UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

RECOGNITION AND NORMALIZATION OF BIOMEDICAL ENTITIES WITHIN CLINICAL NOTES

André Alexandre Dias Leal

Dissertação orientada pelo Prof. Doutor Francisco José Moreira Couto
e co-orientado pelo Prof. Doutor Bruno Emanuel da Graça Martins

DISSERTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2015

Acknowledgments

With this work I finish my academic journey. This achievement would never have been possible without the help of so many people that have been part of my life for the past five years. This acknowledgement is dedicated to all of them.

To my family, for supporting me during this journey with so many little details that made my life easier. Thanks for giving your own time so I could have a little more, and for understanding my absence in the past three years.

To my girlfriend, who patiently listened to all my endless self discussions, helping me to find the right solution even without understanding what I was saying. The donkey family appreciated your time. Thanks for holding the boat, now it is my turn.

To my closest friends, who still remember my name after months of silence.

To my University friends, who helped me to reach this academic level. Without you, I wouldn't be here. A special thank for Pedro Abranches, more than a friend was like a teacher to me. Another special thank for Diogo Gonçalves, whose participation in ULisboa team was essential for this thesis results.

Last but not the least, to my supervisors Francisco Couto and Bruno Martins for being always available and for never giving up on me, even when I disappeared from the map.

For the ones that made this possible

”It always seems impossible until it’s one”

Resumo

Os profissionais de saúde, como parte do seu trabalho, têm a obrigação de registar manualmente o seu conhecimento de forma não estruturada, sendo as notas clínicas um dos vários tipos de documentos gerados. As notas clínicas descrevem a situação clínica dos pacientes, contendo informação relativamente aos seus tratamentos, sintomas, doenças, diagnósticos, procedimentos, etc. A introdução desta informação em *Electronic Health Records (EHRs)* está a ser fortemente encorajada, originando um crescimento exponencial no volume de notas clínicas em formato digital. A disponibilização desta informação em formato digital oferece uma maior liberdade, permitindo uma fácil partilha das mesmas entre instituições médicas, acompanhando assim o percurso do paciente.

Nas notas clínicas a informação é registada utilizando a língua natural desprovida de qualquer estruturação. O registo de informação de forma estruturada, apesar de ser recomendado, condiciona o trabalho dos profissionais de saúde. Tal imposição aumenta o tempo necessário para efetuar o registo do conhecimento assim como impõe limites na descrição de casos fora do comum.

A aplicação de técnicas de prospeção de texto (*text mining*) aparece então como solução para o processamento automático da informação não estruturada permitindo a conversão num formato que permita os sistemas computacionais analisar. Dado que os profissionais médicos utilizam diferentes terminologias de acordo com o contexto e a respetiva especialização, o processamento de notas clínicas comporta vários desafios, dada a sua heterogeneidade, ambiguidade e necessidade contextual.

São várias as técnicas de *text mining* utilizadas para resolver estes desafios, sendo neste trabalho exploradas técnicas de aprendizagem automática (*Machine Learning*), semelhança textual (*Pattern Matching*), conteúdo da informação (*Information Content*) e semelhança semântica (*Semantic Similarity*).

O objetivo deste trabalho consiste no estudo e desenvolvimento de um sistema que permita reconhecer e normalizar entidades biomédicas em notas clínicas, assim como o desenvolvimento da respetiva interface. A tarefa de reconhecimento consiste em identificar entidades relevantes em notas clínicas, sendo que a normalização passa pela atribuição, a cada entidade reconhecida, de um identificador único pertencente a um vocabulário controlado. Para tal, o sistema desenvolvido utiliza técnicas de prospeção de texto e usa a ontologia SNOMED CT como vocabulário controlado. Utiliza ainda dois conjuntos de

notas clínicas, um não anotado e outro anotado manualmente por profissionais de saúde. Este último conjunto é referido como conjunto de treino.

O sistema foi desenvolvido usando uma arquitetura modular em *pipeline*, composta por dois módulos, recebendo como *input* um conjunto de notas clínicas não anotadas. A execução do sistema resulta na anotação automática, isto é, no reconhecimento e normalização das notas clínicas recebidas como *input*.

O primeiro módulo é responsável pelo reconhecimento de entidades biomédicas. A estratégia usada consiste na aplicação de algoritmos de aprendizagem automática de forma a gerar um modelo de reconhecimento baseado em casos passados, isto é, notas clínicas manualmente anotadas. O software de aprendizagem automática *Stanford NER* foi utilizado para gerar modelos *CRF* (*Conditional Random Field*). Este módulo comporta dois processos: o de treino e o de execução.

No processo de treino, cada palavra (ou *token*) existente nas notas clínicas é caracterizada com base num conjunto de propriedades entre as quais: *Brown clusters*, formato do *token*, vizinhança e léxicos pertencentes a vários domínios. A caracterização de cada *token* permite que estes sejam representados junto do algoritmo de aprendizagem automática. Este trabalho utilizou o inovador modelo de segmentação *SBIEON*, permitindo a identificação de entidades não contínuas. O algoritmo de aprendizagem automática vai gerar um modelo de reconhecimento baseado nas propriedades associadas a cada *token*.

O modelo de reconhecimento gerado permite identificar entidades em novas notas clínicas não anotadas, associando a cada *token* existente nas respectivas notas clínicas, uma classe pertencente ao modelo de segmentação escolhido. As entidades relevantes são compostas por *tokens* que tenham sido associados a uma classe relevante.

O segundo módulo do sistema é responsável pela normalização das entidades identificadas pelo módulo de reconhecimento como sendo relevantes. Uma arquitetura modular em *pipeline* é utilizada, sendo cada componente responsável pela normalização de um conjunto restrito de entidades pertencentes a um determinado dicionário. Um total de cinco dicionários são gerados baseados nas notas clínicas de treino (abreviações não ambíguas, entidades não ambíguas e entidades ambíguas) e na ontologia SNOMED CT (entidades ambíguas e não ambíguas).

Os primeiros três componentes normalizam as entidades não ambíguos utilizando uma pesquisa de dicionário. A entidade a normalizar é procurada nos dicionários não ambíguos, e caso seja encontrada uma correspondência, o respetivo identificador é associado. O primeiro componente utiliza o dicionário de abreviações, o segundo o dicionário de notas clinicas de treino não ambíguo e o terceiro o dicionário SNOMED CT não ambíguo.

O quarto e quinto componente normalizam entidades ambíguas pertencentes às notas clínicas de treino e ao SNOMED CT respetivamente. Em ambos, uma pesquisa de dicionário é efetuada para recolher os identificadores candidatos. O quarto componente

desambigua as entidades utilizando uma medida resultante da combinação linear do *Information Content* e da frequência do identificador nas notas clínicas em questão. O quinto componente baseia-se em entidades previamente normalizadas num mesmo documento, utilizando uma estratégia baseada na semelhança semântica. A entidade ambígua com maior semelhança semântica é a escolhida, assumindo desta forma que entidades pertencentes ao mesmo documento devem ser semelhantes entre si.

O último componente normaliza entidades que não estejam representadas em nenhum dos dicionários referidos. Técnicas de *Pattern Matching* são aplicadas de forma a identificar a entidade candidata textualmente mais semelhante. Esta entidade é depois inserida no *pipeline* do sistema, sendo normalizada por um dos componentes anteriormente descritos. Para este componente, medidas como o *NGram Similarity* e *Levenhstein* foram utilizadas, tendo esta ultima medida sido estendida de forma a permitir medir a semelhança textual entre duas entidades sem ter em conta a ordem dos seus *tokens* (*ExtendedLevenhstein*).

A interface desenvolvida permite aos utilizadores introduzirem documentos no formato de texto ou através da introdução de um identificador de um artigo no sistema PUBMED ou de um Tweet, sendo efetuada a recolha do texto associado. A interface permite ainda que os utilizadores corrijam ou adicionem novas anotações ao texto, sendo estas alterações registadas pelo sistema. São ainda apresentadas várias estatísticas em tempo real que permitem aos utilizadores navegar entre documentos.

O sistema apresentado neste trabalho é resultante de duas primeiras iterações. A primeira foi utilizada para participar no SemEval 2014 e foi desenvolvida pela equipa ULisboa da qual fui autor principal. A segunda foi desenvolvida por mim no âmbito deste trabalho e foi utilizada para participar no SemEval 2015. Ambas as competições endereçavam a tarefa de *Analysis of Clinical Text*, sendo os sistemas submetidos avaliados oficialmente usando as medidas: *precision*, *recall*, *F-score* e *accuracy*. De forma a comparar o impacto do uso de *machine learning* no reconhecimento, desenvolvi adicionalmente um módulo de reconhecimento baseada em regras, permitindo assim comparar o desempenho de ambas as estratégias.

Além das avaliações oficiais, o sistema foi igualmente avaliado localmente utilizando as mesmas medidas mas recorrendo a um conjunto de notas clinicas diferentes para avaliação. As avaliações permitiram entender o desempenho do sistema ao longo das várias iterações e do seu potencial atual. Foi possível observar que o sistema apresentado atingiu os objetivos esperados, conseguindo reconhecer e normalizar entidades biomédicas com um elevado desempenho.

Olhando para cada módulo individualmente, observou-se que a utilização de algoritmos de *machine learning* permitiu atingir resultados bastante mais elevados no reconhecimento de entidades, do que aqueles obtidos utilizando uma abordagem baseada em regras. Observou-se ainda que a adição de *Brown clusters* como propriedades durante o treino melhorou o desempenho do sistema. A adição de léxicos produziu um efeito

contrário, reduzindo o desempenho.

Olhando apenas para o módulo de normalização, este conseguiu normalizar entidades com uma confiança de 91.3%. Este valor é bastante superior ao obtido pela primeira iteração do sistema que apenas atingiu uma confiança de 60.2%.

O sistema como um todo foi avaliado oficialmente nas competições mencionadas. No SemEval 2014 o sistema submetido obteve o 14º lugar na tarefa de reconhecimento e o 25º na de normalização. Ja no SemEval 2015, o sistema foi capaz de obter o 2º lugar com uma *precision* de 77.9%, um *recall* de 70.5% e um *F-score* de 74%. A avaliação desta ultima competição assumiu o reconhecimento e a normalização como uma tarefa única. Estes resultados mostram que o sistema evoluiu bastante, atingindo um excelente desemepenho. O sistema conseguiu ainda superar os resultados obtidos pelo sistema da equipa UTH_CCB que na edição de 2014 foi a equipa que obteve a melhor classificação.

Este trabalho apresenta um sistema que apesar de usar técnicas *state of the art* com algumas adaptações, conseguiu atingir um desempenho relevante face a outros sistemas a nível global, possuindo um enorme potencial para atingir melhores resultados. Como trabalho futuro, o módulo de reconhecimento poderá ser melhorado através da introdução de novas propriedades que melhorem a definição das entidades relevantes. Alguns componentes da *pipeline* de normalização podem ser amplamente melhorados, aplicando novas técnicas de desambiguação e *pattern matching*, ou mesmo recorrendo a algoritmos *learning to rank* semelhantes ao apresentado pelo sistema de *DNorm* é visto igualmente como uma mais valia.

Palavras-chave: Prospeção de Texto, Análise de Notas Clínicas, Reconhecimento de Entidades, Normalização de Entidades, Aprendizagem Automática, Semelhança Semântica

Abstract

Clinical notes in textual form occur frequently in Electronic Health Records (EHRs). They are mainly used to describe treatment plans, symptoms, diagnostics, etc. Clinical notes are recorded in narrative language without any structured form and, since each medical professional uses different types of terminologies according to context and to their specialization, the interpretation of these notes is very challenging for their complexity, heterogeneity, ambiguity and contextual sensitivity.

Forcing medical professionals to introduce the information in a predefined structure simplifies the interpretation. However, the imposition of such a rigid structure increases not only the time needed to record data, but it also introduces barriers at recording unusual cases. Thus, medical professionals are already encouraged to record the information in a digital form, but mostyl as narrative text. This will increase the amount of clinical notes to process, and doing it manually requires a huge human effort to accomplish it in a feasible time.

This work presents a system for automatic recognition and normalization of biomedical concepts within clinical notes, by applying text mining techniques and using domain knowledge from the SNOMED CT ontology. The system is composed by two modules. The first one is responsible for the recognition and it is based on the Stanford NER Software to generate CRF models. The models were generated by using a rich set of features and employing a novel classification system, SBIEON. The second module is responsible for the normalization, where a pipeline framework was created. This modular framework leverages on a set of techniques such as (i) direct match dictionary lookup, (ii) pattern matching, (iii) information content and (iv) semantic similarity.

The system was evaluated in the SemEval 2015 international competition, achieving the second best F-score (74%) and the second best precision (77.9%), among 38 submissions. After the competition, this system was improved, increasing the overall performance and reducing the running time by 60%.

Keywords: Text Mining, Analysis of Clinical Notes, Named Entity Recognition, Named Entity Normalization, Machine Learning, Semantic Similarity

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As part of their job, medical professionals are expected to record manually their knowledge in an unstructured form, being clinical notes one of the types of information produced. The introduction of this information into Electronic Health Records (EHRs) has been encouraged, leading to an exponential increase in the number of clinical notes available in the electronic form, which encourage the research in this specific domain.

The manual processing of this data requires a huge human effort to not only structure the information, but also to keep up with the clinical notes growth rhythm. Medical professionals are already encouraged to introduce structured information into EHRs, e.g. according to standard terminologies. However, the imposition of a rigid structure may increase the time needed to record data, and it may also introduce barriers to the recording of unusual cases. Narrative notes are thus still commonly employed.

The narrative within clinical notes is abundant in mentions of clinical conditions, anatomical sites, medications, and procedures, motivating the application of text mining methods for resolving entities, within the text, into standardized and computer-processable formats. However, this task comprises several challenges such as the language specificity used within these notes, which differs significantly from the one used in other domains. We have for instance that abbreviations, which depend on contextual factors and on the specialty of the medical professionals typing the notes, are much more frequently used. The type of the entity, which may be disjoint (non-continuous) or even overlapped, is other known major challenge.

More than recognizing entities within narrative textual notes, it is crucial to assign them a semantic meaning by normalizing them to concepts within a known knowledge base. In this task, ambiguity is also a major issue, given that an entity (i.e. the descriptor of a concept) may represent distinct concepts depending on the context.

Properly addressing the challenges related to the recognition and normalization of entity mentions, as they occur within clinical notes, is thus a challenging problem with

many important applications, such as knowledge discovery among other high level tasks.

1.2 Objectives

The work described in this dissertation comprises two distinct tasks: the recognition and the normalization of biomedical entities within clinical notes. The Named Entity Recognition (NER) task is a well-known problem with already proven results in general domain entities, achieving a performance close to the one achieved by manually annotation [17, 57, 75]. A significant amount of research has been made by the scientific community focusing on biomedical text mining, mostly through international workshops.

The main goal of this work consists on taking advantage of the clinical notes available for research, and develop a high performance system capable of recognizing and normalizing biomedical disorders within clinical notes. Besides the clinical notes, the system also leverages on knowledge retrieved from distinct sources, such as ontologies and biomedical domain specific databases.

Hypothesis It is possible to create a high performance system capable of recognizing and normalizing biomedical disorder entities from English notes in electronic health records, using knowledge retrieved from an ontology and by employing machine learning techniques.

