p are in g or a non conflicting frame of the same object and the triggers overlap, 0 otherwise*
    return p.labels $\subseteq$ g.object.shared_labels and p.labels $\supseteq$ g.labels
        and $|$p.triggers.words $\bigcap$ g.triggers.words$| > 0$

function FRAME_LABEL(P, G)
    ▷ *return the retrieval metrics, where true positives between P and G are computed with FRAME_LABEL_MATCH_SCORE*
    tp $\leftarrow$ MATCH_SUM(P, G, FRAME_LABEL_MATCH_SCORE)
    precision $\leftarrow$ tp/$|$P$|$
    recall $\leftarrow$ tp/$|$G$|$
    f1 $\leftarrow 2 \cdot$ tp/($|$G$|+|$P$|$)
    return (precision, recall, f1)

Appendix B

Metanno: a programmable and modular annotation software

Annotation tools are essential to the development of new information retrieval tasks and models and have been the focus of many development efforts for several years (Neves and Ševa, 2021). We considered three NLP tasks in this thesis: the first, named entity recognition, benefits from many existing annotation tools. The second task is the normalization of medical entities, and requires specialization of tools to speed up labeling and pre-filter the list of candidate concepts. Such specializations can be found in some of the softwares like BRAT, Webanno, prodigy and others. However, the third task of frame extraction did not fit well into the BRAT framework, mostly due to the long-range relationships between named entities. Other tools, such as GATE or the XConc Suite, allow for long relationships through tables and are customizable to some extent, but with minimal to no web support, and these customizations require a substantial amount of work. Overall, we could find no free web-based software with sufficient customization and support for long range dependencies.

B.1 Rationale

Our first observation is that complex custom tasks require specific annotation tools, and no existing software provides sufficient customization features. This may lead to either modifying the ideal annotation scheme to fit existing software and forgoing some annotations, or making the scheme more complex. There are many annotation tools available, but most of them are either proprietary or ill-suited to annotating documents or multi-documents, require complex installation that is not compatible with existing remote working environments, or are difficult to customize.

A second observation is the gain in popularity of the Python language, its simplicity for scripting and its integration into collaborative web IDEs like Jupyter. As a result, the integration of Python into an annotation tool to more fully control its behavior and interact with its inputs

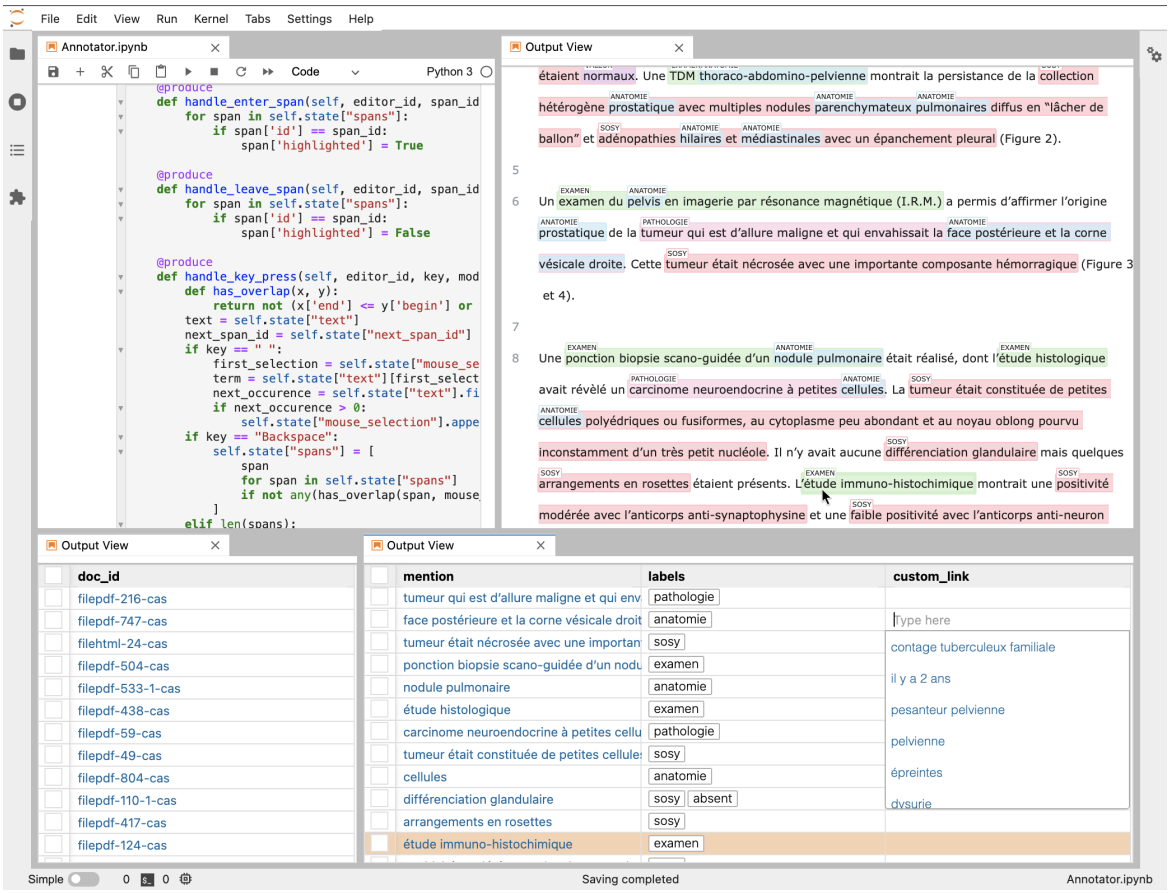


Figure B.1 Example of the Metanno software for named entity recognition with a custom relation column

and outputs from a Python kernel has become possible. This is in contrast to most tools that prefer a more rigid configuration system that does not require programming.

Finally, pre-annotation and integration of machine and user feedback into the annotation process has become a crucial requirement for bootstrapping and accelerating new retrieval tasks. Active learning requires a tight coupling between the model and the interface, and we believe it is essential to anticipate these needs when creating future annotation tools.

Given these observations, the Metanno project was initiated to enable the development of comprehensive and highly customizable annotators through a simple Python API.

B.2 Modelisation

We define some goals for our ideal tool.

Fast software response time Software response times should less than 100ms to allow a "fluid" user experience, without noticeable delay (Card et al., 1983). This should also be the case for unreliable connections, with which web-based annotators like BRAT are not robust.

One programming language Most data science programs are done in Python, and this language has been taught to students for some time. This makes it a candidate of choice to interact with the software.

Completeness Structured data can be easily represented in relational databases with a set of tables. For example, text classification requires only one table for documents. Named entity annotation requires two tables for documents and entities. Frame annotation requires three tables (documents, entities, frame) for example. The example in Figure screenshot shows a possible structure for a named entity annotator with an additional column per entity for relationships. Since most data scientists are used to working with tabular data such as Excel, support for tabular views seemed both natural and necessary to meet most data annotation requirements.

Interactivity Finally, the software should be interactive, both for developing the annotator and for manipulating the input and output data. The Jupyter notebook scheme is ideal for this, and customizations (what happens if the user clicks on an entity, or hovers over it) should be taken into account immediately, without the need to recompile Jupyter, or restart the Python core.

B.3 Workflow

All the app is controlled by a single class instance and all the displayed data is gathered as a single json-like state, replicated on both the client and the Python kernel. Each view rendered in Jupyter, either a text view or a tabular view, uses a derivation of this state (`view_data = fn(app_data)`) and calls functions in the app class whenever an event occurs. An overview of the software workflow can be found in Figure B.2.

Immutable state Every state mutation is recorded by proxying the state, which enables undo/redo operations. This also allows to send patches instead of the full state when the client or the kernel produces a mutation to keep client and kernel states replicas in sync. This immutable paradigm has been popularized by Javascript libraries like Redux and Immer.

Client-kernel communication To avoid having to open a new port, which can slow down the integration if the user does not have the Jupyter environment, all communication between

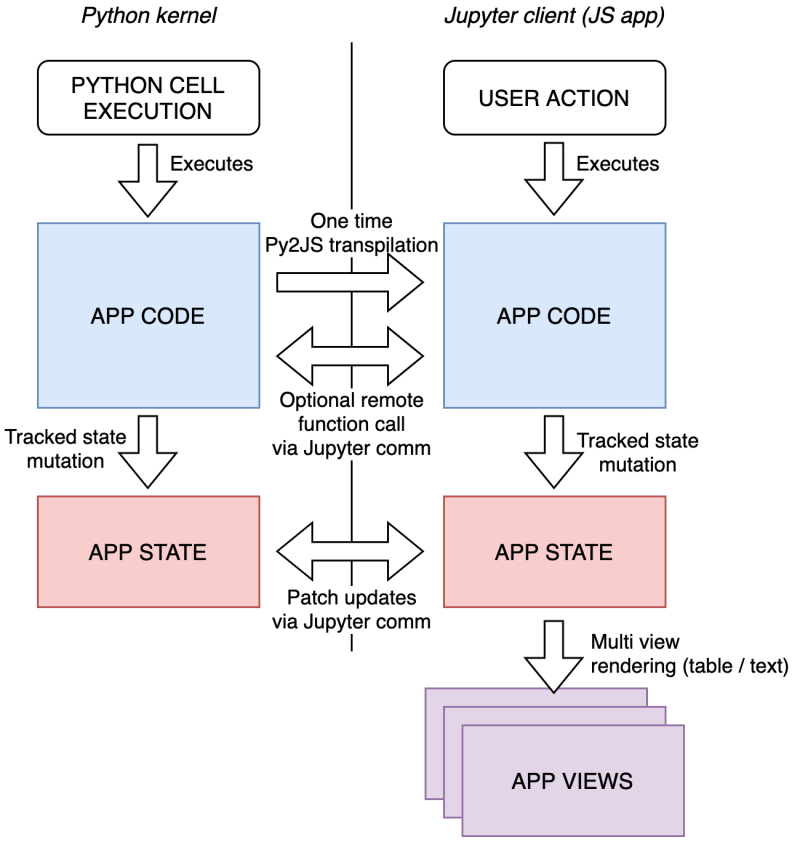


Figure B.2 Overview of the workflow of the annotator

the client and the kernel is done using the Jupyter web-socket. This web-socket is used to transmit remote function calls, state patches and the transpiled app class code.