The system that I have developed is composed by two distinct modules, one for the recognition and other for the normalization.

Recognition

The first module of the system receives as input biomedical clinical notes in textual form without any annotation associated, and identifies the relevant entities present in those notes. Text mining techniques were employed, namely machine learning algorithms.

Normalization

The second system's module receives as input the entities previously identified and produces as output a set of concepts, i.e., the input entities normalized with a unique identifier from the SNOMED CT ontology knowledge base.

Evaluation

The system was evaluated by using English data specifically created for similar research initiatives, such as the international workshop SemEval by using state of the art evaluation metrics. The system also participated in the SemEval 2015 International Workshop, with the goal to improve the results obtained in the SemEval 2014 edition.

1.3 Contributions

This thesis lead to the following main contributions:

- Scientific publications
  - *SemEval 2014* International workshop article and poster describing the developed system [36].
  - *SemEval 2015* International workshop article describing the developed system. Second best precision and F-score in the competition [37].
  - *Bioinformatics Open Day 2015*[1] abstract submission and oral presentation describing this work [38].
- Recognition and Normalization System
  - Modular system with higher performance and efficiency than the one presented at *SemEval 2015*, which achieved the second best precision and F-score in the competition.
  - Novel pipeline framework for addressing the normalization task, leveraging on novel techniques such as the ExtendedLevenshtein distance.
- System Interface
  - Initial prototype of an interface able to automatically annotate biomedical text files from distinct types of sources, allied with other features like real time global statistics and crowd sourcing data retrieval (user manual annotations).

1.4 Document Structure

The rest of this document is organized as follows.

Chapter 2 (State of The Art) provides an overview of the state-of-art in the area, in particular regarding the recognition and normalization of entities in biomedical texts.

Chapter 3 (The Proposed System) describes all the work developed in the context of my M.Sc., in particular the system developed, its modules, and the different approaches and framework used.

Chapter 4 (Experimental Results) Presents and discusses the results obtained by the system in both official and non-official evaluations.

Chapter 5 (Conclusions) Presents the main conclusions of the work already completed, together with my opinion about future work to be done.

[1] `http://bod2015.ciencias.ulisboa.pt/`

Chapter 2

State of The Art

The number of published scientific publications in the biomedical domain is growing at an increasing rate. In the year 2014, the MEDLINE database, which is one of the largest sources of biomedical documents, was composed by more than 21 million of citations, with more than 800,000 added in the same year[1]. These publications are rich in information, but their expansion rate makes it almost impossible for biomedical researchers to keep up with all the work being published. Thus, retrieving, reading and understanding several scientific articles in order to find something useful for their work is a highly time consuming task.

In the same way that scientific publications are expanding, so is the availability of annotated clinical notes in electronic form. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC II) [10] is a known biomedical database composed by more than 40,000 Intensive Care Units (ICUs) containing information from around 33,000 patients, 20% more than the last version of this database. The ability to efficiently retrieve the valuable information contained in these sources would allow professionals to create systems capable of performing more advanced and high level tasks, such as knowledge discovery, relation extraction and text classification.

Since the amount of available biomedical data continues to increase, the biomedical domain is one of the several domains where text mining techniques are employed in order to automatically retrieve the knowledge within narrative text, being the recognition and normalization of named entities two tasks broadly studied. Although each domain has its singularities, they share the same goal which consists on retrieving the knowledge contained in narrative documents and apply it to real life problems. In biomedical domain one of the goals consists on the improvement of diagnosis, prevention and treatments.

Within the biomedical domain, retrieving information from clinical notes is far less explored than, for example, the same task for genes and chemical entities. One of the main constraints is related to the sensitivity of the clinical notes, as they contain patients' private information. Therefore, the release of such documents, although encouraged, is

[1] `www.nlm.nih.gov/bsd/index_stats_comp.html`

still uncommon, resulting on a restricted amount of clinical notes available for research when compared to other fields. Text mining techniques are also employed for addressing other types of challenges besides entity recognition, such as retrieving and relating information from distinct sources [19, 20, 21]. These tasks are often dependent of recognition and normalization systems, as they use this information as input.

Most of the recent work developed in this domain is based on machine learning approaches, a more time-solid solution which consists on the application of artificial intelligence algorithms. With these approaches, one must use as input previously annotated corpora, to allow the algorithm to learn and generate a successful model. Although data manually annotated has been becoming more accessible, manually annotating a clinical note is still demanding and time consuming task for the domain experts. To overcome this disadvantage, systems leveraging on non-annotated data as source of information were also developed [75]. The search for high performance systems addressing the recognition and normalization tasks benefits from the numerous contributions from international competitions and workshops, such as SemEval2015 [16] and ShARe/CLEF [31].

In this Chapter, I intend to contextualize the work developed and also present an overview of the current state-of-the-art on biomedical text mining. The basic concepts in this domain will be described to fully understand the work developed, with special focus on the text mining techniques that will be used on this dissertation, namely the Named Entity Recognition (NER) and the Named Entity Normalization (NEN).

2.1 Clinical Notes in Electronic Health Records

It is required for medical professionals to record information about their activity, in particular for the clinical care area. The patient information is manly recorded in paper, but it is possible to be recorded in a mix of paper and electronic format, which can be represented with structured, narrative, coded or multimedia entries [30].

Such heterogeneity makes it a challenge to create an integrated view of the health and healthcare history of each patient, regardless the institution or medical professional currently affected [25]. To accomplish this, a person-centered electronic health record (EHR) solution was created, which is defined in the International Organization for Standardization (ISO) as *"A repository of patient data in digital form, stored and exchanged securely, and accessible by multiple authorized users"* [2].

Clinical notes are one of the sources of information contained in EHRs. MIMIC II is an example of an EHRs database containing clinical notes. For example, discharge documents electronically recorded are one type of clinical notes, resuming the stay of the patients at the hospital. As these notes are such a rich source of information, the ability to automatically retrieve information became an important challenge, which should be addressed with a fast automated approach to keep up with the increase rate.

2.2 Text Mining

Text mining can be seen as an extension of data-mining from structured databases. It can be defined as *"The process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents"* [65]. Therefore, text mining tasks operate on a finer level of granularity, focusing on small portions of information contained within plain text documents. It differs from similar tasks like Information Retrieval (IR), Text Summarization (TS) and Natural Language Processing (NLP) where the focus is on the document as a whole (i.e., a bigger level of granularity) [11].

Entity recognition and normalization are two examples of tasks that can be addressed by using text mining techniques like machine learning. These techniques can be applied to several domains through distinct types of sources, for instance, social media domain through news text. Unlike general domain where the extraction of person and place names already has highly accurate results [15], the use of text mining techniques in the biomedical domain is still far from obtaining similar results. The results disparity are due to some intrinsic properties from the biomedical domain, such as the non-existence of a standardized nomenclature (abbreviations, synonyms, etc) and the restrictive amount of annotated data when comparing to the one available for other domains [75]. The existence of abbreviations and synonyms are examples of the lack of standardization.

2.2.1 Tasks

Text mining comprises several tasks which are independent of the domain. Next, the tasks which are most related to this work are described, along with a small description of other tasks not addressed in this work but relevant in biomedical domain.

2.2.1.1 Named Entity Recognition

Named Entity Recognition (NER) is one of the two text mining tasks addressed in this work. NER consists on, as the name suggest, the recognition of named entities within narrative text. The recognition of entities consists on the identification of relevant tokens within the given text, being the entity spans the result provided.

NER can be applied in distinct domains by using narrative texts related to those domains. One of the most studied domains is the general which is newswire text, focusing on the identification of person names, locations and organizations from sources like news, articles, reports, etc. These domains already have systems with performance close to the level of human annotation [17, 57]. The highly availability of annotated data together with the simplicity of the domain are two of the reasons for these results.

In the biomedical domain, works have been developed focusing in the recognition of named genes, protein names and chemical compounds, achieving considerable results

[19, 20]. Systems capable of recognizing general biomedical terms and diseases are not so well studied [11].

There are different types of named entities, according to their structure, possible to be recognized. A narrative text may contain the following types of entities:

- Single Token Entities: All entities which consist on a single token. These are the simplest entities to be recognized. For example: '*The patient had an headache*'.
- Continuous Entities: Entities that consist of two or more continuous tokens. For example: '*The rhythm appears to be atrial fibrillation*'. They are also known as non-single token entities.
- Non-Continuous Entities: One of the major challenges in Named Entity Recognition. It consists on entities which are composed by several tokens (non-single token entities), but between the first and the last entity token there is one or more tokens which do not belong to the entity. For example: '*The left atrium is moderately dilated*'.
- Overlapping Entities: The most complex case. It consists of two distinct entities sharing one or more tokens. For example the sentence: '*His abdomen was soft, nontender, and nondistended*', has the entities '*abdomen nontender*' and '*abdomen nondistended*'. Both entities share the first token.

One of the greatest challenges presented in this task consists in not only be able to find the spans of an entity, but to also recognize their exact boundaries. Mastering this challenge, it is possible to identify where the entity starts and ends, even if two entities are overlapped or next to each other.

Since the manual annotation of narrative documents is highly costly, a good automatic NER system would allow one to efficiently generate annotated data. Therefore, there are some authors who believe that mastering this task would allow to efficiently address more complex text mining tasks, as they often require annotated data to be used as input [11].

2.2.1.2 Named Entity Normalization

After performing a NER task, each recognized entity is nothing more than a descriptor that has been identified as a relevant entity, which means that no semantic meaning has yet been assigned to it. Named Entity Normalization (NEN) can be seen either as an individual task, or as an extension of the NER task [16] where the recognition of an entity is only successful if the correct semantic meaning is assigned to the entity.

In the biomedical domain, normalization can be challenging due to the ambiguity associated to the domain lexicon. The same entity may have different meanings according

to the context. For example, the entity *Pressure*, can mean a physical action or a state of stress. The same descriptor represents distinct concepts.

To accomplish this task, a unique identifier belonging to a knowledge base must be assigned to the recognized entity, giving it a semantic meaning to the entity within the given knowledge base. For example, the entity *Pressure*, used in the previous example, may be recognized multiple times within a clinical note but the descriptor itself does not have any information associated, and thus no information can be retrieved from it.

For example, suppose that the SNOMED CT ontology [13] is used as knowledge base, and suppose that this ontology has a set of identifiers that uniquely identify each concept. Assigning one of these identifiers to a recognized named entity will allow one to retrieve all the knowledge presented in this ontology, associated to that concept (attributes, relations, etc). For instance, the entity *Pressure* when associated to the identifier *C0038435* from this ontology, refers to a state of stress[2].

2.2.1.3 Others Tasks

Although the recognition and normalization tasks are the focus of this work, there are other text mining tasks relevant for the biomedical domain.

Relation Extraction

Relation Extraction (RE) consists on the extraction of relations between relevant entities previously identified through NER techniques. The relation may occur between two or more entities, each one with a specific role in the relation. This task allow one to find relations in several domains, such as *disease to gene*, *disease to treatment*, etc. [11]

For example, the sentence '*Variations in TP53 and BAX alleles are unrelated to the development of pemphigus foliaceus*' contains the information that a given gene/protein has no influence (inverse relation) on a given disease [7].

Text Classification

This task aims to automatically determine if a document, or parts of it, has relevant characteristics to a type of information, and assign it to the respective class or category. [11]. For instance, with a dataset composed by text news, this task can be employed in order to organize those texts into the respective categories (e.g. sports, technology, etc).

Synonyms and abbreviation extraction

When applied to the biomedical domain, this task consists on the identification of synonyms and abbreviations of biomedical entities within narrative text. With the growth

[2] `http://bioportal.bioontology.org/ontologies/SNOMEDCT?p=classes&conceptid=http%3A%2F%2Fpurl.bioontology.org%2Fontology%2FSNOMEDCT%2F262188008`

of biomedical literature, biomedical terminologies have also grown. Since biomedical entities have multiple abbreviations and synonyms, the automatic extraction would benefit the research community as it would allow to automatically update the knowledge base used for other tasks [11].

For example, the entity *'appendix-inflammation'* has the synonym *'appendicitis'* [26] whose automatic identification allows the improvement of a knowledge base, making it more comprehensive. This knowledge can also be used to simplify medical discharges delivered to the patients, by using synonyms more easily understood by them.

2.2.2 Techniques

Several techniques can be applied for addressing the tasks previously described. In the following subsections, the most used techniques are described, focusing the ones that are used in this work and applied for the biomedical domain.

2.2.2.1 Natural Language Processing Techniques

Natural Language Processing (NLP) is a field of computer science focused on the processing of texts written by and for humans. Some techniques used for addressing NLP tasks are also employed in text mining, like for example in the recognition task.

Although both the NLP and text mining fields are related to the processing of narrative texts, and thus share techniques, these two fields are different from each other. For instance, NLP tasks are focused on processing the documents as a whole (higher level of granularity), while text mining is more concern about the detailed information contained in the documents (lower lever of granularity) [11]. Text mining commonly uses NLP techniques to parse the input text into a machine-readable form [12].

The following NLP techniques are some of the most commonly used in text mining systems, and they are also broadly applied in the biomedical domain:

Tokenization

In order to make narrative text processable by a machine, the text used as input must be split into units called tokens. Although these tokens are normally associated to single words, they may also consist in numbers, symbols or even phrases. To retrieve these tokens from the text, a tokenization parser must be employed. This parser will split the input text based on a set of predefined rules. A naive approach would split tokens according to a group of pre-defined delimiters symbols, like spaces, dots, commas, etc. However, this naive approach does not always achieve the best results. Depending on the domain text and structure type, more advance heuristics must be applied in order to improve the quality of the tokens retrieved from the text (e.g. deciding when quotes or brackets are parts of the word). The Stanford Tokenizer [45] and Banner [40] are two examples of systems developed specially for text written in the English language.

This is normally the first step in any text processing system and, although it seems pretty straightforward, the wrong implementation of this process may lead to a poor-performing system [71].

Stemming

To reduce the variability of tokens within the narrative text, a stemming process may be applied. This technique consists on normalizing inflected words to their root form. For example, verbs are normalized to their infinitive form (e.g. *connected* and *connects* will be grounded to *connect*). The Porter stemming algorithm, is one of the most known approaches for the stemming problem, consisting on heuristics used to strip the suffix of the token [55].

Lemmatization

This technique is very similar to the Stemming technique previously described. It is also a solution to reduce the variability of tokens within the narrative text, but while the stemming process is based on a set of heuristic to strip the suffix of the token, this approach takes into consideration the context of the token within the text. For that, part-of-speech tags must be assigned to each token, and a word dictionary with lemmas must be compiled from annotated data. The GENIA Tagger, WordNet, Morpha and BioLemmatizer are examples of systems available to perform the lemmatization process in biomedical text [42].

Part-of-speech tagging

For each word in the text, a part-of-speech (POS) tag is assigned to identify nouns, verbs, adjectives, etc. Since the same word may belong to distinct classes, the label is assigned based on the definition of the word itself together with the context. The Stanford NER Part-Of-Speech Tagger [66] and the GENIA Tagger [67] are implementation examples.

2.2.2.2 Machine Learning

Machine Learning (ML) is a scalable and flexible solution that learns through the *experience* retrieved from past cases, and generates a model capable of resolving new future cases [5, 11, 57, 73].

Two types of corpora can be used as input for the ML algorithm: labelled, also known as annotated, and unlabelled corpora. The existence of these documents is essential for the success of any recognition and normalization system.

- Unlabelled Documents, consists on all raw information available regarding the domain associated to the task. For instance, within the biomedical domain based on

clinical notes, unlabelled documents would consist on a set of raw clinical notes i.e., notes that were not annotated or evaluated by any expert in the area and thus, without any additional information associated. Since no effort is required by experts, these data is less expensive and has a higher level of availability than the labelled data.

- Labelled Documents, are one essential resource for the production of high performance systems. Labelled documents consist on documents containing the relevant information identified in the text. The annotated data is also known as golden standard. These annotations are commonly manually generated by experts in the domain and thus, a great effort is required. This means that not only the data is more expensive, but also that these documents are less available.

  For this work, the annotated data consists on clinical notes with manual annotations identifying the medical entities within those documents. The MIMIC II database is a well-known source of clinical notes that after manual annotation can be used as labelled data.

  This information is essential for the production of recognition models based on machine learning approaches as well as for the generation of some knowledge dictionaries sources.

Machine learning algorithms can be divided in two categories: supervised algorithms and unsupervised algorithms, according to the type of documents used as input.

Supervised Algorithms

Supervised algorithms require, in a first instance, the training of a classification model based on the annotated data received as input. This model can then be used to accomplish an automatic classification task, such as the recognition task [17, 36, 57].

Supervised machine learning algorithms are the most common approach used by recognition systems for the NER task in biomedical domain. As this approach requires biomedical experts to manual annotate a set of documents to be used as training data, a considerable effort is intrinsically associated to this approach. For addressing this problem, some authors developed systems that recognize entities based almost entirely in unlabelled data, reducing the initial effort required [70, 75].

The supervised algorithms are based on the configuration of a set of features that will be used to represent the input knowledge (also known as training data). This representation is processed by the machine learning algorithm which generates a model based on that information [24]. In order to not end up with an overfit or too generalized model, both the features and input knowledge used have to be carefully chosen. The adaptability of changing the settings to be used in this process along with the generation of high performance models, made the machine learning approach widely used in NER tasks [11].

Several supervised machine learning algorithms can be applied, being the followed the most common in biomedical domain:

- Association Rules: based on the annotated data, rules are generated by identifying frequent patterns within the corpora. Generated rules can be as simple as: *if X and Y then Z* [74].

- Decision Trees: based on the features retrieved from the training data, a decision tree is generated. The tree is composed by nodes which represent a condition, links which connect different nodes and leaves which represent classes. Following the decision tree from the root, several conditionals nodes will define the branch to follow and thus the class to be assigned. This type of algorithm is easier to understand but it easily gets over-fit to the training set [3].

- Support Vector Machines (SVM): the data's features are represented as points in a vector space. Input instances are mapped to a high-dimensional feature space where a linear decision model is constructed. This generated model is a spatial representation of the training data, clustering similar categories with the largest gap possible between other distinct categories clusters. [14].

- Conditional Random Fields (CRF): Statistical models which create a sequence segmentation of classes, based on the training data. The statistical model will assign the most probable class to each token. In this learning algorithm, the context is taken into consideration for choosing the right classification class [35].

On supervised algorithms, the annotated data must be pre-processed by using some of the techniques described in Section 2.2.2.1. In this pre-processing step, the tokenization technique is the most relevant one, transforming the input text into a set of individual tokens. For each of these tokens, a specific class from a segment representation must be assigned. A segment representation (SR) consists on a set of classes, or labels, that are assigned to a token within the annotated documents belonging to the training set. The classes have a semantic meaning associated, and will be used as input information for the machine learning algorithm to generate the recognition model. Using this approach, the NER task can be seen as a sequence labelling problem, which aims to retrieve the tokens with a specific label associated.

Segment representations can be separated in two distinct groups: Inside/Outside SRs and Start/End SRs [34]. The first SR group distinguishes terms as relevant or irrelevant. Therefore it is only able to identify single token entities as they do not distinguish the entities boundaries. The second SR group, on the other hand, is able to identify the entity boundaries and for that reason is able to identify relevant non-single token entities.

Next, a list with some of the most commonly used segment representations in named entity recognition are presented along with some examples [8, 34]:

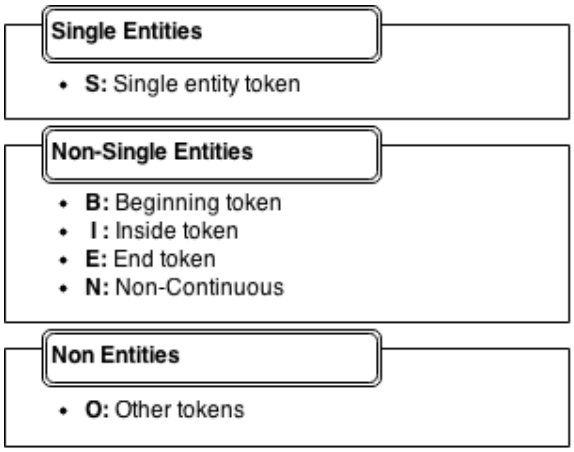

Figure 2.1: Segment representation tags based on the SBIEON classification.

- IO: The most basic segment representation. Each term is tagged as a relevant entity *I* or an irrelevant one *O*. Cannot distinguish adjacent tokens of the same entity, being only able to properly identify single token entities.

- IOB: Allows to distinguish adjacent entities and non-single token entities. Tag *B* represents the beginning of all relevant entities which includes single-token entities, *I* represents the inside and end of a non-single token entity, and finally the tag *O* represents all irrelevant entities.

- SBIEO: A segment representation which belongs to the Start/End group. It allows one to identify single entities by using the *S* tag. Non-single token entities are delimited with the tags *B* and *E*, which represent respectively the beginning token and the end token of the entity. If the entity has more than two tokens, the inside tokens are represented with the tag *I*. All irrelevant terms have the *O* tag.

  For example the sentence '*She experienced severe mental status changes*' would be represented as '*She[O] experienced[O] severe[O] mental[B] status[I] changes[E]*'

- SBIEON: None of the SRs described above allow to represent non-continuous entities. In order to address this challenge, a new segment representation - SBIEON - was created by extending the SBIEO encoding with a new tag, *N* [36]. This tag represents all non-continuous tokens. These tokens are the ones that are inside of a relevant entity but are not part of it. Figure 2.1 resumes the tags used in this segment representation.

  For example the sentence '*The left atrium is moderately dilated*' would be represented as '*The[O] left[B] atrium[I] is[N] moderately[N] dilated[E]*'. The tokens '*is moderately*' are identified as non-entity.

Unsupervised Algorithms

Unsupervised machine learning algorithms are based on unlabelled data and therefore, no processing or additional effort on the part of domain experts is required to set up the input data. These algorithms are used to detect structures within the narrative input data. The most commonly used unsupervised algorithm within text mining is clustering, that as the name suggest, implies the creation of clusters [54]. A cluster is nothing more than a group of entities with similar features. Therefore, each cluster represents a set of unique entities as a whole.

Brown clusters is one known clustering algorithm, used for grouping related words. According to Brown [6], this technique allows to reduce the sparsity of the data, generating a lower-dimensional representation of the unlabelled data used as input. For example, suppose that 1,000 documents are used as input with a total of 1,000,000 unique tokens. By using the Brown algorithm to generate 100 clusters, the one million tokens used as input will be represented by the 100 generated clusters, which comprise them all. Each cluster is composed by the entities which maximize the mutual information of bigrams. Being a bigram a sequence of two adjacent characters in a given token, the more bigrams the two tokens share, the more similar they are. In the end, each cluster will be composed by similar tokens.

By creating this lower-dimensional representation, it is possible to use the knowledge in the input documents to improve the performance of the recognition system as it can create new features for each token during the recognition process by leveraging on the cluster that each token belongs to [69]. Two tokens that belong to the same cluster will be considered similar.

Other Approaches

Besides the machine learning approach, other approaches can be followed in order to address NER tasks.

- Dictionary Based: The most naive approach consists on resolving the NER task as a dictionary matching problem. This approach relies on a pre-built dictionary containing named entities to be recognized and thus, no annotated data is necessary. The process consists on scanning the narrative texts and performing a direct comparison, using string matching techniques, between each token within the text and all dictionary entries.

  Although this approach might achieve reasonable results, it has some flaws. Since the process is based on a direct match, the system's performance will be intrinsically related to the quality and comprehension of the dictionary. Since in the biomedical domain new terminologies are produced continuously and there are several synonyms and abbreviations for each entity, the creation of one such dictionary is a really demanding task [70].

For its simplicity, this approach is the easiest to implement, having the advantage to allow an easy normalization of the entity by simply assigning the identifier assigned to the concept presented in the dictionary. However, such an approach does not always produce the best results. Even with a dictionary contemplating all the named entities to be recognized, the system would still fail as it would not be able to resolve ambiguous entities.

- Rule Based: Other possible approach consists on defining a set of rules or regular expressions that will be applied to the input text to recognize entities. Such rules represent the knowledge retrieved by biomedical experts when manually annotating the biomedical texts. The process of manual annotating entities within biomedical text is by itself a time-consuming task which gets worse with the need to create the rules. There are also always many exceptions to each rule, being almost impossible to map every case in a single rule. Like the dictionary approach, where the generated dictionary needs to be constantly updated, in this approach new rules need to be created and continually updated.

2.2.2.3 Pattern Matching

In the previous section, this technique was briefly described as an alternative approach for the NER task, more precisely by using a dictionary based approach. String similarity algorithms can be used for both the recognition and the normalization task. For instance, a given recognized entity might not match any known concept descriptor within any known knowledge base and, thus, the most similar concept must be retrieved. For that, pattern matching algorithms can be employed, retrieving the most similar concept from the knowledge base, based on the similarity of its descriptor and the recognized entity.

These algorithms are based on the assumption that similar strings are semantically related between them. Several algorithms were developed for addressing this problem by calculating the distance between two strings and therefore their similarity. For instance, if the entity *pain in joint* was recognized within the text, and the entities *painful joint* and *pain in back* were candidates retrieved from the knowledge base, it is necessary to know which one has the higher similarity. In this case, the concept with the descriptor *painful joint* should have the higher similarity.

The similarity is usually normalized between the values 0 and 1, being the first a result when two strings are identical and the second when they are totally distinct. Several distinct algorithms are available [9]:

Levenshtein

Levenshtein distance, also known as edit distance, is the best known string distance algorithm which is defined as the minimum number of elementary operations (insertions,

deletions or substitutions) required to transform one string into another. Each operation has a unitary cost which is calculated using dynamic programming algorithms. The similarity measure can be obtained through:

$$\mathrm{Sim}_{\mathrm{ld}}(S1, S2) = 1.0 - \frac{\mathrm{dist}_{\mathrm{ld}}(S1, S2)}{\mathrm{Max}(|S1|, |S2|)}$$

where $\mathrm{dist}_{\mathrm{ld}}$ is the distance between the two strings according to the Levenshtein distance, and $|S1|$ and $|S2|$ are the length of strings $S1$ and $S2$, respectively. The less operations needed to perform the transformation, the most similar the strings will be. Since an elementary operation is performed at the character-level, this similarity algorithm has a very fine granularity.

n-Gram

n-Grams are sub-strings of size n retrieved from a string to compose a longer one. n-Grams of size 1 are known as uni-grams, of size 2 as bigrams, of size 3 as trigrams, and so on. For instance, the entity *pain* contains the bigrams 'pa', 'ai' and 'in' and the trigrams 'pai' and 'ain'.

The distance between two strings consists on counting the number of n-grams that the strings have in common. The more n-grams in common, the most similar the strings are. The similarity measure is obtained by dividing the number of n-grams in common by the number of n-grams in the shorter string (also known as Overlap coefficient), the union of n-grams in both strings (also known as Jaccard similarity) or by the average number of n-grams in both strings (also known as Dice coefficient) [32, 9].

This algorithm definition allows to compare two strings at character-level (as in Levenshtein distance) but also allows to perform the comparison with larger windows including a sequence of characters (n-grams).

Jaro

This algorithm takes into consideration the insertions, deletions and transpositions. A window no longer than half of the size of the biggest string is defined as the "matching distance". Two characters are counted as a match if they are within the matching window and if they are the same. A transposition consists of characters that are a match within the matching window, but with a different sequence order. For instance *Headache* and *Heaadche* have one transposition [27, 28].

The similarity measure is obtained using:

$$\mathrm{Sim}_{\mathrm{jaro}}(S1, S2) = \begin{cases} 0, & \text{if} |m| = 0 \\ \frac{1}{3} * \left(\frac{m}{|S1|} + \frac{m}{|S2|} + \frac{m-t}{m} \right), & \text{otherwise} \end{cases}$$

where m is the number of matches, t the number of transpositions and $|S1|$,$|S2|$ the length of both strings respectively. The matching window is calculated through:

$$\text{MatchingWindow} = \left[\frac{\text{Max}(|S1|, |S2|}{2}\right] - 1$$

Winkler

Also known as Jaro-Winkler distance, this metric was an improvement over the Jaro algorithm by Winkler. It is based on studies which defended that the initial part of the strings are often more representative (they have for example, less errors) [72]. Based on this assumption, Winkler developed the following formula to increase the Jaro performance by taking into consideration an initial number of characters:

$$\begin{aligned}\text{Sim}_{\text{wink}} &= \text{Sim}_{\text{jaro}}(\text{S1}, \text{S2}) \\ &+ \frac{s}{10}(1.0 - \text{sim}_{\text{jaro}}(\text{S1}, \text{S2}))\end{aligned}$$

where s is number of initial characters identical between the two strings $S1$ and $S2$. For example, *headache* and *heavy* have an s of value 3.

Cosine-Similarity

This algorithm calculates the similarity between two vectors of attributes. Similar vectors will be close to each other in the space model and thus, have higher similarity [18]. The similarity value is calculated using the inner product space of the vectors, derived from the Euclidean formula:

$$\text{Cosine\_Similarity} = \cos(\theta) = \frac{A * B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2 * \sum_{i=1}^{n}(B_i)^2}}$$

where θ is the angle between the vector A and B, A_i and B_i correspond to a single vector attribute in the index position i. $||A||$ and $||B||$ represent the magnitude of each vector. When two vectors are identical, they have a 0° angle between each other. As $\cos(0^\circ)$ is equal to 1, and since for any other angle, the value will always be smaller, it is possible to define the similarity according their inner angle. The smaller the angle, the higher the similarity and therefore the closer the vectors are in space.

Considering a string as a document representing it in a vector space model (VSM) allows one to employ information retrieval techniques and thus use this algorithm to compare strings and retrieve the most similar.

2.2.2.4 Information Content

Ontologies consist on a set of concepts within a given domain that are represented by their properties and the semantic relations between them [22]. The relation *is-a* is one of the most common semantic relations present in ontologies. This relation allows to identify concepts which are more general (closer to the root node) and concepts which are more specific (the ontology's leafs). The Information Content (IC) represents a measure of how informative a concept is, meaning that more specific concepts have higher information content, and more general concepts less information content.

Two approaches can be followed to calculate the information content of a specific concept in an ontology. One of the first approaches was introduced by Resnik [58] which defines the information content as the negative log likelihood:

$$\mathrm{IC}(c) = -\log \mathrm{p}(c))$$

where $\mathrm{p}(c)$ is the probability of a given concept descriptor c being found within a specific corpus.