Python to Javascript transpilation This app class is written in Python by the user and automatically translated into Javascript using the Transcrypt software. This javascript code is sent to the front-end such that every action taken by the user is answered immediately whenever possible. If an action must be executed in the kernel (like file saving) or the client (like scrolling a view to a given annotation), the user can wrap a given function with a specific Python decorator (`@frontend_only` or `@kernel_only`).

Two kinds of views On the client side, the widgets are built in React with state selectors written in Python (and transpiled with Transcrypt). A first widget is the text view renderer, which allows to visualize and annotate multi-line and/or nested text spans on a given text. The second widget is the table widget, based on react-data-grid. Different types of data types are supported like text, hyperlinks, lists of text and list of hyperlinks, which suffice to annotate named entities, relations, events or frames. Custom input suggestions can be provided using the app shared state for each column.

B.4 Perspectives

A first version of the software is available at https://github.com/percevalw/metanno. Much work remains to be done, including providing documentation and examples, more traditional Excel-like functionality for tabular views, visualization of relationships in text, and support for more data types, such as images or PDFs.

Résumé étendu

Extended French summary

Les documents cliniques hospitaliers (comme les rapports d'hospitalisation ou de consultation, les comptes rendus, les rapports de radiologie, les rapports d'anatomo-pathologie, les transmissions infirmières, les lettres de sortie et les prescriptions, ou encore les lettres des médecins) constituent des documents riches en informations pour diverses applications telles que le recrutement de patients pour la recherche clinique, la surveillance épidémiologique, le codage médical et les outils d'aide à la décision (Wang et al., 2018c). Ces documents sont essentiellement rédigés en langage naturel, qui se prête bien à une description exhaustive et exacte des informations, permet de détailler les cas particuliers et facilite la saisie des informations. On estime ainsi que plus de 80 % des données hospitalières sont collectées sous forme textuelle (Raghavan et al., 2014). Malheureusement, le texte libre ne se prête pas facilement aux traitements informatiques standard. En revanche, les représentations structurées améliorent la qualité et la réutilisation des données des patients pour les soins cliniques (y compris l'aide à la décision), l'audit et la recherche cliniques, le codage médical pour l'allocation des ressources et la gestion des services de santé. Nous nous intéressons à la structuration automatique de documents textuels. Cette discipline, communément appelée extraction d'information (IE) dans le traitement automatique du langage (TAL), englobe de nombreux domaines de recherche.

Structuration La structuration est le processus de transformation d'un échantillon de texte libre en une vue organisée des informations qu'il contient. L'échantillon de texte peut être une seule phrase, un paragraphe, un rapport entier ou même un dossier de patient contenant plusieurs rapports. Ces représentations structurées peuvent prendre différentes formes, comme l'illustre la Figure B.3. Dans le cas de d'une classification, nous pouvons attribuer à chaque échantillon une étiquette unique à partir d'une liste prédéfinie, telle que le type de rapport, le sexe d'un patient, ou une réponse oui/non à une question. La classification multi-étiquette permet de classer les échantillons avec plusieurs étiquettes, comme le type de rapport et un score de risque de cancer s'il s'agit d'une mammographie. Un autre type de structure

se concentre sur la notion d'entité. La reconnaissance d'entités vise à extraire un nombre variable d'objets, comme par exemple les lésions observées dans un rapport de radiologie. Les différentes entités sont généralement mentionnées explicitement dans le texte par un mot-clé ou une expression, mais elles peuvent aussi être composées de plusieurs morceaux ou être implicites. Comme pour les tâches de classification, chaque entité peut être caractérisée par une ou plusieurs étiquettes. L'extraction d'entités a été le sujet de nombreuses études depuis plusieurs décennies, et de nombreuses solutions ont ainsi été proposées. La tâche bien connue de reconnaissance d'entités nommées (① dans la figure B.3) correspond à l'extraction de mention d'entités ayant un début, une fin et une seule étiquette. Cependant, l'extraction d'entités imbriquées ou se chevauchant fait toujours l'objet de recherches actives. De plus, l'extraction d'entités plus exotiques, contenant plusieurs étiquettes et/ou plusieurs parties (③ dans la figure B.3), est encore loin d'être résolue, malgré la pertinence de ces entités dans certains domaines comme pour l'extraction d'informations cliniques. Nous appellerons ces entités des entités structurées, par opposition aux entités nommées classiques. Les étiquettes elles-mêmes peuvent être définies spécifiquement selon la tâche à accomplir ou empruntées à des bases de concepts médicaux. Le processus d'appariement entre entités et les concepts de ces bases est appelé normalisation (② dans la figure B.3). Ces bases de données sont souvent riches en informations: les ontologies fournissent des relations entre les concepts, et les terminologies fournissent des synonymes pour définir ces concepts et les identifier dans le texte. Leur utilisation favorise l'interopérabilité entre les systèmes.

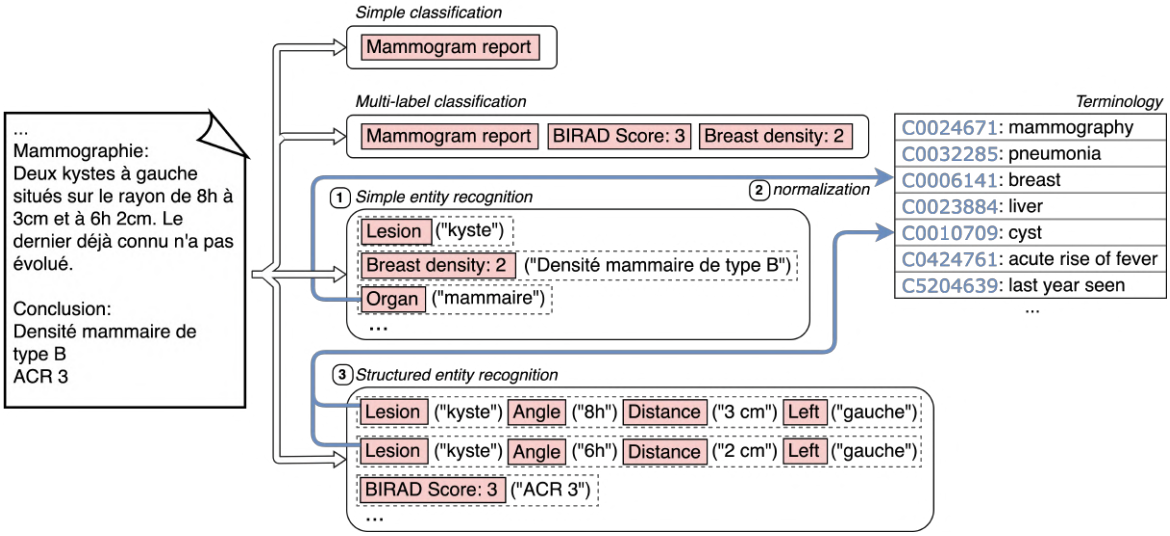


Figure B.3 Vue d'ensemble des différents objectifs de structuration, avec normalisation des concepts

Défis liés à la supervision de l'apprentissage Au cours des dernières décennies, le besoin d'analyse de documents médicaux, couplé à la croissance rapide des entrepôts de données

de santé et au nombre croissant de publications scientifiques biomédicales, a conduit au développement d'approches de TAL dans les domaines général et biomédical. L'avènement de l'apprentissage automatique, en particulier l'apprentissage profond, s'est accompagné de la promesse de décrire une tâche à l'aide d'exemples à partir desquels généraliser, plutôt que de construire des règles spécifiques à un domaine et à une langue. Ces méthodes sont devenues très populaires et ont démontré leur supériorité dans un grand nombre de domaines. Toutefois, les possibilités offertes par ces méthodes se sont également accompagnées d'un besoin critique de données annotées: de nombreuses méthodes d'apprentissage modernes entrent dans la catégorie de l'apprentissage supervisé, c'est-à-dire qu'elles nécessitent la création d'un ensemble de données annotées (par des experts humains) pour permettre l'apprentissage d'un modèle qui peut ensuite être appliqué à de nouvelles données. Le coût temporel de l'annotation des documents et les besoins élevées en annotations des approches par apprentissage profond représentent un obstacle à l'automatisation de l'extraction d'informations. Cependant, dans de nombreux cas, il existe des ressources de connaissances médicales auxiliaires, telles que des terminologies, qui ne se présentent pas sous la forme d'exemples annotés. L'injection de ces connaissances dans les modèles d'apprentissage fait encore l'objet de recherches actives. De plus, le processus d'annotation lui-même est loin d'être trivial. En effet, la conception d'un schéma qui concilie simplicité, expressivité et cohérence est un défi en soi.

Traitement du langage clinique français Les difficultés liées au traitement du langage naturel sont nombreuses. En effet, le langage naturel est sujet à des ambiguïtés sémantiques et syntaxiques. Comme tout document écrit, un rapport clinique peut contenir des fautes d'orthographe, des erreurs grammaticales, voire des contradictions. De plus, l'informatisation de ces rapports et leur conversion vers/depuis des formats portables (par exemple PDF) peuvent introduire des artefacts difficiles à traiter informatiquement. Outre ces "erreurs", la compréhension du langage naturel des rapports cliniques nécessite un certain sens commun, ainsi que de connaissances médicales. Il est fréquent de rencontrer des termes qui ne font pas partie d'aucune des ressources fournies à la machine, et ce malgré le nombre considérable de synonymes présents dans les terminologies évoquées précédemment. Lors du développement de modèles, en particulier dans le domaine clinique, il faut également tenir compte de structures linguistiques spécifiques telles que les conjonctions elliptiques, ou la segmentation hiérarchique des relations. Malgré les améliorations récentes des modèles de langage naturel, la compréhension automatique du langage, et, a fortiori, des documents cliniques en français, est encore loin d'être résolue. L'anglais dispose de beaucoup plus d'outils de traitement et de ressources terminologiques que les autres langues, et les approches anglaises ne sont pas toutes directement transposables au français par exemple. De plus, bien qu'il existe de nombreux travaux en français sur le TAL dans le domaine général, mais bien moins dans le domaine biomédical (Névéol et al., 2018). À titre d'exemple, bien qu'étant la cinquième langue la plus représentée dans la version 2019 de la terminologie UMLS, le français ne dispose de

synonymes que pour 3,5% des concepts présents. Par conséquent, un aspect important de ce travail est le développement de méthodes pour le TAL clinique en français.

Une étude de cas Nous abordons la tâche de structuration de rapports de radiologie. Cette étude s'inscrit dans le cadre du projet EZMammo, dont l'objectif principal est d'optimiser l'entrepôt de données cliniques de l'Assistance Publique des Hôpitaux de Paris (APHP) et de valider les prédictions d'un algorithme d'analyse de mammographies par apprentissage profond. Une tâche préliminaire de cette évaluation consiste à construire un jeu de données de mammographies étiquetées avec le diagnostic de cancer et les lésions trouvées dans les rapports correspondants. Dans le cas de lésions suspectes, l'examen radiologique est suivi d'une analyse cytologique. Il faut alors faire correspondre les résultats des deux rapports pour étiqueter la mammographie originale avec le diagnostic définitif. Ce traitement implique qu'il soit possible d'extraire des comptes rendus plusieurs entités médicales (comme des procédures, scores ou lésions) et leur caractéristiques spatiales, temporelles et morphologiques. Ces extractions peuvent ensuite être utilisées pour filtrer et aligner les résultats entre les images radiologiques, les rapports de mammographie et les rapports d'anatomo-pathologie. Comme nous le verrons, les entités à extraire se composent de plusieurs étiquettes et de plusieurs parties textuelles, et entrent donc dans la catégorie des entités structurées. Cette tâche d'extraction d'entités structurées se compose de plusieurs sous-tâches, à savoir l'extraction d'entités nommées pour localiser les mentions d'objets et leurs caractéristiques, la normalisation pour les étiqueter finement, et la composition de ces mentions pour aboutir à des entités structurées.

Questions de recherche

Une première ligne de questionnement découle des problèmes liés aux représentations structurées. La simple extraction d'entités et la normalisation peuvent ne pas être suffisantes pour représenter adéquatement les informations présentes dans un rapport clinique. Ainsi, **quelle structure est la mieux adaptée à l'extraction d'informations dans le domaine clinique ?** Dans le cas d'entités structurées, **comment modéliser un système pour regrouper les différentes parties d'une même entité ?** Plus généralement, **dans le cas d'entités simples comme structurées, quelles sont les difficultés rencontrées lorsque ces entités se chevauchent dans le texte, et quelles méthodes peuvent être utilisées pour surmonter ces difficultés ?**

Notre deuxième série de questions relève du langage lui-même. L'anglais étant la langue prédominante de la recherche en TAL, **peut-on construire des modèles de TAL pour d'autres langues que l'anglais comme le français ?**. Une question subsidiaire se pose : **quand peu de ressources sont disponibles dans les langues autres que l'anglais, comme dans le cas de la normalisation, est-il encore possible d'appliquer des modèles d'apprentissage à ces langues ?**

Enfin, notre dernière question concerne le besoin critique de données annotées des méthodes par apprentissage profond. Le coût de l'annotation des documents médicaux étant élevé, **quelles techniques peuvent être mises en œuvre pour entraîner des algorithmes d'apprentissage profond avec peu de données ?**

Reconnaissance d'entités nommées

Nous étudions d'abord la tâche de reconnaissance d'entités nommées (① dans la Figure B.3), et plus précisément, la tâche de reconnaissance d'entités nommées imbriquées, ou avec chevauchement. L'adaptation des modèles classiques d'étiquetage de séquence aux entités imbriquées reste un défi. À cette fin, nous proposons deux approches supervisées utilisant des réseaux de neurones. Notre première approche (figure B.4) utilise un modèle d'étiquetage auto-régressif, qui prédit itérativement des entités sans chevauchement dans une phrase. La seconde méthode (figure B.5) est basée sur un modèle de d'étiquetage combiné à un modèle d'appariement de bornes.

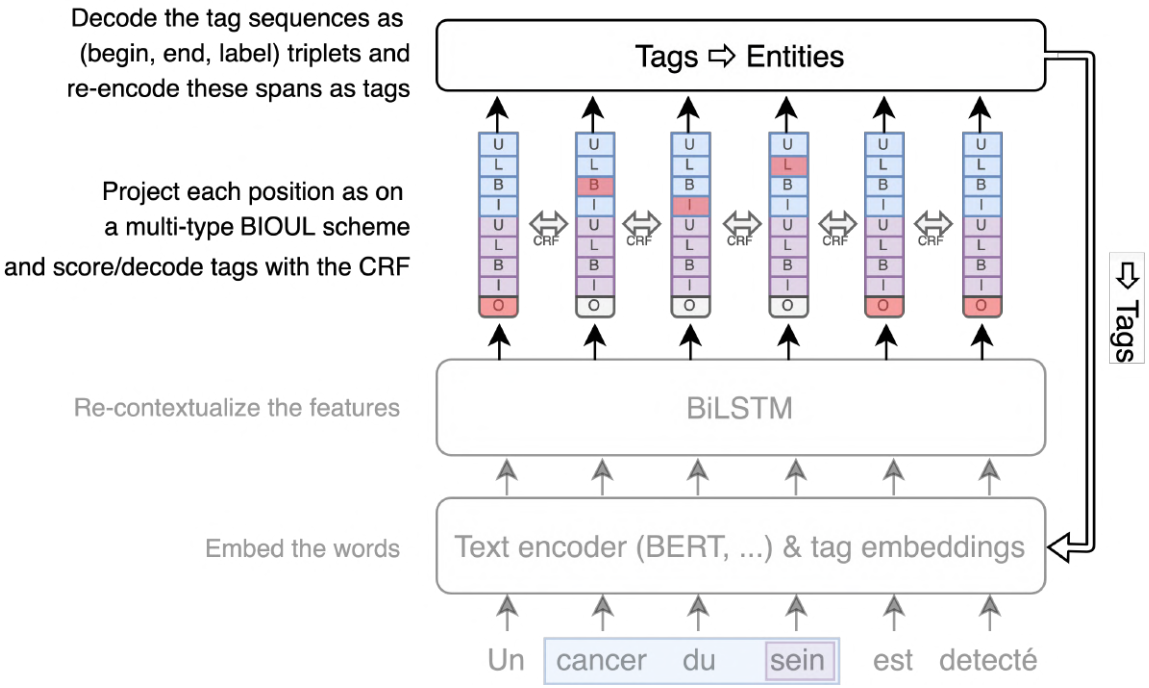


Figure B.4 Modèle de reconnaissance d'entités nommées auto-régressif

Nous étudions l'impact des caractéristiques des mots sur les performances du modèle, et observons que l'ajout du contexte de chaque phrase (phrases voisines) améliore nettement les performances, ce qui peut s'expliquer par le pré-entraînement du modèle BERT, qui est "habitué" aux phrases relativement longues. L'ajout de caractéristiques liées aux caractères de chaque mot bénéficie également à nos modèles dans une moindre mesure, tandis que l'ajout

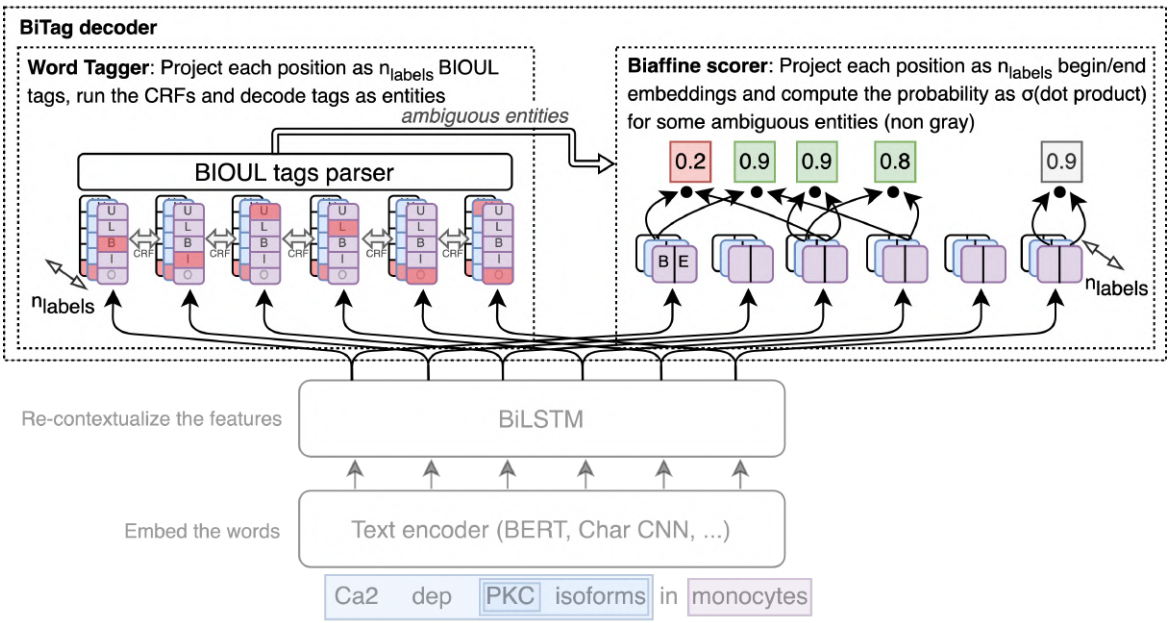


Figure B.5 Modèle de reconnaissance d'entités nommées par étiquetage de séquence appariement biaffine