This approach has some implications, since it imposes the existence a considerable amount of unlabelled data to calculate the frequency of each concept. This dependency may also generate some biased results due to the corpus characteristics and also create data sparseness if the corpus size is considerable. Another approach consists on using the ontology descriptors exclusively, to estimate the frequency of a given concept, removing the coupling relation to the unlabelled data. This corpus free approach is known as intrinsic information content [63].

The main intrinsic IC algorithms are described next:

Seco

$$\mathrm{IC}(c) = 1 - \frac{\log(\mathrm{hypo}(c) + 1)}{\log(\mathrm{Max}_{nc})}$$

where hypo returns the number of hyponyms (children) of the concept c, and $\mathrm{Max_{nc}}$ is a variable that defines the maximum number of concepts that exist in the ontology. Seco estimates the frequency of a given concept by summing the number of children together with the concept itself [63].

Zhou

$$\mathrm{IC}(c) = k(1 - \left(\frac{\log(\mathrm{hypo}(c) + 1)}{\log(\mathrm{Max}_{nc})}\right) + (1 - k)\left(\frac{\log(\mathrm{deep}(c))}{\log(\mathrm{deep}_{max})}\right)$$

This formula is composed by two branches, where the first one consists on Seco's formula. So, hypo represents the number of children c, and $\mathrm{Max_{nc}}$ the maximum number of concepts in the ontology. $\mathrm{deep(c)}$ represents the concept c depth, and $\mathrm{deep_{max}}$ the maximum depth of the ontology. k is a variable used to balance the

two branches in this formula. With the addition of the second branch, this algorithm takes in consideration the relative position of the concept within the ontology, differentiating concepts with different levels of generality [77].

Sanchez

$$\mathrm{IC}(c) = -\log\left(\frac{\frac{|\mathrm{leaves}(c)|}{|\mathrm{subsumers}(c)|} - 1}{\mathrm{max.leaves} + 1}\right)$$

where $\mathrm{leaves(c)}$ represents the number of leaves of c, $\mathrm{subsumers(c)}$ represents the number of parents of c, and $\mathrm{max.leaves}$ is the number of nodes which are leaves in the ontology. This algorithm is considered an improvement when compared to the algorithm from Seco and Zhou [61], since it takes into consideration the relation between the number of parents with the number of children of a given concept. In this way, this algorithm not only differentiates concepts within different level of generality but also reduces the ontology dependence.

2.2.2.5 Semantic Similarity

Given two concepts from a given ontology, semantic similarity measures can be applied to return a numerical value representing the closeness in meaning between those two concepts, or alternately their distance. This measure allows one to compare the similarity between two concepts. It is assumed that concepts with high semantic similarity will more likely be related.

The semantic similarity algorithms are intrinsically related to the information content of the compared concepts. Any of the previously mentioned IC algorithms can be applied in the following algorithms:

Resnik

$$\mathrm{Sim}_{\mathrm{res}}(c1, c2) = \mathrm{IC}(\mathrm{LCS}(c1, c2))$$

where LCS (Least Common Subsumer) represents the most specific ancestor that both concepts $c1$ and $c2$ have in common. IC represents the information content of the given concept. This algorithm assumes that the more specific the common ancestor of two concepts is, the most similar those two concepts are [58].

This algorithm fails to differentiate pairs of concepts that have the same common ancestor resulting in the same similarity value, even though their distance to the common ancestor is different.

Lin

$$\mathrm{Sim}_{\mathrm{lin}}(c1, c2) = \frac{2 * \mathrm{IC}(\mathrm{LCS}(c1, c2))}{\mathrm{IC}(c1) + \mathrm{IC}(c2)}$$

Lin improved the Resnik similarity algorithm by introducing the ratio between the IC of the common ancestor (i.e. the information that the concepts have in common) and the IC of both concepts (i.e. the information needed to fully describe them)[41].

Jiang and Conrath

$$\mathrm{Sim}_{\mathrm{j\&c}}(c1, c2) = (\mathrm{IC}(c1) + IC(c2)) - 2 * \mathrm{IC}(\mathrm{LCS}(c1, c2))$$

Similar to the Lin algorithm, Jiang and Conrath proposed a formula which calculates the distance between two concepts instead of the similarity. This algorithm result will consist on the distance between the information of the common concept and the information contained by both concepts. Unlike the previous algorithm where high values represented higher similarities, in this algorithm, the lower the result the higher the similarity [29].

2.3 Tools

Distinct implementations are available for the algorithms described in this work. Machine learning implementations will have a special focus, due to their popularity on existing text mining solutions for the biomedical domain. Several machine learning implementations are available and although some of them were developed for a specific domain, they can be adapted for other domains when using the appropriate corpus as input. In this section, some of the most common used software solutions are briefly described.

2.3.1 Stanford NER

Stanford NER is an open source software which implements a machine learning algorithm by employing a linear chain conditional random field (CRF) approach for building probabilistic models based on training data. Since it leverages on the existence of annotated data, it is considered as a tool following a supervised approach.

This implementation incorporates, in addition to the local features commonly used in this type of approach, non-local features by using Gibbs sampling [17]. This allows the classification to be performed using more of the information that is available in the text, even if not locally. Stanford NER was developed specially for the recognition of general domain entities, such as person names, cities, etc. It allowed an error reduction of 9% from the previous state of the art systems.

Stanford NER allows the definition of a set of features to be used for the model training, like the addition of unsupervised approaches (clusters) and normal token features such as the window size (number of tokens which defines the local feature windows), token shape, token Part-of-Speech, etc. Distinct segment representations are also possible to employ.

Although Stanford NER was not specially developed for the biomedical domain, it can be applied to this specific domain [36] and achieve promising results.

2.3.2 Apache cTakes

Clinical Text Analysis and Knowledge System (cTakes) is an open-source system developed specially for the information extraction from electronic clinical records. This software is composed by distinct modules commonly used in NLP tasks: *Sentence boundary detector*, *Tokenizer*, *Part-of-Speech tagger* and NER annotator. cTakes follows a sequential approach where each module is executed individually based on the output of the previous module. The recognition module is implemented by using a terminology-agnostic dictionary look-up algorithm, where each named entity is mapped to a specific dictionary like for example, a dictionary generated based on the SNOMED CT ontology [62].

2.3.3 MetaMap

Software developed by the National Library of Medicine (NLM) to map biomedical text to the concepts presented in the UMLS Metathesaurus. The MetaMap algorithm consists on the parsing of the text from which some tokens variations and synonyms are derived from the recognized entities. In addition to the derived tokens, more candidates are retrieved from the available metathesaurus. Each candidate is them evaluated by calculating the strength of the mapping between the named entity and the candidate entity. The most appropriate candidate is chosen as the primary normalization solution [4].

2.3.4 CRFSuite

CRFsuite is a Named Entity Recognition toolkit which was designed as a pipeline system composed by distinct NLP modules from different implementations such as tokenizer, POS-tagger and lemmatizer. GENIA-tagger implementation is employed to perform the tokenization, POS-tagger a lemmatization of the text. Machine learning is employed to train a recognition model. This software has achieved good results in some NER tas from the biomedical domain [53].

2.3.5 BANNER

Banner [40] is a NER system specially developed for the biomedical domain. This system follows a three-stage pipeline approach: The first stage (i) consists on the tokenization of the input data. Next (ii) a set of features like part-of-speech tags to be processed by the machine learning algorithm are generated for each token by using the Dragon toolkit. Finally, (iii) a CRF model is generated based on the input features provided on the previous task.

2.3.6 Lucene

Lucene [44] is one software package including all the pattern matching algorithms previously described in Section 2.2.2.3. This package allows the creation of index words from dictionaries based on a given pattern matching algorithm. This results in an easy and fast search for the most similar entities according to a specific chosen algorithm and within an indexed dictionary.

2.4 Corpora and Datasets

All algorithms demand the existence of a corpus to be used as knowledge base input. Dictionary domain lexicons, ontologies, labelled data and unlabelled data are some basic corpora often used for the recognition task. Without them it is impossible to execute some algorithms, such as the ones using the machine learning approach. For the normalization task, it is necessary to have a knowledge base, for example an ontology, in order to normalize the recognized entities within a set of identifiers described on that specific environment. The relations in the knowledge base can also be used for specific algorithms such as information content and semantic similarity. In this section the main corpora and datasets used in biomedical text mining are described.

2.4.1 Ontologies and Databases

The quality and quantity of the information available [11] is essential for the development of high performance systems. Many efforts are involved in order to create such knowledge for the most different domains, including the biomedical one. The knowledge can be represented using distinct formats, from the most basic which consists on simple lists of terms stored in databases, to structured ontologies.

An ontology consists on a data structure build to represent concepts and their relations within a given domain of interest [22]. Thus, an ontology is nothing more than an explicit specification of a conceptualization of the specific domain that it is intended to represent [23]. Each concept is represented by a unique identifier within the ontology along with a set of properties that characterizes it (e.g., concept descriptor or label, concept definition, etc.) and a set of semantic relations between them. These relations, together with the properties, provide the meaning for the ontology and all the concepts within it.

Next, some of the best known databases and ontologies in the biomedical domain will be briefly described, with a special focus on the ones used in this work.

2.4.1.1 MEDLINE Database

Created by the United States National Library of Medicine's (NLM) in 1946, MEDLINE is a database with more than 21 million references to journal articles related to the biomed-

ical domain. The citations are from more than 5,600 international journals in about 40 languages. An average of 3,000 references are added daily contributing to the increase of this dataset [51]. MEDLINE is one of the main biomedical databases available.

2.4.1.2 PubMed

PubMED is a free access articles database created in 1996. This database contains more that 23 million citations for biomedical literature from life science journals, online books and MEDLINE, which is one of the primary components. Several fields from the biomedical domain are incorporated in this database such as medicine, healthcare system and preclinical sciences. The PubMed database can be seen as a free access extension of the MEDLINE database, since this last represents the largest subset of the PubMed [49, 48].

2.4.1.3 MeSH

The articles within MEDLINE/PubMED are indexed with specific keywords to a structured database called Medical Subject Heading (MeSH) [43]. MeSH is composed by terms naming descriptors organized in a structured way where root descriptors are more abstract and leaf descriptors more specific. This database allows one to search for articles within MEDLINE/PubMed. MeSH is not an ontology, since the structure is based on links and not semantic relations as expected in an ontology. The 2014 MeSH release was composed by 27,149 descriptors [47].

MeSH is an essential tool to perform relevant and more complex searches through all available references in the MEDLINE/PubMED database, allowing one to research the most relevant sources about a given subject through a set of queries.

2.4.1.4 SNOMED CT Ontology

The Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) is the most comprehensive healthcare terminology, containing more than 1 million descriptors for around 200,000 concepts. It contains information mainly in the English language, although it already has some support for Spanish [13, 52].

The SNOMED CT ontology comprises a significant number of clinical terms used in EHRs and, thus, it is an important source of information with enormous value for EHRs studies. As this information is structured in a computer processable way, information systems can easily leverage on this source of knowledge to retrieve information intrinsically related to the concept itself, or to the relations that the concept shares with other concepts in the ontology.

2.4.1.5 Unified Medical Language System

The Unified Medical Language System (UMLS) is, as the name suggests, a unified ontology based on several knowledge bases. UMLS (2014AB release) contains approximately 10 million concept descriptors for more than 3 million concepts within more than 130 families. In total, this ontology is composed by 21 different languages, being English the most abundant source (i.e. making 75% of the ontology). Portuguese, on the other hand, represents 1.38%[3]. These values have greatly increase since 2004, where there were only 900,000 concepts in the ontology compared with the 3 million concepts within the new release [50].

UMLS was not exclusively developed for the biomedical domain, but some of the most known sources of biomedical information are integrated into it. MeSH and SNOMED CT are examples of biomedical ontologies comprised within UMLS.

UMLS is composed by three separated sources: the metathesaurus, the semantic network and the Specialist Lexicon [1]. The metathesaurus source is composed by terms and concept definitions from distinct languages, as well as their relations. The metathesaurus represents the main source of information from this ontology.

The semantic network consists of a set of semantic categories. Each concept within UMLS has at least one semantic type associated. The semantic network also provides semantic relations between concepts, improving in this way the amount of information possible to retrieve from the ontology. The semantic network contains 133 semantic types and 54 semantic relationships. Semantic relations can be for example of type: *is a*, *inverse is a*, *part of*, *occurs in*, among several others. The relations *is a* and *inverse is a* are the most common, and are used to create the notion of specification and abstraction of a given concept. For instance, if the concept *C1* has a relation *is a* with the concept *C2*, it means that the concept *C1* is a specification of the concept *C2*.

The specialist lexicon provides lexicon information, which includes an English lexicon together with many biomedical vocabulary. It is essentially used as source information for NLP systems.

Each concept within the UMLS ontology is identified with a unique identifier. The concepts also have a set of attributes associated, being the set of possible descriptors an example. A UMLS unique identifier is referred as *CUI*, being in the format *CXXXXX*. For example, the concept *Pressure* has the unique identifier *C0038435*.

2.4.1.6 MIMIC II

The Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC II) is a free public database composed by tens of thousands of Intensive Care Unit (ICU) patients [59, 60].

[3] `www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/statistics.html`

This database is an extension from the previous MIMIC database, and it has been growing largely through time. The first version, named MIMIC, was created in 1991, being used till 2000. In 2001, a new version named MIMIC II was created being the one used at the moment. The MIMIC database contained around 90 subjects with a total of 121 records where the newer version, MIMIC II (release 2.6 April 2011) contains around 33,000 subjects with more than 40,000 ICU's stays [10]. At the present, MIMIC II comprises the MIMIC database. Comparing the previous MIMIC II release (2.5) with the present one, it is possible to observe an increase around 20% both in patients and ICUs stays[4].

MIMIC II version 2.6 (April 2011) consists of three distinct databases[5]:

- MIMIC II Waveform Database: Includes the waveform digitalized signals from exams like ECGS, respiration etc. It contains more than 23,000 record sets for around 13,000 ICU patients

- MIMIC II Waveform Database Matched Subset: Around 5,000 waveform records and numeric records from MIMIC II Waveform Database, which have been matched and aligned with around 3,000 MIMIC II Clinical Database records.

- MIMIC II Clinical Database: Clinical records from around 33,000 patients. These records contain results of laboratory tests, medications and notes & reports. The last one contains discharge summaries, nursing progress notes, ECGs, etc.

  For this work, this is the most relevant database from MIMIC II. This specific database of records has a considerable amount of clinical notes rich in information regarding the medical state and progress of the patient, while he was under observation.

2.5 Performance Assessment

Distinct systems, when executed under the same circumstances, should be compared using standard measures in order to allow a fair comparison between them. For systems that use, for example, different corpora as training data, no rigid comparison is possible, since their results are inherently related to the corpora used for training and evaluation.

Clinical NLP are evaluated according to two distinct measures, namely precision and recall, that can be combined in a third possible measure, the F-score. The normalization systems can also be evaluated according to their accuracy. These measures can be described as a class match problem where the notion of *true positive*, *true negative*, *false positive* and *false negative* is required. The following confusion matrix describes each one of these concepts.

[4] `http://physionet.org/mimic2/mimic2_statistics.shtml`
[5] `physionet.org/physiobank/database/`

predicted class	golden class	
	positive	negative
positive	**true positive**	false positive
negative	false negative	**true negative**

Table 2.1: Confusion matrix to represent evaluation measures.