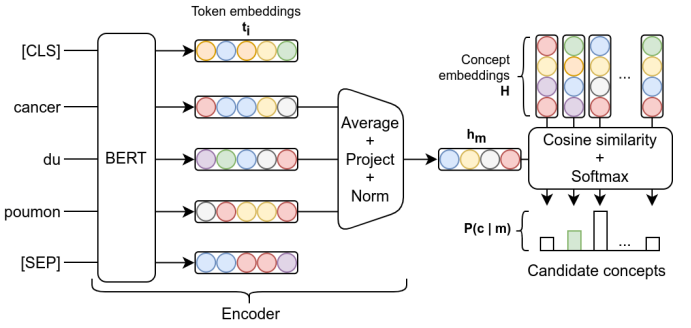
de plongements non-contextualisés FastText a un impact faible et mitigé. Enfin, prendre la moyenne des plongements de sous-mots pour chaque mot semble la méthode d'agrégation la plus efficace pour obtenir des plongements par mots en utilisant BERT. Nos deux décodeurs, malgré leurs architectures différentes, obtiennent des résultats comparables sur chacun des jeux de données. Concernant notre première approche de décodeur auto-régressif, pour laquelle il faut choisir si l'on apprend au modèle à prédire d'abord les entités les plus courtes ou les plus longues, nous observons que l'ordre optimal diffère selon le jeu de données. Cela peut être dû à des différences de domaine, de langage ou de distribution des entités. Un résultat majeur de nos expériences est la divergence entre les métriques d'évaluation exactes et approximatives. En effet, concernant notre seconde approche d'extraction par étiquetage de séquence et appariement de bornes, nous observons que l'appariement de bornes seul obtient des résultats similaires selon la métrique d'évaluation exacte des entités, mais des résultats bien moindres selon une métrique relaxée, qui ne nécessite pas d'obtenir exactement les mêmes bornes de début et de fin. Il s'agit d'un résultat important, car cette métrique n'est pas souvent renseignée dans les publications de recherche, et reflète pourtant un objectif utile qui est de trouver une entité, même si ses délimitations dans le texte sont imparfaites, plutôt que de ne rien trouver. Cet objectif est notamment pertinent dans des systèmes d'extraction en cascade, dans lesquels les sorties d'un système d'extraction d'entités sont utilisées par un autre système en aval. De plus, d'après nos expériences, cette différence entre les deux types de métriques semble être d'autant plus importante que le jeu de données est petit et contient des entités avec des limites de début/fin ambiguës. Enfin, comme attendu, notre approche de

combinaison par ensemble de modèles s'avère efficace pour améliorer les performances d'un système d'extraction d'entités nommées.

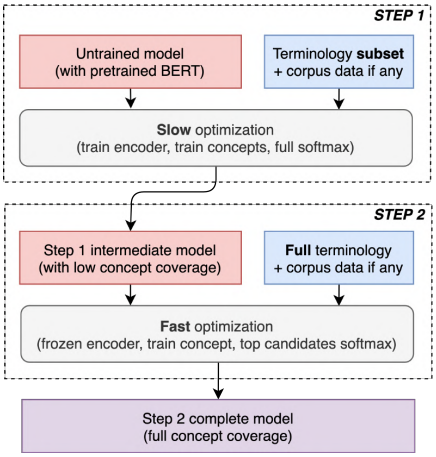
Normalisation

Nous nous concentrons ensuite sur la normalisation des entités nommées médicales (② dans la Figure B.3). Plus précisément, nous abordons la tâche de normalisation des entités médicales dans des langues autres que l'anglais pour de grandes terminologies médicales contenant des centaines de milliers de concepts, et avec peu ou pas d'échantillons annotés. Notre objectif est d'appareiller des entités nommées avec un concept dans une terminologie. Nous profitons de la nature multilingue des terminologies pour améliorer la normalisation des concepts dans les langues non anglaises sans traduction ni supervision directe.

Nous proposons à cet effet une architecture de classification standard dans laquelle nous calculons des représentations pour les entités et les concepts dans un espace multidimensionnel commun. La probabilité pour chaque entité d'appartenir à un concept est ensuite déterminée à partir de la similarité des deux représentations. Le nombre de concepts pouvant être très grand, nous proposons une technique pour accélérer l'entraînement et réduire le besoin en mémoire en découpant l'entraînement en deux étapes. Le modèle chargé de représenter les entités est d'abord pré-entraîné sur un sous-ensemble de la terminologie, puis les représentations de tous les concepts sont apprises en ne considérant que les candidats les plus probables pour chaque synonymes.



(a) Modèle de normalisation par classification



(b) Entrainement en deux étapes sur un grand nombre de concepts

Figure B.6 Modèle et méthode proposés pour la normalisation multilingue à grand échelle

Notre méthode obtient de très bons résultats sur les jeux de données Quaero FrenchMed (textes en français) et Mantra (textes anglais, français, espagnol, allemand et néerlandais). Nos expériences montrent tout d'abord que notre technique d'apprentissage en deux étapes ne

semble pas avoir d'effet négatif sur la performance du modèle, comparé à un modèle appris en une seule étape, mais réduit le temps d'apprentissage de 15h à 7h pour une terminologie de presque 1 million de concepts. Contrairement à ce que nous attendions, en comparant différents modèles BERT français, anglais, et multilingue sur Quaero, le modèle BERT pré-entraîné ne semble pas avoir d'effet majeur sur la performance. En revanche, l'entraînement sur différentes langues, pour un même nombre de concepts, a un effet notable sur les prédictions. Nous observons cela au travers d'une évaluation sur Quaero et Mantra en combinant les langues disponible dans leur terminologies respectives. Sur le jeu de données Mantra, nous observons que l'entraînement sur les cinq langues obtient de meilleurs résultats pour les langues hors anglais, en comparaison à des entrainement bilingues séparés *anglais* + *langue en question*. Enfin, nous retrouvons les similarités linguistiques entre le français et l'espagnol d'une part, et l'allemand et le néerlandais d'autre part, en observant quelle langue, autre que l'anglais et la langue d'évaluation, contribue le plus par sa terminologie à la performance de normalisation dans chaque langue.

Entités structurées

Enfin, nous nous concentrons sur le problème de l'extraction d'entités structurées à partir de rapports de radiologie du sein (③ dans la Figure B.3). Ces rapports contiennent des informations riches et utiles sur l'état physique d'un patient, son historique clinique, ainsi que les évaluations et les recommandations du médecin.

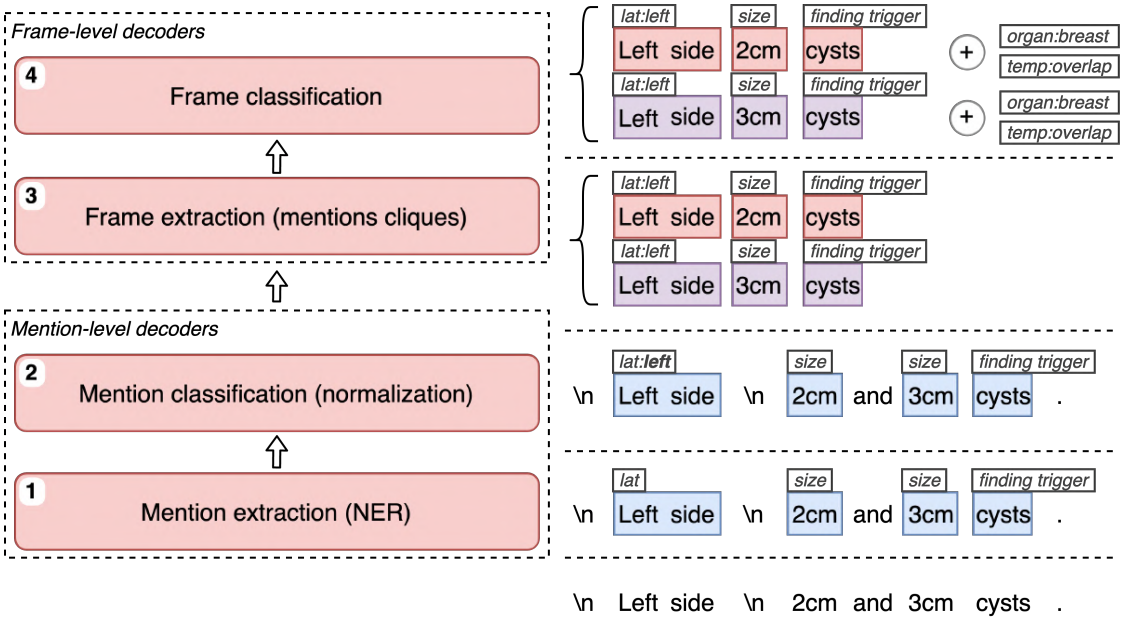
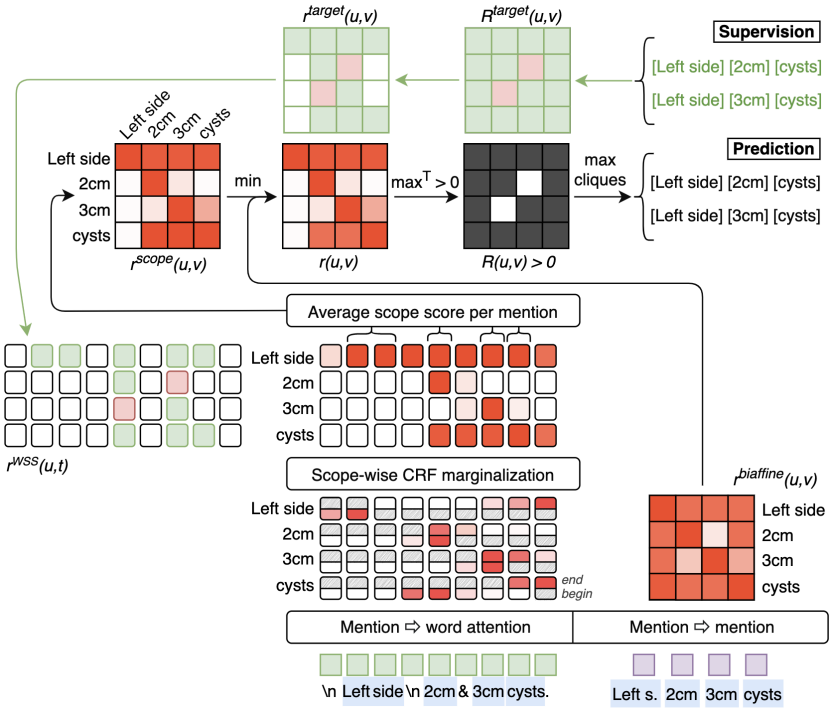
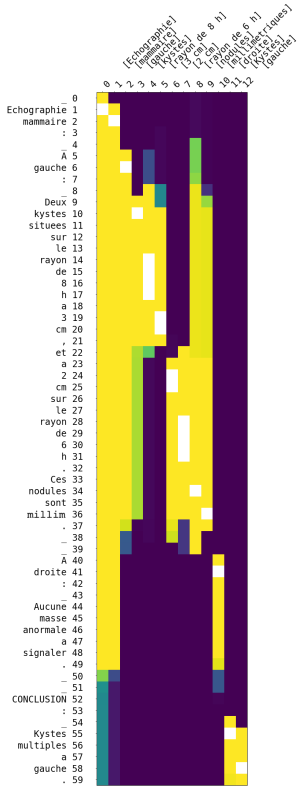


Figure B.7 Différentes étapes de structuration des comptes rendus

Nous décrivons un schéma d'annotation pour l'extraction d'entités radiologiques, de procédures et de scores à partir de ces rapports. En utilisant ce schéma, nous construisons un nouveau corpus de 120 documents manuellement annotés issus de l'entrepôt de données cliniques de l'APHP. Nous étudions également la génération automatique de ces annotations. Bien que de nombreuses méthodes existent pour des sujets connexes tels que l'extraction d'événements, le slot-filling ou la reconnaissance d'entités nommées discontinues, un défi dans notre étude réside dans le fait que les rapports cliniques contiennent généralement des cadres qui se recouvrent et s'étendent sur plusieurs phrases ou paragraphes. Nous proposons une nouvelle méthode qui résout ces difficultés et l'évaluons sur le nouveau corpus annoté.