The concepts presented in the Table 2.1, are a result of the relation between the predicted class (the one assigned in the process) and the golden class (the correct assignment). The concepts can be described as follows:

- True Positive: If the identified class is present in the golden file. A true relevant class was identified as intended.
- True Negative: If The class is not present in the golden file, and the system did not identified it. Both assumed it as an irrelevant class.
- False Positive: If the identified class is not present in the golden file. The class was incorrectly identified.
- False Negative: If the golden file class has not been identified by the system. The class was supposed to be identified, but the system did not.

The following measures are defined based on the concepts previously described, where in this work, a class can be an entity (recognition task) or a unique identifier (normalization task).

Precision represents the correctly identified classes among all the classes that were identified as relevant. This value is maximum when no class was wrongly assumed as relevant (false_positive). Therefore, all the identified classes are present in the golden file.

$$\text{Precision} = \frac{\text{true\_positives}}{\text{true\_positives} + \text{false\_positives}}$$

For example, if four classes are identified as relevant, and if those classes are present in the golden file, than a 100% precision is achieved. If one of them was wrongly predicted (not present in the golden file), than the precision drops to 75%.

Recall represents the classes that were correctly predicted from all the golden standard classes. This value is maximum when the predicted classes covers all the golden standard classes.

$$\text{Recall} = \frac{\text{true\_positives}}{\text{true\_positives} + \text{false\_negatives}}$$

For example, if the golden file consists on four classes, and all are predicted, than a 100% recall is achieved. If one of them was not predicted, than the recall value drops to 75%.

F-score is a balanced combination of the previous two measures.

$$\mathrm{F-score} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Accuracy is the measure used by the SemEval 2014 competition [56] to evaluate the normalization systems.

$$\mathrm{Accuracy}_{\mathrm{strict}} = \frac{\mathrm{true\_positives} \cap \mathrm{N}_{\mathrm{correct}}}{\mathrm{T_g}}$$

$$\mathrm{Accuracy}_{\mathrm{relaxed}} = \frac{\mathrm{true\_positives} \cap \mathrm{N}_{\mathrm{correct}}}{\mathrm{true\_positives}}$$

The previous formulas represent the *strict* and the *relaxed* evaluation, where $\mathrm{T_g}$ represents the total number of disorder mentions in the Golden standard and $\mathrm{N}_{\mathrm{correct}}$ the number of entities which where correctly normalized.

The strict relaxed evaluation allows to evaluate the system performance taking into consideration the NER task. The relaxed evaluation measures the amount of correctly assigned identifiers, assuming only the correctly identified entities (takes only in consideration the true positives to ground the results).

For example, assuming that the golden file consists on ten classes, and only six are predicted (60% of recall), if only three of those six entities had the correct identifier assigned, then the strict accuracy would be of 30% and the relaxed accuracy of 50%.

2.6 Evaluation Series

Several international evaluation series exist with the purpose of stimulating the search on several fields such as text mining in the biomedical domain. These competitions consist on a series of evaluations to test new systems and approaches for the present tasks and challenges.

In this section, two well known international competitions regarding the recognition and normalization of biomedical entities, will be provided together with a brief overview of some of the systems developed for those competitions.

2.6.1 SemEval 2014 Workshop

The Semantic Evaluation (SemEval) competition was created with the aim to stimulate the exploration of the meaning in narrative texts. The first edition took place in Sussex, in 1998 and the last one in 2015 at Denver. SemEval comprises several task in distinct domains. For example, the SemEval 2014 edition (which was hold in Dublin) proposed

	Train	Development	Test
Notes	199	99	133
Words	94k	88k	153k
Disorder mentions	5,816	5,351	7,998
CUI-less mentions	1,939 (28%)	1,750 (32%)	1,930 (24%)
CUI-ied mentions	4,117 (72%)	3,601 (67%)	6,068 (76%)
Contiguous mentions	5,165 (89%)	4,912 (92%)	7,374 (92%)
Discontiguous mentions	651 (11%)	439 (8%)	6,24 (85)

Table 2.2: Distribution of annotated data in terms of notes and disorder mentions across the training, development and test sets in SemEval 2014 [56].

10 different tasks such as the *Analysis of Clinical Text* [56]. This task is related to the biomedical domain, and consists on first, recognizing disorder mentions within narrative clinical notes, and then normalizing them with a UMLS unique identifier limited to the SNOMED CT source only. For the recognition task 21 teams participated with a total of 43 submissions, while for the normalization task only 18 teams participated, resulting in 37 submissions.

2.6.1.1 Resources

For the *Analysis of Clinical Text* task, both labelled and unlabelled data were provided. The released corpora consisted on documents from the MIMIC II database. The documents belonged to distinct fields: discharge summaries, electrocardiograms, echo-cardiograms and radiology reports. Labelled documents were manually annotated and normalized with a unique identifier from UMLS, limited to the SNOMED CT ontology. This corpora was also used in ShARe/CLEF evaluation lab.

Labelled Data

A set of manually annotated documents was provided to the participants. The annotated data that was released is described in Table 2.2. This corpora was released in sets for both training (Train corpora) and testing (Development corpora).

Unlabelled Data

The organization released unlabelled notes belonging to the MIMIC clinical notes database. This corpora was provided to allow the participants to employ unsupervised techniques. Surprisingly, only two teams [36, 33] apply this amount of data to generalise lexical features. The team UTH_CCB used of-the-shelf Brown clusters [76].

Knowledge Base

For this competition, only the SNOMED CT [13] subset within the UMLS ontology 2012 ab version was taken into consideration. Therefore, it should only be assigned a CUI to a disorder entity belonging to this specific ontology subset. All other entities, even if they can be normalized to other categories or ontology subsets, must be assigned as CUI-Less.

2.6.1.2 Performance Assessment

This competition leveraged on the metrics described in the Section 2.5 to evaluate the submitted systems for both the recognition and normalization task. For the recognition evaluation, a relaxed and strict evaluation was performed for the precision, recall and F-score metrics, where.

- Strict: The recognized spans must be a *perfect match* with the golden standard spans. For example: The recognized entity *Head pain* is a perfect match with the golden standard *Head Pain*
- Relaxed: The recognized spans must *overlap* the golden standard spans. For example: The recognized entity *Pain* overlaps the golden standard *Head Pain*

2.6.2 ShARe/CLEF eHealth

ShARe/CLEF eHealth is an evaluation lab with special focus on the medical domain. This competition was created in 2013. In 2014 a second edition was held, promoting the continuity of the research started in the first edition. In this last edition, three tasks were proposed: The first task, information visualisation, which aims to help patients to read and understand their discharge summaries. The second task, information extraction, is a continuation of the previous edition task, which was focused on the NER and NEN of medical disorders from clinical notes. The third and last task, information retrieval, is focused on retrieving from a set of documents, the ones with relevant information according to the executed queries [31].

The 2013 evaluation lab edition held a task which aimed for the recognition and normalization of disorder mentions in clinical notes. For that reason, this edition lab is the one which is more relevant for this work [64].

Detailed information about the 2013 edition is given next:

2.6.2.1 Resources

This evaluation lab released a set of clinical notes belonging to the MIMIC II knowledge base. This clinical notes were manually annotated and were normalized with a unique

identifier from UMLS, limited to the SNOMED CT ontology.

Labelled data

For the recognition and normalization task, 200 labelled notes were release for training the systems. This corpora belongs to the MIMIC II database and it is identical to the one released for the SemEval 2014 edition [56].

Unlabelled data

No unlabelled data was provided with the exception of the 100 documents released to evaluate the system.

Knowledge Base

The SNOMED CT ontology was the one used.

2.6.2.2 Performance Assessment

The systems were evaluated based on their precision, recall, F-score and accuracy according to the metrics described in the Section 2.5. Both strict and relaxed types of evaluation, described in Section 2.6.1.2, were performed.

2.6.3 Systems

The evaluation series that were previously described encouraged the development of recognition and normalization systems. In this section, some of the most relevant systems are briefly described and compared.

2.6.3.1 ULisboa

I, as main developer of the team ULisboa, participated in the SemEval2014 edition where a recognition and normalization system was developed.

The ULisboa [36] recognition system is based on a CRF model generated through the Stanford NER software. Stanford NER was set up with a full set of annotated corpora which was automatically tokenized using a set of rules. To resolve the non-continuous entities challenge, the Stanford NER software was extended to support the SBIEON segment representation developed within this work. This SR was described in the Section 2.2.2.2. It has an extra tag *N* to represent tokens inside entities that are not part of it. Besides the common features used in machine learning algorithms, Brown clusters and domain lexicons were also employed as additional features.

Brown clusters were generated from the unlabelled data provided, which consisted on more than 400,000 MIMIC II documents containing millions of tokens. These tokens were clustered into 100 distinct clusters, that were used to represent the entire corpora

during the training process. The domain lexicon was generated from lists with names of drugs and diseases retrieved from DBPedia.

The normalization component was based on a Pattern Matching approach. Leveraging on the Lucene package, the best candidates from SNOMED CT were retrieved according to a sequence of three algorithms: first, according to the n-gram distance, next by Jaro-Winkler distance and finally according to the Levenshtein distance. This sequence was based on the assumption that metrics based on longer sequences are more informative than the ones based on character-level. For ambiguous entities, the concept with lower information content was chosen, assuming that more general concepts have higher probability to be found on the text. The Resnik algorithm was used to compute this metric.

Submissions

Three runs were submitted to this competition:

- **Run 1:** A 2nd order CRF model was trained using the SBIEON segment representation, 100 Brown clusters and all domain lexicons available. The normalization used the pattern matching approach based on Lucene dictionaries.
- **Run 2:** Identical to the first run, but using the SBIEO segment representation.
- **Run 3:** Identical to the second run, except for the normalization module which uses a direct match approach instead of the pattern matching algorithm.

2.6.3.2 UTH_CCB

The UTH_CCB team's system achieved the top results for both the recognition and normalization tasks in SemEval 2014 [76]. This system is a result of a participation in both the ShARe/CLEFF 2014 and SemEval 2014 edition. Their recognition approach consisted on two distinct machine learning algorithms, namely a CRF and a SSVM. These two models were then combined with the MetaMap system. The outputs obtained from the MetaMap, CRF and SSVM recognition models were combined according to three distinct approaches: Machine Learning ensemble, Majority Voting Based ensemble and Direct Merging of the entity recognition results from the three models. They have explored features like bag-of-word, part-of-speech from Stanford tagger, Brown clusters inferred from Wikipedia data, random indexing and semantic categories of words based on UMLS, MetaMap and cTakes output.

This team leverages on an adapted version of the BIO classification to represent continuous entities and non-relevant entities. For non-continuous entities, four new classes were created: $D\{B,I\}$ for non-overlapping entities and $H\{B,I\}$ for overlapping entities. Their segment representation set is composed by a total of seven classes: $\{B, I, O, DB, DI, HB, HI\}$.

System	Segment Representation	Machine Learning	Main Features	Normalization
ULisboa	- SBIEON	- CRF (Stanford NER)	- Brown clusters (MIMIC II) - Domain Lexicon	- Lucene (n-Gram, Levenshtein and Jaro-Winkler) - Information Content
UTH_CCB	- BIO + D{B,I} + H{B,I}	- SSVM + CRF - MetaMap ensemble	- Brown clusters (Wikipedia) - Part of Speech	- Cosine Similarity - tf-idf (entities as single documents)

Table 2.3: State of the art systems Comparison.

For the normalization the team employed a Vector Space Model (VSM) based approach to map the most promising CUI to a given disorder mention. All disorder mentions were considered documents and therefore the cosine-similarity was used to score and rank the candidate terms. Only disorders top-ranked were considered. Non-disorders top-ranked were replaced with 'CUI-less' value.

2.6.3.3 System Comparison

In this section, a brief comparison is performed between the previously described systems. Table 2.3 resumes all the systems described presenting their primary features.

Both systems addressed the recognition of non-continuous entities with similar approaches by defining a new segment representation capable of taking these types of entities in consideration. The recognition model presented in the UTH_CCB is far more complex that the one developed by the team ULisboa, which employed less features and leverage in a single machine learning algorithm. ULisboa have on the other hand explored the unlabelled data released by the organisation, inferring Brown clusters. UTH_CCB also used Brown cluster but they were based on DBPedia information. Not using similar data for inferring the cluster may reduce the quality of the obtained results [76].

For the normalization, while ULisboa leveraged on Lucene indexes for finding the most proper candidate for each recognized entity, UTH_CCB considered each entity as a documents and addressed this task similarly to an Information Retrieval task.

2.6.3.4 DNORM

The normalization task is often associated to rule based approaches since machine learning techniques have difficulties to resolve some of the main problems inherent to this task, like for example the concept ambiguity. The DNorm system is referred by the authors as the first automated system to use pairwise learning to rank (L2R), for ranking the candidates for the normalization task [39].

The DNorm system relies on the Banner [40] system to accomplish the recognition task. For the normalization task, each candidate is represented by a set of features defined by the user and converted into the TF-IDF space. The features can be quantitative such as the string smilarity according to distinct algorithms, features of the candidate, etc.

Based on the candidate features and on the weight matrix defined on training, a score is associated to each candidate, being the candidate with the highest score the one who should be chosen.

The weight matrix will have its values assigned according to the training data used. The weight value will be the one who sets the higher score for the correct solutions on the training set, and at the same time the lower score for the wrong ones.

Results proved that the weight matrix values, which are automatically obtained through training, produce a higher performance system than simply using the cosine similarity approach, which as a fixed weight matrix, or even Lucene. Their tests used PubMed abstracts and MeSH identifiers as the knowledge base.

Chapter 3

The Proposed System

This section describes all the work I developed in order to accomplish the proposed objectives. One of the main goals is the development of a high performance system capable of recognizing and normalizing names for diseases and disorders within clinical notes.

First, an overview of the system is presented, where its architecture from a higher level of abstraction its described, followed by a detailed description of each of the system's modules and respective components, along with the datasets used by them. This system is a result of two previous iterations. The first version was developed by the team ULisboa, where I was the main developer, and it was used to participate in the SemEval 2014 edition. This participation details were described in Section 2.6.1. The second version was developed for the SemEval 2015 edition, which took place during the course of this work (October 2014 to January 2015) and it is described in this chapter. These two participations allowed me to assess the developed system performance in an international competition.

In the end of this chapter, the prototype of an interface developed for this system is presented. This last topic will be brief since the system interface was not a main objective of the work and it is still in its early stage of development.

3.1 System Overview

The system is composed by two major modules: the recognition and the normalization modules. Although these modules are responsible for addressing each task independently, in this system they are connected through a modular pipeline.

Figure 3.1 represents a higher level of abstraction of the system execution. The system initially receives as input a set of labelled and unlabelled clinical notes. These documents are split into training set and execution set. The first one is composed by all labelled documents and a set of unlabelled data used to train a recognition model and generate additional knowledge like dictionaries, clusters, etc. The second one comprises all the unlabelled clinical notes intended to be automatically annotated.

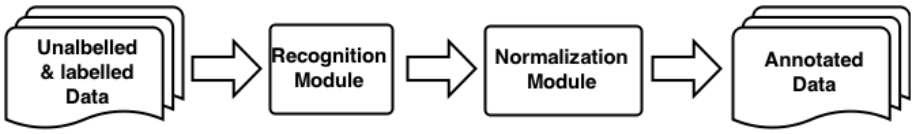

Figure 3.1: System pipeline.