(a) Modèle d'extraction de cadres par cliques de mentions et relations de portées



(b) Exemple de portées prédites par le modèle

Figure B.8 Modèle proposé et prédiction de cadres dans un compte rendu clinique

Le système proposé se compose de quatre modules illustrées dans la Figure B.7, entraînés conjointement. Un premier module ① extrait les entités nommées, à savoir les mentions d'objets ou de leurs caractéristiques. Le second module ② effectue la normalisation de ces entités vers une terminologie. À la différence du modèle présenté dans la section précédente, cette terminologie est très petite (moins de 50 concepts), les entités peuvent avoir plusieurs concepts, et nous prenons en compte le contexte de chaque entité. Un troisième module ③ extrait des "cadres": des cliques de mentions. Ces cliques sont générées à partir de

relations binaires entre les mentions, visant à savoir si les deux mentions appartiennent à la même entité. Pour chaque relation, nous calculons d'une part un score par produit scalaire entre les représentations de chaque mention, et introduisons le mécanisme de relation par portée. Ces relations par portées visent à déterminer si une mention est située dans la zone d'effet d'une autre mention, sans supervision spécifique de ces zones. Ce module est illustré plus en détail dans la Figure B.8a. Enfin, le dernier module ④ rempli dans chaque cadre les champs qu'aucune mention n'a pu justifier explicitement. Nous proposons également plusieurs techniques pour injecter des connaissances auxiliaires par le biais de contraintes, d'augmentation du jeu de données et d'une petite terminologie.

En évaluant notre système sur le nouveau jeu de données annotées, nous montrons que l'ajout d'informations auxiliaires peut améliorer les performances du modèle dans le contexte d'une petite quantité de données. Cette information est précieuse pour l'annotation et le développement de nouveaux systèmes de recherche d'information dans d'autres domaines, où les mots ou phrases clés sont connus à l'avance. Dans ce contexte, notre système commence à obtenir des résultats avec presque aucun document annoté. Notre méthode de détection de relations par portées améliore significativement les prédictions, et il en va de même pour plusieurs astuces de modélisation que nous implémentons, à savoir l'attention relative et une modification du mécanisme de connexion résiduelle standard. Nous montrons également que les portées peuvent être apprises sans aucune heuristique, ou annotation spécifique, et qu'elles fournissent un moyen interprétable de visualiser les prédictions du modèle, comme l'illustre la figure B.8b.

Ces différentes contributions, concernant l'extraction et la normalisation d'entités simples et structurées dans les rapports médicaux, montrent que le traitement automatique du langage clinique français est un sujet complexe qui mérite des approches spécifiques, tant du point de vue de la modélisation du système que du point de vue de la collecte des données et de leur injection dans les modèles.

Bibliography

Afzal, Z., Akhondi, S. A., Van Haagen, H. H., Van Mulligen, E. M., and Kors, J. A. (2015). Biomedical concept recognition in French text using automatic translation of English terms. In *CEUR Workshop Proceedings*, volume 1391.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alawad, M., Yoon, H. J., and Tourassi, G. D. (2018). Coarse-to-fine multi-task training of convolutional neural networks for automated information extraction from cancer pathology reports. *2018 IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2018*, 2018-Janua:218–221.

Alex, B., Haddow, B., and Grover, C. (2007). Recognising nested named entities in biomedical text. In *ACL 2007 - Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72.

Alfonseca, E. and Manandhar, S. (2002). An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. *Proceedings of the 1st International Conference on General WordNet Mysore India*, 69(6):1–9.

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., and McDermott, M. (2019). Publicly Available Clinical. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arbabi, A., Adams, D. R., Fidler, S., and Brudno, M. (2019). Identifying clinical terms in medical text using ontology-guided machine learning. *Journal of Medical Internet Research*, 21(5).

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings. AMIA Symposium*, pages 17–21.

Asahara, M. and Matsumoto, Y. (2003). Japanese Named Entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, volume 1, pages 8–15, Morristown, NJ, USA. Association for Computational Linguistics.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Aubin, S. and Hamon, T. (2006). Improving Term Extraction with Terminological Resources. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4139 LNAI, pages 380–387.

Bahdanau, D., Bosc, T., Jastrzębski, S., Grefenstette, E., Vincent, P., and Bengio, Y. (2017). Learning to Compute Word Embeddings On the Fly.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, volume 1, page 86, Morristown, NJ, USA. Association for Computational Linguistics.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):675–721.

Beltagy, I., Lo, K., and Cohan, A. (2020a). SCIBERT: A pretrained language model for scientific text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3615–3620.

Beltagy, I., Peters, M. E., and Cohan, A. (2020b). Longformer: The Long-Document Transformer.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble. In *Proceedings of the fifth conference on Applied natural language processing* -, pages 194–201, Morristown, NJ, USA. Association for Computational Linguistics.

Bitterman, D. S., Miller, T. A., Mak, R. H., and Savova, G. K. (2021). Clinical Natural Language Processing for Radiation Oncology: A Review and Practical Primer. *International Journal of Radiation Oncology Biology Physics*, 110(3):641–655.

Björne, J. and Salakoski, T. (2011). Generalizing Biomedical Event Extraction. *Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task Portland Oregon June Association for Computational Linguistics*, page 183–191.

Björne, J. and Salakoski, T. (2013). TEES 2.1: Automated Annotation Scheme Learning in the BioNLP 2013 Shared Task. *BioNLP Shared Task 2013 Workshop*, pages 16–25.

Björne, J. and Salakoski, T. (2015). TEES 2.2: Biomedical Event Extraction for Diverse Corpora. *BMC Bioinformatics*, 16(16):1–20.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(DATABASE ISS.):D267.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. *in Proceedings of the Sixth Workshop on Very Large Corpora*, pages 152–160.

Bramer, G. R. (1988). International statistical classification of diseases and related health problems - Tenth revision.

Brin, S. (1999). Extracting patterns and relations from the world wide web. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1590:172–183.

Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020-Decem.

Burek, P., Scherf, N., and Herre, H. (2019). Ontology patterns for the representation of quality changes of cells in time. *Journal of Biomedical Semantics*, 10(1).

Bustos, A., Pertusa, A., Salinas, J. M., and de la Iglesia-Vayá, M. (2020). PadChest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797.

Cabot, C., Lelong, R., Grosjean, J., Soualmia, L. F., and Darmoni, S. J. (2016). Retrieving Clinical and Omic Data from Electronic Health Records. *Studies in health technology and informatics*, 221:115.

Card, S. K., Newell, A., and Moran, T. P. (1983). *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., USA.

Cardon, R., Grabar, N., Grouin, C., and Hamon, T. (2020). Presentation of the DEFT 2020 Challenge : open domain textual similarity and precise information extraction from clinical cases. In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCIT*, pages 1–13.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:213–229.

Castano, J., Gambarte, M. L., Park, H. J., Avila Williams, M. d. P., Perez, D., Campos, F., Luna, D., Benitez, S., Berinsky, H., and Zanetti, S. (2016). A Machine Learning Approach to Clinical Terms Normalization. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 1–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Castro, S. M., Tseytlin, E., Medvedeva, O., Mitchell, K., Visweswaran, S., Bekhuis, T., and Jacobson, R. S. (2017). Automated annotation and classification of BI-RADS assessment from radiology reports. *Journal of Biomedical Informatics*, 69:177–187.

Chiaramello, E., Pinciroli, F., Bonalumi, A., Caroli, A., and Tognola, G. (2016). Use of "off-the-shelf" information extraction algorithms in clinical informatics: A feasibility study of MetaMap annotation of Italian medical notes. *Journal of Biomedical Informatics*, 63:22–32.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*.

Collins, M. and Singer, Y. (1999). Unsupervised Models for Named Entity Classification. *Proceedings of EMNLP/VLC-99*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 160–167, New York, New York, USA. ACM Press.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Copara, J., Knafou, J., Naderi, N., Moro, C., Ruch, P., and Teodoro, D. (2020). Contextualized {F}rench Language Models for Biomedical Named Entity Recognition. *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCIT*, pages 36–48.

Dai, H. J., Lai, P. T., Chang, Y. C., and Tsai, R. T. H. (2015). Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of Cheminformatics*, 7(Suppl 1):S14.

Dai, X., Karimi, S., Hachey, B., and Paris, C. (2020). An Effective Transition-based Model for Discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5860–5870, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dalloux, C., Claveau, V., Grabar, N., Oliveira, L. E. S., Moro, C. M. C., Gumiel, Y. B., and Carvalho, D. R. (2020). Supervised learning for the detection of negation and of its scope in French and Brazilian Portuguese biomedical corpora. *Natural Language Engineering*, 27(2):181–201.

Datta, S., Bernstam, E. V., and Roberts, K. (2019). A frame semantic overview of NLP-based information extraction for cancer-related EHR notes.

Datta, S., Si, Y., Rodriguez, L., Shooshan, S. E., Demner-Fushman, D., and Roberts, K. (2020a). Understanding spatial language in radiology: Representation framework, annotation, and spatial relation extraction from chest X-ray reports using deep learning. *Journal of Biomedical Informatics*, 108(February):103473.