For example, suppose that the system has received as input a set of unlabelled and labelled clinical notes to be used as training data, along with a single clinical note to be annotated containing the sentence *'She was diagnosed with Hepatitis B'*. The recognition module will identify the relevant entities (disease disorders) within this clinical note, identifying the entity *'Hepatitis B'*. The entity is then used as input for the normalization module which leverages on a mixture of dictionary and rule based approaches to assign a unique identifier belonging to the UMLS metathesaurus, restricted to the SNOMED CT ontology. This module's output consists on a pair composed by the entity descriptor identified within the input data (*'Hepatitis B'*) normalized with a unique identifier (*C0019163*), which together represent an annotation.

3.1.1 Pipeline

The system pipeline (Figure 3.1) was developed based on a modular architecture as it allows one to easily evaluate and optimize each system's components performance independently, and therefore understand where it is possible to improve. New strategies are possible to employ by only changing the relevant components without modifying the remaining parts of the system. This architecture was an important decision for this work, as it allowed me to gradually improve the final system and also address both recognition and normalization tasks individually.

3.2 Data Sources

This system requires a significant amount of information in the format of biomedical health records, more specifically, clinical notes. Besides this large amount of text, it also requires a controlled vocabulary to be used as knowledge base to accomplish the normalization task.

The data sources used in this work and described in this section are the ones provided for the SemEval 2015 workshop [16].

3.2.1 Ontology

For this work, the UMLS (version 2012 ab), restricted to the SNOMED CT, was the ontology chosen to be used as controlled vocabulary. The system leverages on this ontology

	Train	Development	Test
Notes	298	133	100
Words	182k	153k	109k
Disorder mentions	11,144	7,971	-
CUI-less mentions	3,357 (30%)	1,926 (24%)	-
CUI-ied mentions	7,787 (70%)	6,045 (76%)	-
Contiguous mentions	10,053 (90%)	7,345 (92%)	-
Discontiguous mentions	1,091 (10%)	626 (8%)	-

Table 3.1: Distribution of annotated data in terms of notes and disorder mentions across the training, development and test sets in SemEval 2015 [16].

to not only normalize entities but also to generate additional knowledge for the system. Further details about this ontology may be found in Sections 2.4.1.4 and 2.4.1.5.

3.2.2 Competition Datasets

Every system requires some existing knowledge in order to, based on it, generate new knowledge (recognition models, clusters, dictionaries, etc). Two types of documents can be used to generate such knowledge: labelled and unlabelled documents. The dataset used for this work comprises both types of documents, being this knowledge a requirement for any of the strategies developed for this work (e.g., dictionary based, rule based, machine learning, etc).

3.2.2.1 Labelled Notes

Labelled notes (also known as golden standard or annotated corpora) are an example of initial knowledge commonly used for both the recognition and normalization task. This knowledge consists on a set of documents that were manually annotated by domain experts. Each annotated document identifies a set of entities which are disease disorders along with their uniquely identifiers that represent each entity within a given knowledge base. These annotations were performed based on the ontology used in this work (Section 3.2.1).

Similarly to the SemEval 2014 edition, this corpora is composed by a set of clinical notes retrieved from MIMIC II database and manually annotated by domain experts, with special focus on the SemEval 2015 *Analysis of Clinical Text* task requirements. Using the information presented in the Table 3.1, and also comparing it to the information in Table 2.2, the following information about the datasets can be inferred:

- The SemEval 2015 training set comprises both the SemEval 2014 training and development sets. This corpus was used as initial knowledge for several recognition and normalization strategies.

It is composed by 298 documents, with a total of 11,144 entities, from four distinct biomedical types: discharge summaries, electrocardiogram, echo-cardiogram and radiology reports.

- The SemEval 2015 development set comprises the SemEval 2014 test set. This corpora is completely independent of the training set and it is used to evaluate the system.

  Smaller than the training set, it comprises a total of 7,971 entities within 133 documents, all discharge summaries.

- The SemEval 2015 test set is composed by new clinical notes. The size of this dataset is smaller, comprising only 100 clinical notes. Since no annotations were provided for the test dataset, only the first two subsets are used in this work for both training and evaluating the system performance locally (non-official evaluations).

The analysis of the entity distribution between the two SemEval editions shows that although the clinical notes used are the same, the annotation were revised.

Labelling Format

Two distinct labelling formats were took into consideration for this work. One was presented in the SemEval 2014 edition and the other on the SemEval 2015 edition.

- **SemEval 2014:** In this year edition, each annotated entity was represented by using the following format:

  00098-016139.text||Disease_Disorder||C0221755||1141||1148||1192||1198

  In this format, each field is separated by using two vertical bars. The first field identifies the document, the second one the entity type and the third field represents the unique identifier. The following fields are the spans which identifies the start and end character of the recognized entity. Continuous entities will have additional spans to represent all the entity portions. Since for this work only disease disorder mentions are used, the second field has always the same value. This dataset is described in the Section 2.2.

- **SemEval 2015:** For the most recent edition, a new format was imposed. The following example represents an entity in the 2015 labelling format:

  00098-016139.text|1141-1148,1192-1198|C0221755

Similar to the 2014 format, each entity is identified by the document were it is found, its unique identifier and the respective entity spans within the document. The fields are now separated by a single vertical bar. No entity type is specified on this new format as it is assumed that all entities are disease disorders. The spans are now identified using a single field where each entity portion is represented with the format *'initial_span - end_span'*. The hyphen is the delimiter used. Each non-continuous entity portion is separated by using a comma as a delimiter.

Although for this work only the 2015 SemEval corpora was used, some parsers had already been developed based on the SemEval 2014 corpora [36]. In order to use these legacy tools, and since no additional knowledge is represented in the new labelling format, a parser was developed to convert the 2015 format into the 2014 format. Thus, this is the only format that will be used in this work.

3.2.2.2 Unlabelled notes

The unlabelled corpora consists on clinical notes retrieved from the MIMIC II database. A total of 404,302 documents are used, being 431 from the annotated corpora previously described. For these documents only the raw information, without any annotations, is taken into consideration.

This knowledge is used to employ unsupervised techniques such as clustering. This dataset is considerably larger than the annotated data, since no manual intervention is required.

3.3 Recognition

The first task addressed in this work consists on the recognition of relevant named entities within narrative biomedical text, more precisely disease disorder mentions within clinical notes. In the previous chapter, several state of the art approaches were described, each one comprising its advantages and disadvantages. Two of these approaches were employed for the recognition module: dictionary based and machine learning.

3.3.1 Dictionary Based

Humans, in order to manually recognize named entities within narrative text, normally follow an approach similar to a dictionary based lookup. First a physical or conceptual representation of a dictionary is created with the entities intended to be recognized, and then a direct match is performed between those entities and the ones within the text. Naturally, additional process are executed by humans, sometimes unconsciously, in order to improve the results.

Similarly, the first approach developed for this work consisted on the employment of a dictionary based algorithm. This approach is known for its simplicity and reasonable results although far from the ones possible to achieve by employing more advanced techniques. This approach was developed with the intention to be used as baseline standard, allowing to compare and understand the impact that more complex approaches have on the results of this specific task.

3.3.1.1 Corpora

The training data (Section 3.2.2.1) was the only source of information used for this strategy. The entities within the training set can be separated into two distinct categories: continuous and non-continuous entities. The first category comprises all the entities that are defined with one or more continuous tokens, and the second one all the entities that have at least one token disjoint from the remaining tokens of the entity (see examples in Section 2.2.1.1). One out of ten entities within the training set are non-continuous. Although this number may seem small, it is significant due to the limited amount of data.

3.3.1.2 Method

A dictionary based approach leverages on the existence of a dictionary containing the named entities intended to be recognized. Therefore, a dictionary must be generated based on the clinical notes belonging to the training set.

For each entity type (category) identified, I developed a specific matching algorithm:

Continuous Entities

These entities can be looked up in the input text by using a simple direct string match approach. Therefore, the dictionary only needs to store the entity itself.

For example, the spans *10||18* within a given document with the sentence *'I have an headache'*, identify the named entity *headache*, more precisely the initial and final character of the entity within that given document. Every reference that is a direct match to this named entity will be considered a relevant entity.

Single character entities are considered a special case. They do not have any intrinsic information to allow a proper direct match solution, being completely dependent of the context. For that reason these entities are preferably droped off, as they induce a lot of false positives.

Non-continuous Entities

These entities are far more complex than continuous ones, as they have irrelevant words between the begin and end of the entity itself. For that, the correct identification of these entities presents a challenge. To address these entities three specific approaches were developed.

- Direct match as a whole: This approach converts these entities into continuous ones, by considering only the first and last entity span. An entity is then composed by both the relevant and irrelevant portions, being The dictionary composed by these new continuous entities.

  For example, the spans *4||11||21||29* which identify the non-continuous entity *'abdomen notender'* within the sentence *'The abdomen is soft, notender'*, will be represented in the dictionary as *'abdomen is soft, notender'*, containing all the information between the first and last span presented.

- Relevant direct match only: This approach consists on also converting these entities into continuous ones, but unlike the previous approach, it only takes into consideration the relevant portions of the entity.

  Leveraging on the previous example, the spans *4||11||21||29* will be represented in the dictionary as a continuous entity *'abdomen nontender'*.

- Relevant direct match with irrelevant window: This approach is based on the definition of an *'irrelevant window'*, which consist on the maximum number of characters that are acceptable to exist between two relevant portions of an entity. With this approach, the dictionary needs to comprise all the relevant portions of an entity, and verify if they are found in the text within an irrelevant window apart.

  Considering the same example where the spans *4||11||21||29* represent an entity within the sentence *'The abdomen is soft, notender'*. Using this approach with an irrelevant window of size 15 would result on a positive match since the irrelevant portion *' is soft, '* has less than 10 characters (spaces included). If the window used was only 5 characters, no match would be performed.

In any of the previous algorithms, the dictionary is composed by case sensitive entities as no post-processing techniques were applied.

3.3.2 Machine Learning

The dictionary baseline approach leverages entirely on the quality of the dictionary, and even with a comprehensive dictionary the context is never taken into consideration, resulting in performance losses. A more complex and common approach for the recognition of named entities consists on the employment of machine learning algorithms.

Several tools are available for the implementation of a machine learning approach. The Stanford NER software [17] is the one used to generate CRF models [35] in this work. I decided to use this software as it has proven to achieve promising results in the recognition of general domain named entities, and because it is open source, allowing me to extend it. This last feature was a relevant requirement for this work, as a new segment representation needed to be implemented.

Since this approach is based on a learning algorithm, two distinct procedures are available: training and execution. The first one consists on using the input labelled data to create a recognition model, while the second one consists on using the respective model to automatically identify relevant entities within the unlabelled data.

The following subsections describe the corpora used in this module as well as the training and execution procedure.

3.3.2.1 Corpora

Machine learning algorithms are feature base approaches, and because of that, besides the commonly used annotated data to generate CRF models, additional knowledge sources can be used to generate new features and thus improve the CRF model performance.

This module leverages on the data sources described in the Section 3.2, more precisely the SNOMED CT ontology (Section 3.2.1) and the labelled and unlabelled notes (Section 3.2.2). Some external knowledge was also used to generate domain lexicons.

Domain lexicons consist on a set of documents, where each document comprises several named entities related to each other and from the same domain. For this module, The following individual domain lexicons from the four identified knowledge sources have been generated.

- SNOMED CT
  - All SNOMED CT named entities
  - Separated SNOMED CT named entities
    - Disease disorder
    - Non-disease disorder
- Golden Corpora (Labelled Clinical Notes)
  - All golden named entities
- DBPedia
  - Drugs named entities
  - Disease disorder entities
- OBO
  - Cardiovascular diseases entities
  - Infectious diseases entities
  - Human diseases entities
  - Symptoms named entities

Figure 3.2: Supervised training process.

3.3.2.2 Training

The training process consists on five steps represented in the Figure 3.2:

Input (Annotated Data)

The input consists on a set of annotated clinical notes which must be representative for the generation of a higher performance model.

Pre-Processing

It is the first recognition module component. It is responsible to process the clinical notes used as input. This component commonly comprises several techniques available to prepare the input data for the classification component, being the tokenization technique the only one used in this work.

The tokenization algorithm follows a rule based approach in order to transform the input text into tokens. These rules are generated by leveraging on a specific set of delimiters (single characters) abstracted into the following groups:

- Letters: It comprises all characters that are used to form words.
- Digits: All the characters that are numbers. Together with the previous group the alpha-numeric character dictionary is represented.
- Space: Special character that has a group for itself. It is the most commonly used delimiter in natural language.
- Special characters: Characters that are represented by using an escape sequence. For example the new line and tab character.
- Non alpha-numeric characters: All the remaining characters that are not comprised by any of the previous groups. For example brackets, punctuation marks, etc.

Based on these delimiter groups, the following rules were generated by taking into consideration both the current delimiter and the previous one:

- Delimiter ignored: A delimiter must be ignored when the current character is a delimiter space or belongs to the special character group, and it is preceded by other space or special character. For example, a new line character followed by

a tab character. When this rule is activated, only one of the delimiter characters must be taken into consideration, since the additional characters do not provide any relevant information.

- Basic delimiter: The space character and the characters belonging to the special character group are considered basic delimiters. However, this rule is only employed if the previous rule (delimiter ignored) is not applicable. With this rule, both sentences and words are split into tokens.

  For example, the sentence *'The patient is 40-years-old'*, is converted into four tokens: *'The'*, *'Patient'*, *'is'* and *'40-years-old'*.

- Extended delimiter: When the current character belongs to the non-alpha-numeric character group. Leveraging only on the previous rule, the token '40-years-old' is generated, which is incorrect. Being the *hyphen* a non-alpha-numeric character, this last token is split into five distinct tokens: '40', '-', 'years', '-' and 'old'.

- Non-delimiters: All characters which belong to the alpha-numeric group are considered textual information and thus are ignored. These characters when grouped form a token.

This pre-processing component is the same as the one used in the SemEval 2014 [36] and SemEval 2015 [37] participations.

Classification

The classification component is only required for the training procedure. It receives as input the text pre-processed and structured into tokens by the previous component, and assigns to each token a single class from the SBIEON classification model. The SBIEON segment representation was created in [36] and, as previously described in Section 2.2.2.2, it allows one to represent non-continuous entities, and thus, address one of the major challenges in this task. The classification process follows the information present in the annotated data. Some classification examples are provided in the same section.

Supervised Learning/Feature Generation

This component is responsible for the generation of a CRF recognition model. The model is generated through a supervised learning process which leverages on the information associated to each token in the format of features. Each token is represented by a vector of features.

In the previous component, each token had assigned a class from the SBIEON classification which by itself adds an important semantic value, making it processable by the machine learning algorithm. However, this information is not sufficient to represent the

token completely. Therefore, additional features are generated for each token, increasing their amount of processable and representative information. This knowledge, when included in the recognition model, allows one to better represent each token and thus improve the overall learning process.

Token features leverage on the token itself (target token) and on the previous and following tokens within a certain window. All the tokens within this window represent the current context. The main features used to generate the recognition model are the following:

- The token
- The SBIEON class assigned to the token
- The position of the token within the sentence
- The n-Grams of the token (see Section 2.2.2.3 for more detail)
- The lemma of the token (see Section 2.2.2.1 for more detail)
- The token shape, which is defined by a set of features associated to the current token. For example, capital letters, the existence of numbers, etc.
- Domain lexicons (see Section 3.3.2.1 for more detail). This feature consists on indicating to what domain lexicon, if any, a given token belongs to.
- Clustering. A technique which allows to generate a feature indicating to which cluster the token belongs to. Brown clusters was the clustering technique chosen and implemented by using the source code provided by Turian [68]. (see Section 2.2.2.2, *Unsupervised Algorithms* for more detail)

Each of these features are assigned to each token individually, but they are also used to infer more complex features by taking into consideration the context where they are inserted. For example, with a window/context of size 2, the individual features from the two previous and the two next tokens are combined with the current token features. The resulting features are assigned to the current token. In this way, each token is represent not only by its individual features but also by a set of features which represent the context where the token is inserted. The window size value represents the model's order.

Output (Recognition Module)

When the training process is finished, a CRF recognition model is generated containing the knowledge learned. This model is ready to be used in the execution procedure.

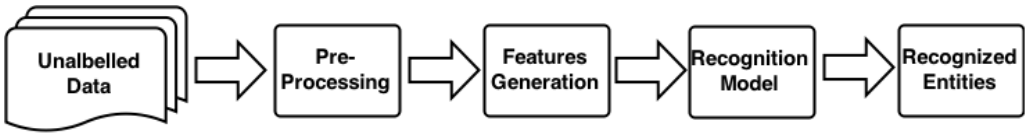

Figure 3.3: Supervised execution process.

3.3.2.3 Execution

The execution process architecture is similar to the one followed by the training procedure, being most of the components shared. The process is represented in Figure 3.3, and it consists on five stages:

Input (Unlabelled Data)

The input consists on unlabelled clinical notes intended to be automatically annotated.

Pre-Processing

The unlabelled notes received as input are pre-processed in the same way that was described in the training process (see Section 3.3.2.2). This component is thus identical to the pre-processing component using during the training procedure.

Feature Generation

Although there are no annotations available to define the right classes to assign to each token, a set of features are still generated for each token in order to represent it during the model execution. Based on these features, the recognition model will be able to assign the most appropriate class according to the learning process.

The feature generation process is identical to the one described in the training process (see Section 3.3.2.2).

Recognition Model

The CRF recognition model generated in the training process is used to assign the most probable class to each token. The tokens with the classes associated to entities are the ones considered relevant. Since the model uses the SBIEON model, the relevant tokens are the ones with the classes 'S', 'B', 'I' and 'E'.

Output (Recognized Entities)

The output of the execution procedure consists on the relevant disease disorder named entities identified within the given clinical notes. The entities are represented by using the labelling format described in Section 3.2.2.1.

3.4 Normalization

The system's second major module is responsible for the normalization of the disease and disorder named entities recognized by the recognition module. A named entity received as input does not have any semantic meaning associated to it and, thus, it needs to be normalized with a UMLS unique identifier (CUI), if one is available. Otherwise, the special identifier *CUI-less* must be assigned. This assignment represents a major challenge for the system, as it must be capable of successfully disambiguating ambiguous entities and hence, assign the correct unique identifier. A normalized entity is referred to as a concept as it now has a semantic meaning associated. Therefore, a concept is nothing more than an entity (concept descriptor) with a CUI associated.

3.4.1 Architecture

This module follows a fully modular pipeline architecture, being composed by six components, each one developed independently. This allows one to easily replace a component for another one using a distinct strategy, without changing the pipeline's execution. Each component addresses a set of named entities by leveraging on a specific source of knowledge.

Figure 3.4 illustrates this module's pipeline. Two major strategies are used: dictionary lookup and similarity search. The first strategy is the simplest and it consists on looking up the input entity in a set of pre-defined dictionaries. The second one allows one to address entities which are not comprised in any dictionary by leveraging on pattern matching techniques. Additional and distinct techniques are used by some of these components to addressed ambiguous entities.

The pipeline receives as input named entities (previously recognized), and outputs the same entities now with a unique identifier associated.

3.4.2 Data Sources

Although this module does not follow a machine learning approach, it still requires annotated data to generate knowledge for the system. Two types of knowledge are used by this module: dictionaries and an ontology.

3.4.2.1 Ontology

SNOMED CT is the controlled vocabulary used by the normalization module. More details are provided in the system overview (Section 3.2.1).

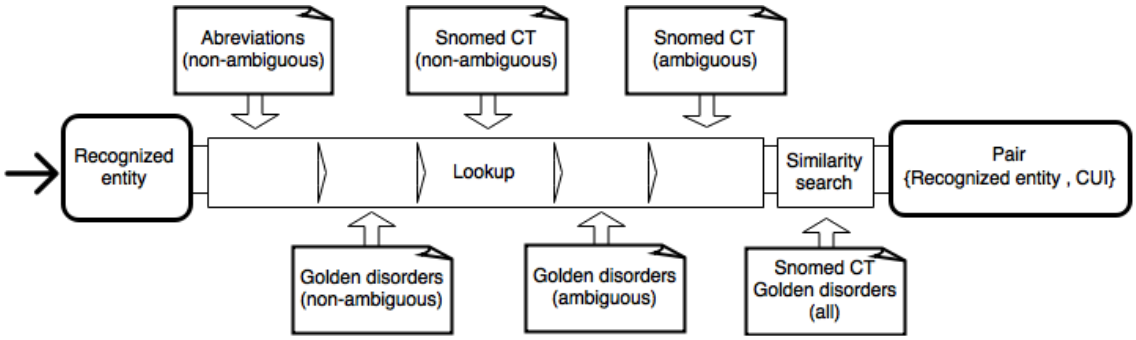

Figure 3.4: Normalization pipeline.

3.4.2.2 Dictionaries

Dictionaries are an essential part of this module. A set of dictionaries containing concepts (an entity and the respective CUI) were generated from two sources of information, labelled clinical notes and the SNOMED CT ontology. Each dictionary took also into consideration the type of information contained in those sources, more precisely if the entity is ambiguous or non-ambiguous. An entity is considered ambiguous when for the same descriptor, more than one CUI may be assigned.

A total of six dictionaries are generated from the mentioned data sources, and used in this module as illustrated in Figure 3.4:

Labelled Clinical Notes

A dictionary is generated containing all the relevant entities comprised in these clinical notes, along with their respective unique identifiers. This dictionary is used based on the assumption that previously annotated entities are most likely to be normalized with the same unique identifier. The retrieved entities allows one to expand the amount of known relevant entities, as the clinical notes may contain abbreviations or synonyms not contemplated on the controlled vocabulary SNOMED CT.

From this source of information, three distinct dictionaries are created

- Abbreviation dictionary: Contains all concepts whose descriptor is up to 5 capital letters and is non-ambiguous. This descriptor pattern often represents abbreviations. For example, the concept *ASD* with the identifier *C0018817* is an abbreviation for *arterial septal defect*.

- Golden dictionary (ambiguous): Consists on all concepts that do not belong to the abbreviation dictionary and whose descriptor is **ambiguous**.

- Golden dictionary (non-ambiguous): Consists on the remaining entities and the respective CUI that do not belong to neither one of the previous dictionaries i.e., that are **non-ambiguous**.

SNOMED CT Ontology

Each concept within an ontology is identified by its uniquely identifier (CUI) which is associated to a set of descriptors (entities) and to a unique semantic type. Although for this work only the annotation of disease disorders is intended, the following dictionaries were generated without using any semantic type filter. This approach implies the application of a filter during the pipeline execution.

- SNOMED dictionary (ambiguous): Contains all SNOMED CT entities that are **ambiguous** along with the respective CUI.
- SNOMED dictionary (non-ambiguous): Contains all SNOMED CT entities that are **non-ambiguous** along with the respective CUI.

The entities retrieved from this ontology, and used to generate the mentioned dictionaries, require the application of a post-processing algorithm to improve the data quality and standardize the description format. The following post-processing rules were employed:

- Removal of quotation marks from the entities. For example: *"Glue ear"* would be filtered to *'Glue ear'*.
- Truncation of entities started by the characters '[D]', '[X]' or '[M]'. For example: *'[D]Paraesthesia'* would be truncated to *'Paraesthesia'*.
- Removal of entities which contain the suffix '-RETIRED-', since this "tag" represents entities that are not used anymore in this ontology version. For example: *'Paromomycin -RETIRED-'* would be removed.
- Truncation of entities with the suffix "NOS". For example: *'Parotid gland, NOS'* would be truncated to *Parotid gland*.
- Truncation of entities with the semantic type suffix. For example: *'Parotitis (disorder)'* would be truncated to *'Parotitis'*.
- All duplicated dictionary entries are removed in order to improve the system performance.

Comprehensive Dictionary This dictionary is generated based on both sources of information and it comprises all the dictionaries aforementioned.

In the aforementioned dictionaries, ambiguous entities are all entities that may have more than one possible CUI associated in the specific data source. For example, the Golden dictionary (non-ambiguous), contains all entities which are non-ambiguous on the labelled clinical notes data source.

3.4.3 Method

In this section a description of the pipeline is presented, describing each component's strategy to addresses the affected entities. This section also describes the novel pattern matching procedure that was developed, as well as the strategies employed to resolve the two major normalization challenges: the entity disambiguation and the assignment of the special identifier *CUI-less*.

3.4.3.1 ExtendedLevenshtein Distance

For this work, I developed the ExtendedLevenshtein distance, a novel pattern matching algorithm. It is based on a best-match approach which measures the similarity between a *target* descriptor (recognized entity) and a *candidate* descriptor (dictionary entity), regardless of their token's orders.

This measure is calculated as follows:

1. Both *target* (d_t) and *candidate* (d_c) descriptors are split into tokens (S_{dt} and S_{dc}) using the $\mathrm{SplitTokens}$ function.

$$S_{dt} = \mathrm{SplitTokens}(d_t)$$
$$S_{dc} = \mathrm{SplitTokens}(d_c)$$

2. For each *target* token in S_{dt}, the Levenshtein distance is computed against all *candidate* tokens within S_{dc}. The target token with the minimum value (distance) is the one chosen, being this value associated to it. The candidate token in S_{dc} which minimizes the Levenshtein distance is removed from the set for the following iterations. Each candidate token can only be matched to a unique target token

3. The ExtendedLevenshtein distance is computed by summing the distances associated to each target token in S_{dt}. This value is normalized using the number of tokens that compose the target entity, $|S_{dc}|$. This distance is represented by the following formula:

$$\mathrm{Sim}(d_t, d_c) = \begin{cases} -1, & \text{if} |S_{dt}| > |S_{dc}| \\ \dfrac{\sum\limits_{w_{dt} \in S_{dt}} \mathrm{BestMatch}(w_{dt}, S_{dc})}{|S_{dt}|}, & \text{otherwise} \end{cases}$$

In the formula, we have that

$$\mathrm{BestMatch}(w_{dt}, S_{dc}) = \mathrm{Min}\{\mathrm{LevDist}(w_{dt}, w_{dc}) : w_{dc} \in S_dc\}$$

where $\mathrm{BestMatch}$ returns the minimum Levenshtein distance between the token w_{dt} and all available tokens in S_{dc}.

For example, let *abdominal pain* be the target entity and *pain in the abdominal* one of the candidates. Using the Levenshtein distance, these two entities would only be considered 23.8% similar. With the ExtendedLevenshtein distance, these entities are considered 100% similar, as they share the two target tokens. If for example, the target entity was *abdominal painful* instead, then the Extendedlevenshteindistance would consider these entities 75% similar, versus the 19% resulting from the regular Levenshtein distance.

3.4.3.2 Disambiguation techniques

Being able to correctly assign a unique identifier to a given entity represents a great challenge. For example, the entity *Pressure* can represent the concept of doing pressure on something or feeling pressure about a given occasion. Both have the same descriptor but distinct CUIs.

The following techniques were employed to resolve ambiguous cases:

- **Information Content**: Technique described in the Section 2.2.2.4. A software solution capable to calculate this measure by using the SNOMED CT ontology as knowledge base was developed. Third party software packages like UMLS::Similarity [46] could have been used, but this solution lacked in performance and uses a distinct version of the SNOMED CT, resulting in incorrect measures. For these reasons, I developed a higher performance solution which supports any of the information content measures previously described.

- **Semantic Similarity**: Technique described in the Section 2.2.2.5. A software solution capable of calculating the semantic similarity measure between two given concepts was also developed. By developing the software, I was able to choose the SNOMED CT version supported and also optimize the implementation. This software supports all semantic similarity measures described in the mentioned section.

- **CUI Frequency**: It consists on the probability of a given CUI being assigned to a given unique identifier within a set of documents. For example, the entity *Pressure* within the annotated data has the identifier *C0038435* assigned 20 times while the identifier *CUI-less* is only 3 times assigned. It assumes that CUIs more frequently assigned on past cases are more likely to be assigned on future cases.

3.4.3.3 CUI-Less Assignment

Since the identifiers (CUIs) within the dictionaries generated based on the SNOMED CT data source were not validated, an additional procedure must be employed to certify the validity of this identifiers. Therefore, when a CUI from one of these dictionaries is chosen to be assigned to a given recognized entity, the CUI semantic type is first validated. If the

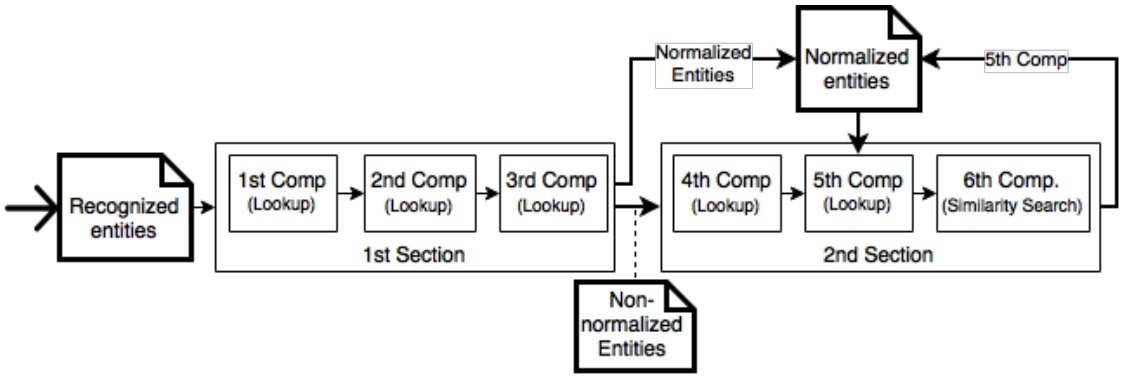

Figure 3.5: Normalization module flow execution.

identifier does not belong to a *disease disorder* semantic type, then the *CUI-less* identifier is assigned.

This validation could have been performed during the dictionary generation. However, if in the future new semantic types are intended to be recognized, the dictionaries would have to be re-generated. By decoupling this validation, only this part needs to be updated to support the new requirements.

3.4.3.4 Pipeline

Following the flow illustrated in Figure 3.5, all the non-ambiguous entities (first section) are first normalized, resulting on two distinct sets: one with the already normalized entities, and another with the remaining non-normalized entities. This last set is then used as input for the second section (ambiguous and non-dictionary entities), where they will be normalized. The set with the already normalized entities is used as knowledge by the fifth component for applying a semantic similarity approach.