Datta, S., Ulinski, M., Godfrey-Stovall, J., Khanpara, S., Riascos-Castaneda, R. F., and Roberts, K. (2020b). Rad-SpatialNet: A Frame-based Resource for Fine-Grained Spatial Relations in Radiology Reports. *LREC ... International Conference on Language Resources & Evaluation : [proceedings]. International Conference on Language Resources and Evaluation*, 2020:2251.

De Cao, N., Izacard, G., Riedel, S., and Petroni, F. (2020). Autoregressive Entity Retrieval. *9th International Conference on Learning Representations*.

Deléger, L., Merabti, T., Lecrocq, T., Joubert, M., Zweigenbaum, P., and Darmoni, S. (2010). A twofold strategy for translating a medical terminology into French. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, 2010:152–156.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., and Adam, H. (2014). Large-scale object classification using label relation graphs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8689 LNCS, pages 48–64.

Deng, P., Chen, H., Huang, M., Ruan, X., and Xu, L. (2019). An ensemble CNN method for biomedical entity normalization. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 143–149, Stroudsburg, PA, USA. Association for Computational Linguistics.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186.

Dey, R. and Salemt, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. *Midwest Symposium on Circuits and Systems*, 2017-Augus:1597–1600.

Dogan, R. I. and Lu, Z. (2012). An inference method for disease name normalization. *AAAI Fall Symposium - Technical Report*, FS-12-05:8–13.

Domingos, P. and Lowd, D. (2009). Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 7(1):1–153.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. W. (2019). Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Doğan, R. I., Leaman, R., and Lu, Z. (2014). NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

D'Souza, J. and Ng, V. (2015). Sieve-based entity linking for the biomedical domain. In *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, volume 2, pages 297–302.

Ehrmann, M. (2008). Les Entitées Nommées, de la linguistique au TAL : Statut théorique et méthodes de désambiguïsation.

El Boukkouri, H., Ferret, O., Lavergne, T., Noji, H., Zweigenbaum, P., and Tsujii, J. (2020). CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Stroudsburg, PA, USA. International Committee on Computational Linguistics.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M. M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

Fakhraei, S., Mathew, J., and Ambite, J. L. (2020). NSEEN: Neural Semantic Embedding for Entity Normalization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11907 LNAI:665–680.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.

Ferré, A., Deléger, L., Zweigenbaum, P., and Nédellec, C. (2019). Combining rule-based and embedding-based approaches to normalize textual entities with an ontology. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 3443–3447.

Ferré, A., Zweigenbaum, P., and Nédellec, C. (2017). Representation of complex terms in a vector space structured by an ontology for a normalization task. In *BioNLP 2017*, pages 99–106, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fillmore, C. J. (1982). Frame semantics.

Finkel, J. R. and Manning, C. D. (2009). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing Volume 1 - EMNLP '09*, volume 1, page 141, Morristown, NJ, USA. Association for Computational Linguistics.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.

Fisher, J. and Vlachos, A. (2019). Merge and Label: A Novel Neural Network Architecture for Nested NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5840–5850, Stroudsburg, PA, USA. Association for Computational Linguistics.

Freitas, F., Schulz, S., and Moraes, E. (2009). Survey of current terminologies and ontologies in biology and medicine. *Reciis*, 3(1):7–18.

Gangadharaiah, R. and Narayanaswamy, B. (2019). Joint multiple intent detection and slot labeling for goal-oriented dialog. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 564–569, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gérardin, C., Vaillant, P., Wajsbürt, P., Gilavert, C., Bellamine, A., Kempf, E., and Tannier, X. (2021). Classification multilabel de concepts médicaux pour l'identification du profil clinique du patient (Multilabel classification of medical concepts for patient's clinical profile identification).

Ghiasvand, O. and Kate, R. (2014). UWM: Disorder Mention Extraction from Clinical Text Using CRFs and Normalization Using Learned Edit Distance Patterns. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 828–832, Stroudsburg, PA, USA. Association for Computational Linguistics.

Golik, W., Bossy, R., Ratkovic, Z., and Nédellec, C. (2013). Improving term extraction with linguistic analysis in the biomedical domain. *Research in Computing Science*, 70(1):157–172.

Grosjean, J., Merabti, T., Dahamna, B., Kergourlay, I., Thirion, B., Soualmia, L. F., and Darmoni, S. J. (2011). Health multi-terminology portal: a semantic added-value for patient safety. *Studies in health technology and informatics*, 166:129–38.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Gu, B. (2006). Recognizing nested named entities in GENIA corpus. *HLT-NAACL 2006 - BioNLP 2006: Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis, Proceedings of the Workshop*, pages 112–113.

Guo, Y., Liu, Y., Georgiou, T., and Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93.

Hanisch, D., Fundel, K., Mevissen, H. T., Zimmer, R., and Fluck, J. (2005). ProMiner: Rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(SUPPL.1):1–9.

Harris, Z. S. (1954). Distributional Structure. *Distributional Structure, WORD*, 10(3):146–162.

Hassanpour, S. and Langlotz, C. P. (2016). Information extraction from multi-institutional radiology reports. *Artificial Intelligence in Medicine*, 66(1):29–39.

He, P., Liu, X., Gao, J., and Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention.

He, T., Puppala, M., Ogunti, R., Mancuso, J. J., Yu, X., Chen, S., Chang, J. C., Patel, T. A., and Wong, S. T. (2017). Deep learning analytics for diagnostic support of breast cancer disease management. *2017 IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2017*, pages 365–368.

Heimonen, J., Björne, J., and Salakoski, T. (2010). Reconstruction of semantic relationships from their projections in biomolecular domain.

Hermans, A., Beyer, L., and Leibe, B. (2017). In Defense of the Triplet Loss for Person Re-Identification.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hoffer, E. and Ailon, N. (2015). Deep Metric Learning Using Triplet Network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9370, pages 84–92.

Huang, M., Névéol, A., and Lu, Z. (2011). Recommending MeSH terms for annotating biomedical articles. *Journal of the American Medical Informatics Association*, 18(5):660–667.

Huang, Z., Xu, W., Kai, Y., and Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, abs/1508.0.

Jain, S., Agrawal, A., Saporta, A., Truong, S. Q., Duong, D. N., Bui, T., Chambon, P., Zhang, Y., Lungren, M. P., Ng, A. Y., Langlotz, C. P., and Rajpurkar, P. (2021). RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. (NeurIPS).

Jean, S., Cho, K., Memisevic, R., and Bengio, Y. (2015). On using very large target vocabulary for neural machine translation. In *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, volume 1, pages 1–10.

Ji, Z., Wei, Q., and Xu, H. (2020). BERT-based Ranking for Biomedical Entity Normalization. *AMIA Summits on Translational Science Proceedings*, 2020:269.

Jiang, J., Guan, Y., and Zhao, C. (2015). WI-ENRE in CLEF eHealth Evaluation Lab 2015: Clinical named entity recognition based on CRF. In *CEUR Workshop Proceedings*, volume 1391.

Jonnagaddala, J., Jue, T. R., Chang, N.-W., and Dai, H.-J. (2016). Improving the dictionary lookup approach for disease normalization using enhanced dictionary and query expansion. *Database*, 2016(2016):baw112.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2019). SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Ju, M., Miwa, M., and Ananiadou, S. (2018). A Neural Layered Model for Nested Named Entity Recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1446–1459, Stroudsburg, PA, USA. Association for Computational Linguistics.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Heafield, H. H. K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F., and Birch, A. (2015). Marian: Fast neural machine translation in c++. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, pages 116–121.

Kantor, B. and Globerson, A. (2020). Coreference resolution with entity equalization. Technical report.

Katiyar, A. and Cardie, C. (2018). Nested Named Entity Recognition Revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 861–871, Stroudsburg, PA, USA. Association for Computational Linguistics.

Khandelwal, A. and Sawant, S. (2020). NegBERT: A transfer learning approach for negation detection and scope resolution. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, pages 5739–5748.

Kim, J. D., Ohta, T., Tateisi, Y., and Tsujii, J. (2003). GENIA corpus - A semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(SUPPL. 1):i180–i182.

Kim, Y. (2014). Convolutional neural networks for sentence classification. Technical report.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Klein, D., Smarr, J., Nguyen, H., and Manning, C. D. (2003). Named entity recognition with character-level models. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*, volume 4, pages 180–183, Morristown, NJ, USA. Association for Computational Linguistics.

Kong, L., D'Autume, C. d. M., Ling, W., Yu, L., Dai, Z., and Yogatama, D. (2020). A Mutual Information Maximization Perspective of Language Representation Learning. In *ICLR*.

Kors, J. A., Clematide, S., Akhondi, S. A., Van Mulligen, E. M., and Rebholz-Schuhmann, D. (2015). A multilingual gold-standard corpus for biomedical concept recognition: The Mantra GSC. *Journal of the American Medical Informatics Association*, 22(5):948–956.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:66–75.

Kuo, C. J., Ling, M. H., Lin, K. T., and Hsu, C. N. (2009). BIOADI: A machine learning approach to identifying abbreviations and definitions in biological literature. *BMC Bioinformatics*, 10(SUPPL. 15):1–10.

Lacson, R., Harris, K., Brawarsky, P., Tosteson, T. D., Onega, T., Tosteson, A. N. A., Kaye, A., Gonzalez, I., Birdwell, R., and Haas, J. S. (2015). Evaluation of an Automated Information Extraction Tool for Imaging Data Elements to Populate a Breast Cancer Screening Registry. *Journal of Digital Imaging*, 28(5):567.

Lafferty, J., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, 8(June):282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, Stroudsburg, PA, USA. Association for Computational Linguistics.

Leaman, R., Doğan, R. I., and Lu, Z. (2013). DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Leaman, R. and Lu, Z. (2016). TaggerOne: Joint named entity recognition and normalization with semi-Markov Models. *Bioinformatics*, 32(18):2839–2846.

Lee, H.-C., Hsu, Y.-Y., and Kao, H.-Y. (2015). An enhanced CRF-based system for disease name entity recognition and normalization on BioCreative V DNER Task. *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pages 226–233.

Lee, J., Yoon, W., Kim, S. S., Kim, D., Kim, S. S., So, C. H., and Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, B. (2021). Named Entity Recognition in the Style of Object Detection.

Li, F., Lin, Z., Zhang, M., and Ji, D. (2021). A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4814–4828, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, H., Chen, Q., Tang, B., Wang, X., Xu, H., Wang, B., and Huang, D. (2017). CNN-based ranking for biomedical entity normalization. *BMC Bioinformatics*, 18(S11):385.

Li, H. and Lu, W. (2018). Learning with structured representations for negation scope extraction. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2:533–539.

Li, J., Sun, A., Han, J., and Li, C. (2020). A Survey on Deep Learning for Named Entity Recognition. Technical report.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C. H., Leaman, R., Davis, A. P., Mattingly, C. J., Wiegers, T. C., and Lu, Z. (2016). BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database : the journal of biological databases and curation*, 2016:baw068.

Liberman, L. and Menell, J. H. (2002). Breast imaging reporting and data system (BI-RADS). *Radiologic Clinics of North America*, 40(3):409–430.

Limsopatham, N. and Collier, N. (2016). Normalising medical concepts in social media texts by learning semantic representation. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2:1014–1023.

Lin, C., Miller, T., Dligach, D., Bethard, S., and Savova, G. (2021). EntityBERT: Entity-centric Masking Strategy for Model Pretraining for the Clinical Domain. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 191–201, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, volume 2, pages 768–774, Morristown, NJ, USA. Association for Computational Linguistics.

Lin, H., Lu, Y., Han, X., and Sun, L. (2019). Sequence-to-Nuggets: Nested Entity Mention Detection via Anchor-Region Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lipscomb, C. E. (2000). Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–6.

Liu, H. and Xu, Y. (2018). A deep learning way for disease name representation and normalization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10619 LNAI, pages 151–157.

Liu, X., Bordes, A., and Grandvalet, Y. (2015). Extracting biomedical events from pairs of text entities. *BMC Bioinformatics*, 16(S10):S8.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Lu, W. and Roth, D. (2015). Joint Mention Extraction and Classification with Mention Hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., and Hajishirzi, H. (2019). A general framework for information extraction using dynamic span graphs. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:3036–3046.

Luo, Y., Song, G., Li, P., and Qi, Z. (2018). Multi-task medical concept normalization using multi-view convolutional neural network. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 5868–5875.

Luoma, J. and Pyysalo, S. (2021). Exploring Cross-sentence Contexts for Named Entity Recognition with BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 904–914, Stroudsburg, PA, USA. International Committee on Computational Linguistics.

Marko, K., Baud, R., Zweigenbaum, P., Borin, L., Merkel, M., and Schulz, S. (2006). Towards a multilingual medical lexicon. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 534–538.

Martin, L., Muller, B., Ortiz Suarez, P. J., Dupont, Y., Romary, L., de la Clergerie, E., Seddah, D., and Sagot, B. (2020). CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Stroudsburg, PA, USA. Association for Computational Linguistics.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*, volume 4, pages 188–191, Morristown, NJ, USA. Association for Computational Linguistics.

McCray, A. T., Burgun, A., and Bodenreider, O. (2001). Aggregating UMLS semantic types for reducing conceptual complexity. In *Studies in Health Technology and Informatics*, volume 84, pages 216–220.

Mengge, X., Yu, B., Zhang, Z., Liu, T., Zhang, Y., and Wang, B. (2020). Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Metke, A. and Karimi, S. (2016). Concept Identification and Normalisation for Adverse Drug Event Discovery in Medical Forums. *Bmdid*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Miwa, M., Sætre, R., Kim, J. D., and Tsujii, J. (2010). Event extraction with complex event classification using rich features. *Journal of Bioinformatics and Computational Biology*, 8(1):131–146.

Miwa, M., Thompson, P., Korkontzelos, I., and Ananiadou, S. (2014). Comparable study of event extraction in newswire and biomedical domains. *COLING 2014 - 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*, pages 2270–2279.

Mondal, I., Purkayastha, S., Sarkar, S., Goyal, P., Pillai, J., Bhattacharyya, A., and Gattu, M. (2019). Medical Entity Linking using Triplet Network. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Moore, C., Farrag, A., and Ashkin, E. (2017). Using Natural Language Processing to Extract Abnormal Results from Cancer Screening Reports. *Journal of patient safety*, 13(3):138.

Muis, A. O. and Lu, W. (2017). Labeling Gaps Between Words: Recognizing Overlapping Mentions with Mention Separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Stroudsburg, PA, USA. Association for Computational Linguistics.

Musen, M. A. and Team, t. P. (2015). The Protégé Project: A Look Back and a Look Forward. *AI matters*, 1(4):4.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Névéol, A., Dalianis, H., Velupillai, S., Savova, G., and Zweigenbaum, P. (2018). Clinical Natural Language Processing in languages other than English: opportunities and challenges. *Journal of Biomedical Semantics*, 9(1):12.

Névéol, A., Grouin, C., Leixa, J., Rosset, S., and Zweigenbaum, P. (2014). The Quaero French medical corpus: A ressource for medical entity recognition and normalization. Technical report.

Neves, M. and Ševa, J. (2021). An extensive review of tools for manual annotation of documents. *Briefings in Bioinformatics*, 22(1):146–163.

Nguyen, T. M. and Nguyen, T. H. (2019). One for all: Neural joint modeling of entities and events. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 6851–6858.

Organization, W. H. (1978). International classification of diseases : [9th] ninth revision, basic tabulation list with alphabetic index.

Paris, N., Doutreligne, M., Parrot, A., Tannier, X., Paris, N., Doutreligne, M., Parrot, A., and Tannier, X. (2019). Désidentification de comptes-rendus hospitaliers dans une base de données OMOP. In *TALMED 2019 : Symposium satellite francophone sur le traitement automatique des langues dans le domaine biomédical*.

Paşca, M., Lin, D., Bigham, J., Lifchits, A., and Jain, A. (2006). Organizing and searching the World Wide Web of facts - Step one: The que-million fact extraction challenge. *Proceedings of the National Conference on Artificial Intelligence*, 2:1400–1405.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perez, N., Accuosto, P., Bravo, A., Cuadros, M., Martínez-Garcia, E., Saggion, H., and Rigau, G. (2020). Cross-lingual semantic annotation of biomedical literature: Experiments in Spanish and English. *Bioinformatics*, 36(6):1872–1880.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:2227–2237.

Phan, M. C., Sun, A., and Tay, Y. (2020). Robust representation learning of biomedical names. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 3275–3285.

Pires, T., Schlinger, E., and Garrette, D. (2020). How multilingual is multilingual BERT? Technical report.

Pradhan, S., Elhadad, N., Chapman, W., Manandhar, S., and Savova, G. (2014). SemEval-2014 Task 7: Analysis of Clinical Text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 54–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pyysalo, S., Ohta, T., and Ananiadou, S. (2013). Overview of the Cancer Genetics (CG) task of BioNLP Shared Task 2013. *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 58–66.

Qin, L., Xu, X., Che, W., and Liu, T. (2020). AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qiu, J. X., Yoon, H. J., Fearn, P. A., and Tourassi, G. D. (2018). Deep Learning for Automated Extraction of Primary Sites from Cancer Pathology Reports. *IEEE Journal of Biomedical and Health Informatics*, 22(1):244–251.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Radford, A., Narasimhan, T., Salimans, T., and Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. *OpenAI Blog*, pages 1–12.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2020). Language Models are Unsupervised Multitask Learners. Technical Report May.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:1–67.

Raghavan, P., Chen, J. L., Fosler-Lussier, E., and Lai, A. M. (2014). How essential are unstructured clinical narratives and information fusion to clinical trial recruitment? *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2014:218–23.

Ramshaw, L. A. and Marcus, M. P. (1999). Text Chunking Using Transformation-Based Learning. pages 157–176.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL '09*, page 147, Morristown, NJ, USA. Association for Computational Linguistics.

Rau, L. (1990). Extracting company names from text. In *Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume i, pages 29–32. IEEE Comput. Soc. Press.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Riloff, E. and Jones, R. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. *Proceedings of the National Conference on Artificial Intelligence*, (1032):474–479.

Roberts, K., Si, Y., Gandhi, A., and Bernstam, E. V. (2019). A framenet for cancer information in clinical narratives: Schema and annotation. In *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, pages 272–279.

Roller, R., Kittner, M., Weissenborn, D., and Leser, U. (2018). Cross-lingual candidate search for biomedical concept normalization. Technical report.

Ruder, S., Søgaard, A., and Vulić, I. (2019). Unsupervised Cross-Lingual Representation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–53.

Sang, E. F. T. K. and De Meulder, F. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 20:142–147.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., and Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Savova, G. K., Tseytlin, E., Finan, S., Castine, M., Miller, T., Medvedeva, O., Harris, D., Hochheiser, H., Lin, C., Chavan, G., and Jacobson, R. S. (2017). DeepPhe: A natural language processing system for extracting cancer phenotypes from clinical records. *Cancer Research*, 77(21):e115–e118.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 815–823.

Schuemie, M. J., Jelier, R., and Kors, J. A. (2007). Peregrine: Lightweight gene name normalization by dictionary lookup. In *Proc of the Second BioCreative Challenge Evaluation Workshop*, pages 131–133.

Schweter, S. and Akbik, A. (2020). FLERT: Document-Level Features for Named Entity Recognition.

Sekine, S. (1998). NYU: Description of the Japanese NE system used for MET-2. *Proceedings of the 7th Message Understanding Conference (MUC-7)*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, pages 1715–1725, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sevgili, O., Shelmanov, A., Arkhipov, M., Panchenko, A., and Biemann, C. (2020). Neural Entity Linking: A Survey of Models based on Deep Learning. *arXiv*.

Shen, D., Zhang, J., Zhou, G., Su, J., and Tan, C.-L. (2003). Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine -*, volume 13, pages 49–56, Morristown, NJ, USA. Association for Computational Linguistics.

Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., and Lu, W. (2021). Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shibuya, T. and Hovy, E. (2020). Nested Named Entity Recognition via Second-best Sequence Learning and Decoding. *Transactions of the Association for Computational Linguistics*, 8:605–620.

Si, Y. and Roberts, K. (2018). A Frame-Based NLP System for Cancer-Related Information Extraction.

Sider, T. (2001). *Four-Dimensionalism*. Oxford University Press.

Sohrab, M. G. and Miwa, M. (2018). Deep Exhaustive Model for Nested Named Entity Recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Stroudsburg, PA, USA. Association for Computational Linguistics.

Spackman, K. A., Campbell, K. E., and Côté, R. A. (1997). SNOMED RT: a reference terminology for health care. *Proceedings of the AMIA Annual Fall Symposium*, 4(SUPPL.):640.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Steichen, O., Rossignol, P., Daniel-Lebozec, C., Charlet, J., Jaulent, M. C., and Degoulet, P. (2007). Maintenance of a computerized medical record form. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, pages 691–695.

Steinkamp, J. M., Chambers, C., Lalevic, D., Zafar, H. M., and Cook, T. S. (2019). Toward Complete Structured Information Extraction from Radiology Reports Using Machine Learning. *Journal of Digital Imaging*, 32(4):554–564.

Stenetorp, P., Pyysalo, S., Topíc, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). BRAT: A Web-based tool for NLP-Assisted text annotation.

Straková, J., Straka, M., and Hajic, J. (2019). Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Stroudsburg, PA, USA. Association for Computational Linguistics.

Strubell, E., Verga, P., Belanger, D., and McCallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 2670–2680.

Sugimoto, K., Takeda, T., Oh, J. H., Wada, S., Konishi, S., Yamahata, A., Manabe, S., Tomiyama, N., Matsunaga, T., Nakanishi, K., Matsumura, Y., K, S., T, T., JH, O., S, W., S, K., A, Y., S, M., N, T., T, M., K, N., and Y, M. (2021). Extracting clinical terms from radiology reports with deep learning. *Journal of Biomedical Informatics*, 116:103729.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to Fine-Tune BERT for Text Classification? In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11856 LNAI, pages 194–206.

Sung, M., Jeon, H., Lee, J., and Kang, J. (2020). Biomedical Entity Representations with Synonym Marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650, Stroudsburg, PA, USA. Association for Computational Linguistics.

Taira, R. K., Soderland, S. G., and Jakobovits, R. M. (2001). Automatic structuring of radiology free-text reports. *Radiographics*, 21(1):237–245.

Tan, Z., Shen, Y., Zhang, S., Lu, W., and Zhuang, Y. (2021). A Sequence-to-Set Network for Nested Named Entity Recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3936–3942, California. International Joint Conferences on Artificial Intelligence Organization.

Tang, B., Hu, J., Wang, X., and Chen, Q. (2018). Recognizing Continuous and Discontinuous Adverse Drug Reaction Mentions from Social Media Using LSTM-CRF. *Wireless Communications and Mobile Computing*, 2018:1–8.

Tang, B., Wu, Y., Jiang, M., Denny, J. C., and Xu, H. (2013). Recognizing and encoding disorder concepts in clinical text using machine learning and vector space model. *CEUR Workshop Proceedings*, 1179.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT Rediscovers the Classical NLP Pipeline. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 4593–4601.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pages 2214–2218.

Tiedemann, J. and Thottingal, S. (2020). OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.

Tomita, E., Tanaka, A., and Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42.

Trieu, H. L., Tran, T. T., Duong, K. N., Nguyen, A., Miwa, M., and Ananiadou, S. (2020). DeepEventMine: End-to-end neural nested event extraction from biomedical texts. *Bioinformatics*, 36(19):4910–4917.

Tsuruoka, Y., McNaught, J., Tsujii, J., and Ananiadou, S. (2007). Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, 23(20):2768–2774.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. *ACL 2010 - 48th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 384–394.

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2167, pages 491–502. Springer Verlag.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Tutubalina, E., Miftahutdinov, Z., Nikolenko, S., and Malykh, V. (2018). Medical concept normalization in social media posts with recurrent neural networks. *Journal of Biomedical Informatics*, 84:93–102.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem:5999–6009.

Vincze, V., Szarvas, G., Farkas, R., Móra, G., and Csirik, J. (2008). The BioScope corpus: Biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(SUPPL. 11):1–9.

Wajsbürt, P., Sarfati, A., and Tannier, X. (2021a). Medical concept normalization in French using multilingual terminologies and contextual embeddings. *Journal of Biomedical Informatics*, 114:103684.

Wajsbürt, P., Taillé, Y., Lainé, G., and Tannier, X. (2020). Participation de l'équipe du LIMICS à DEFT 2020. In *Actes de la 6e conference conjointe Journees d'Etudes sur la Parole (JEP, 33e edition), Traitement Automatique des Langues Naturelles (TALN, 27e edition), Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECIT*.

Wajsbürt, P., Taillé, Y., and Tannier, X. (2021b). Effect of Depth Order on Iterative Nested Named Entity Recognition Models. pages 428–432.

Wallace, E., Wang, Y., Li, S., Singh, S., and Gardner, M. (2019). Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5306–5314, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, B. and Lu, W. (2018). Neural Segmental Hypergraphs for Overlapping Mention Recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, B. and Lu, W. (2019). Combining Spans into Entities: A Neural Two-Stage Approach for Recognizing Discontiguous Entities. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6215–6223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, B., Lu, W., Wang, Y., and Jin, H. (2018a). A Neural Transition-based Model for Nested Mention Recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1017, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018b). CosFace: Large Margin Cosine Loss for Deep Face Recognition. Technical report.

Wang, J., Shou, L., Chen, K., and Chen, G. (2020). Pyramid: A Layered Model for Nested Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S., and Liu, H. (2018c). Clinical information extraction applications: A literature review. *Journal of Biomedical Informatics*, 77(June 2017):34–49.

Weld, H., Huang, X., Long, S., Poon, J., and Han, S. C. (2021). A survey of joint intent detection and slot-filling models in natural language understanding.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wright, D., Katsis, Y., Mehta, R., and Hsu, C.-N. (2019). NormCo: Deep Disease Normalization for Biomedical Knowledge Base Construction. *Akbc 2019*.

Wu, S. and Dredze, M. (2020). Are All Languages Created Equal in Multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.

Xiang, W. and Wang, B. (2019). A Survey of Event Extraction from Text. *IEEE Access*, 7:173111–173137.

Xu, M., Jiang, H., and Watcharawittayakul, S. (2017). A local detection approach for named entity recognition and mention detection. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1237–1247.

Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. (2020). LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., and Qiu, X. (2021). A Unified Generative Framework for Various NER Subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on*

Natural Language Processing (Volume 1: Long Papers), pages 5808–5822, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding.

Yu, J., Bohnet, B., and Poesio, M. (2020). Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 2020-Decem.

Zhai, Y., Guo, X., Lu, Y., and Li, H. (2019). In defense of the classification loss for person re-identification. Technical report.

Zhang, J., Shen, D., Zhou, G., Su, J., and Tan, C.-L. (2004). Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411–422.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232.

Zheng, C., Cai, Y., Xu, J., Leung, H.-f., and Xu, G. (2019). A Boundary-aware Neural Model for Nested Named Entity Recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 357–366, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhou, G., Zhang, J., Su, J., Shen, D., and Tan, C. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.

Zhou, G. D. (2006). Recognizing names in biomedical texts using mutual information independence model and SVM plus sigmoid. *International Journal of Medical Informatics*, 75(6):456–467.

Zweigenbaum, P., Baud, R., Burgun, A., Namer, F., Jarrousse, E., Grabar, N., Ruch, P., Le Duff, F., François Forgete, J., Douyéref, M., and Darmoni, S. (2003). UMLF: a Unified Medical Lexicon for French. *AMIA ... Annual Symposium proceedings /AMIA Symposium. AMIA Symposium*, page 1062.

Abstract
Hospital clinical documents are rich sources of information for various applications such as patient recruitment for clinical research, epidemiological surveillance, medical coding, and decision support tools. However, being primarily written in natural language, these documents are not easily amenable to large-scale computer processing and must first be structured. We aim to extract entities mentioned in these documents, whether simple or structured, i.e., containing several labels or parts, and normalize them with concept bases. We contribute to several natural language processing (NLP) tasks, namely named entity recognition (NER), medical entity normalization, and structured entity extraction. In particular, we investigate training deep learning models in low data settings, for languages other than English and in the clinical domain. We structure our approach in three steps: tag, normalize, and compose. We first propose two methods to tag simple entities, especially when they overlap in texts. We then develop a large-scale multilingual model to normalize them in several languages. Finally, to compose simple entities into structured entities, we propose a new method based on mention cliques and scope relations. We evaluate it to a new annotated dataset of breast imaging reports.
Keywords: [nlp, structure, extraction, tag, normalize, compose, clinical, multilingual]

Résumé
Les documents cliniques hospitaliers constituent de riches sources d'information pour diverses applications telles que le recrutement de patients pour la recherche clinique, la surveillance épidémiologique, le codage médical et les outils d'aide à la décision. Cependant, étant essentiellement rédigés en langue naturelle, ces documents ne se prêtent pas aisément à des traitements informatiques à grande échelle et doivent d'abord être structurés. Nous visons à extraire les entités mentionnées dans ces documents, qu'elles soient simples ou structurées, c'est-à-dire contenant plusieurs étiquettes ou parties, et à les normaliser selon des bases de concepts. Nous contribuons à plusieurs tâches de traitement du langage naturel (TAL), à savoir la reconnaissance des entités nommées, la normalisation des entités médicales et l'extraction d'entités structurées. Nous nous intéressons notamment à l'entraînement de modèles par apprentissage profond (deep learning) dans des conditions de données limitées, pour des langues autres que l'anglais et dans le domaine clinique. Nous structurons notre approche en trois étapes : surligner, normaliser et composer. Nous proposons d'abord plusieurs méthodes pour surligner des entités simples, notamment lorsqu'elles se chevauchent dans les textes. Nous développons ensuite une approche multilingue à grande échelle pour les normaliser dans plusieurs langues. Enfin, pour composer ces entités simples en entités structurées, nous proposons une nouvelle méthode basée sur les cliques de mentions et les relations de portée. Nous l'évaluons sur un nouveau corpus annoté de comptes rendus cliniques de mammographies.
Mots clé: [tal, structure, extraction, surligner, normaliser, composer, clinique, multilingue]