The pipeline's components are organized so that the entities with higher confidence are normalized by the first components leaving the most complex entities, and thus with lower confidence, for last. The level of confidence was defined according to the following rules:

- Non-ambiguous entities have **higher** confidence than the ambiguous entities, since the former have only one candidate CUI, allowing a direct association.
- Concepts retrieved from the dictionaries generated based on the labelled clinical notes have **higher** confidence than the concepts retrieved from the dictionaries generated based on the SNOMED CT ontology. This rule is valid for two reasons: First, the concept descriptors within the SNOMED CT, although post-processed, may still contain some irregularities which affects the performance. The second reason is based on the assumption that the CUI assigned to a given entity by a domain expert is likely to be assigned to the same entity in future clinical notes.

Based on the flow above described, a detailed description of each component's strategy is presented.

Abbreviation Dictionary Lookup (1st Component)
This component uses the abbreviation dictionary as data source. The non-ambiguity along with its specificity makes this dictionary the one with higher confidence.

This component uses a dictionary lookup approach, which consists on simply assigning the identifier associated to the direct matched concept. Suppose that the recognized entity *ASD* matches an Abbreviation dictionary entry which has the identifier *C0018817* associated. Therefore, this will be the CUI assigned to the *ASD* entity. If the entry's CUI was the special identifier *CUI-less*, then this would be the assigned identifier.

Golden Dictionary Lookup (2nd Component)
This component is identical to the previous one with the exception of the source knowledge which consists on the Golden dictionary (non-ambiguous).

These two components were not merged as the data sources used have distinct levels of confidence. By splitting in these two components, it is possible to infer that entities normalized by the first component have a higher level of confidence (due to its specificity) than the ones normalized by this component.

SNOMED Dictionary Lookup (3rd Component)
This is the last component responsible for addressing non-ambiguous entities. It leverages on a dictionary lookup approach identical to the one employed by the previous components and uses the SNOMED dictionary (non-ambiguous) as data source.

Since this component leverages on a dictionary generated based on the SNOMED CT data source, the validation described in the Section 3.4.3.3 is employed to validate if the chosen CUI is in fact a valid choice. If not, then it is assumed that no valid CUI is available in the controlled vocabulary to represent this entity, and the *CUI-less* identifier is assigned.

Ambiguous Golden Entities (4th Component)
This component leverages on the Golden dictionary (ambiguous) as data source, and it is the first component responsible for addressing ambiguous entities. This component first employs a dictionary lookup to retrieve all unique identifiers possible to be assigned to the recognized entity. Each of these possible identifiers are evaluated by using a measure linearly composed by both the frequency and the information content of each candidate. The identifier who maximizes this measure is the one chosen to be assigned to the entity.

This component's algorithm is described as follows. Let E be an entity that was recognized by the recognition module and that belongs to the Golden dictionary (ambiguous):

1. All CUI candidates associated to the entity *E* are retrieved from the dictionary by using a lookup approach.

2. The frequency of each candidate CUI is calculated by summing the number of times each candidate CUI was assigned to the entity *E*, within the labelled notes.

3. For each candidate CUI, the information content is calculated by leveraging on Seco's algorithm [63]. If the CUI candidate consists on the *CUI-less* identifier, a fixed information content value must be defined. In this work, the lowest information content value was assigned. The *CUI-less* identifier does not belong to the SNOMED CT ontology and thus, it does not have any information associated.

4. These two individual measures are linearly combined for each CUI candidate. This new measure represents the likelihood of that CUI to be the correct one, comprising both the knowledge retrieved from the ontology and from the labelled corpora.

5. The CUI candidate with the highest measure value is the one chosen to be assigned to the entity. Since *CUI-less* identifiers are already taken into consideration, no additional processing is required to decide if the chosen identifier is valid.

Ambiguous SNOMED Entities (5th Component)
This component leverages on the SNOMED dictionary (ambiguous) as data source. Identical to the previous component, a dictionary lookup approach is employed to retrieve the candidate CUIs for a given entity. Unlike the previous component, this dictionary was generated based on the SNOMED CT ontology, which means that no labelled notes are available to employ the same disambiguation approach taken by the previous component.

The candidate CUIs are disambiguated by employing a semantic similarity approach. With this approach, each candidate is semantically compared to all previously normalized entities within the same clinical note. The candidate CUI with the highest semantic similarity will be the chosen one. This approach assumes that concepts within the same clinical note must share some semantics.

For example, suppose that a couple of entities within a given clinical note were already normalized, and that those concepts are semantically related to the patient anxiety. Based on this knowledge built during the module's execution, if the entity *Pressure* is intended to be recognized, then the CUI which is more related to the previous concepts is chosen. In this case, the entity would be normalized with the CUI that represents *Pressure* as a state of spirit, and never as a mechanical process.

This approach however has some challenges:

- **Special identifiers**: The previous normalized entities may have been assigned with the *CUI-less* identifier. These normalized entities are ignored as no valuable knowledge can be retrieved from them.

- **Amount and quality of normalized entities**: Since this approach leverages on the knowledge generated by the system execution it is possible, based on the recognized entities' order used as input, that no entity was yet normalized in a given clinical note. To mitigate this challenge, all non-ambiguous entities are first normalized (first section), and only then the second section including this component will address the affected entities. Entities addressed by this component are also added to this knowledge, increasing its size.

  If the entities within a given clinical note are wrongly normalized, then this component's performance will be affected. The prioritization of the components which address non-ambiguous entities ensures that normalized entities used as knowledge have a certain confidence. The addition of entities normalized by this component follows the assumption that the normalized entities have a reasonable level of confidence, as they are highly similar to the entities already within the knowledge.

This component's algorithm is as follows: Let *E* be an entity that was recognized by the recognition module and that belongs to the Golden dictionary (ambiguous). Let also *NK* (normalization knowledge) be the set of entities which were previously normalized.

1. All candidate CUIs associated to the entity *E* are retrieved from the dictionary by using a dictionary lookup approach. Each candidate is represented as *C*.

2. The semantic similarity between each candidate *C* and each concept within the normalization knowledge *NK* is calculated. Note that only concepts that belong to the same clinical note are used. The Resnik [58] measure is employed to calculate the semantic similarity value.

3. Each candidate *C* will have assigned the highest similarity value calculated between himself and all concepts within the *NK* knowledge set.

4. If the candidate *C* with the highest semantic similarity is considered valid (see Section 3.4.3.3), then it is assigned to the recognized entity. Otherwise, the special identifier *CUI-less* is assigned.

Similarity Search (6th Component)

This last pipeline component addresses all the entities which are not comprised in any of the available dictionaries. It leverages on a comprehensive dictionary to retrieve the most similar entity using a pattern matching approach. ExtendedLevenshtein (Section 3.4.3.1), Levenshtein and NGram5 (2.2.2.3) are the similarity algorithms that are used. The retrieved entity is then normalized by leveraging on one of the previous component's strategy.

Let *E* be an entity that was recognized by the recognition module and does not belong to any of the generated dictionaries.

1. A dictionary *D* is generated by composing all the dictionaries generated for the previous components. This dictionary is indexed according to both Levenshtein and ExtendedLevenshtein distance, by using the Lucene software [44]. The indexing allows one to retrieve the *n* most similar entities according to these measures.

2. The *n* most similar entities according to the indexes are retrieved. Let *S* (suggestions) represent this set of entities.

   Only suggested entities with a distance higher than a given *delta* were taken into consideration to avoid false positives. For this system, I chose a *delta* of 0.5 and a limit of 300 suggestions per index [37].

3. If no suggestion is available, then the special identifier *CUI-less* is assigned to the entity *E*.

4. Each entity within the suggestions set *S*, will be evaluated according to a composed formula that takes in consideration the Levenshtein (fine granularity), ExtendedLevenshtein (token's order) and NGram of size 5 (higher granularity).

$$\begin{aligned}\mathrm{Sim}(d_c, d_t) = 0.15 * \mathrm{Lev}(d_c, d_t) + 0.15 * \mathrm{NGram5}(d_c, d_t) \\ + 0.7 * \mathrm{LevExt}(d_c, d_t)\end{aligned}$$

   In the formula, Sim represents the similarity between the target d_t and candidate d_c descriptor. Lev, $\mathrm{NGram5}$ and LevExt represent the Levenshtein, NGram of size 5 and ExtendedLevenshtein distance, respectively. This formula was composed according to a separate empirical evaluation described in Section 4.4.

5. The entity with the highest similarity value according to the previous formula is the one chosen. This entity belongs to one of the available dictionaries, meaning that one of the previous components is capable of normalizing it. For example, if the retrieved entity belongs to the SNOMED dictionary (non-ambiguous), then the 3rd component's approach is employed.

3.5 Previous Iterations

The system described in this chapter results from the improvement of the two versions that were used to participate in SemEval 2014 and 2015. In this section, these systems will be briefly described and compared.

3.5.1 SemEval 2014 Workshop

Since this competition took place before this thesis has started (April 2014), a detailed description of the competition and the system submitted was provided in the state of the

art, more precisely in the Section 2.6.1. I was one of the main developers responsible for the development of this system, which allowed me to use it as starting point for this work.

3.5.2 SemEval 2015 Workshop

SemEval 2015 edition [16] is very similar to the previous year's edition (SemEval 2014) described in Section 2.6.1. They both held an *Analysis of Clinical Text* task where the participants where challenged to recognize and normalize disease disorders within clinical notes. For this competition, 16 teams participated with a total of 38 submissions.

3.5.2.1 Resources

The data sources released for the SemEval 2015 competition were also used by the system presented in this thesis. A detailed description was provided in Section 3.2.

3.5.2.2 Assessment

In this year's edition, the recognition and normalization tasks were evaluated as a unique task. Therefore, the systems are only evaluated based on their precision, recall and F-score, being also used both strict and relaxed evaluation as follows:

- Strict: The recognized spans must be a *perfect match* with the golden standard spans and the normalization must be well succeeded.
- Relaxed: The recognized spans must *overlap* the golden standard spans and the normalization must be well succeeded.

With this evaluation, the normalization performance will dictate the system's overall performance. Even with a perfect recognition algorithm, the system may achieve poor results if a poor normalization algorithm is employed.

3.5.2.3 System

I leveraged on the existing SemEval 2014 system, aiming to create a much higher performance system capable to match the top performancing systems in 2015 edition.

This system followed the same strategy used in SemEval 2014 to accomplish the recognition task, being the normalization module the major improvement. I created a pipeline normalization framework, which was the first iteration to achieve the final pipeline described in Section 3.4.3.4. This pipeline is illustrated in Figure 3.6, where four distinct components are distinguished: Abbreviations lookup, golden disorders lookup, SNOMED lookup and finally similarity search leveraging on the SNOMED CT controlled vocabulary

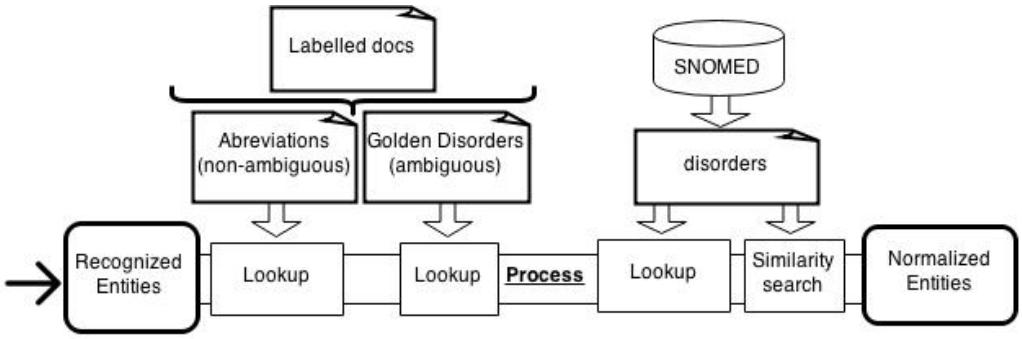

Figure 3.6: SemEval 2015 normalization pipeline.

The first three components of the pipeline employ a dictionary lookup approach. If no match is found, then the most similar entity within the SNOMED CT ontology is retrieved and normalized. This entity is retrieved by employing the same techniques showed in Section 3.4.3.4, more precisely by the sixth system's component.

Ambiguous entities, in all components, are disambiguated by choosing the entity with the lowest information content (according to Shanchez [61]). This assumes that more generic entities are more likely to appear on a text.

3.5.3 Submissions

Three runs were submitted to this competition. All runs used the same normalization strategy, being the recognition module the only variation. The runs were as follows:

1st Run: This run relied on 2nd order CRF model using the SBIEON segment representation and a rich set of features which include 100 Brown clusters and a set of domain lexicons, namely: SNOMED CT disorders, drugs and diseases from DBPedia and a list of disorders retrieved from the annotated data. The model only used the clinical notes within the training dataset for training.

2nd Run: This run is identical to the first one, with the exception of the domain lexicon, which was not included.

3rd Rum: Again, Run 1 was used as baseline. The only differences consisted on the amount of clinical notes used to train the model, which now included the development dataset, increasing by 133 the number of clinical notes used.

3.6 System Comparison

A brief comparison is performed in Table 3.2. This table allows one to understand the evolution of the system and the techniques that were employed for both the recognition and normalization tasks. Only each system's best submission is presented.

	SemEval 2014	SemEval 2015	Final System
Recognition Module			
- Segment Representation	SBIEON	SBIEON	SBIEON
- Recognition Model	CRF 2nd order	CRF 2nd order	CRF 1st order
. Software	Stanford NER	Stanford NER	Stanford NER
. Brown clusters	100 clusters	100 clusters	200 clusters
. SNOMED CT Lexicon	Yes	Yes	No
. DBPedia Lexicon	Yes	Yes	No
. Golden Lexicon	Yes	No	No
Normalization Module			
- Architecture	N/A	Pipeline (4 components)	Pipeline (6 components)
-Techniques			
. Lucene Indexing	Yes	Yes	Yes
. Levenshtein	Yes	Yes	Yes
. NGram	Yes (Size 2)	Yes (Size 5)	Yes (Size 5)
. Jaro-Winkler	Yes	No	No
. ExtendedLevenshtein	No	Yes	Yes
- Disambiguation			
. Information Content	Yes (Seco, lowest IC)	Yes (Seco, lowest IC)	Yes (Seco, auxiliar)
. Semantic Similarity	No	No	Yes (resnik)
. Annotation Frequency	No	No	Yes (along with the IC)

Table 3.2: Comparison of the system versions described in this work.

The information provided by this table, shows that the recognition approach used by the three systems did not suffer considerable changes, relying on the Stanford NER software and on similar features. As for the normalization approach, the two last system versions leverage on a similar pipeline architecture. This new strategy is considered an improvement when compared to the simply rule based approach used by the SemEval 2014 system.

3.7 User Interface

Together with the SemEval 2014 team, an initial prototype of a user interface for this system was developed. We intended to create an interface that would allow users to annotate disease disorders within clinical texts. Besides this main requirement, the interface was also intended to provide additional features to a complete Web application, such as supporting crowd sourcing techniques and document retrieval from distinct providers.

This chapter is intended to provide an overall vision of the system, including the features implemented and the technologies used.

3.7.1 Overview

Figure 3.7 presents the main interface of the web system, which is composed by three panes:

- Left Pane: Contains the input forms for the user to indicate the document type to annotate along with the respective identifier. This pane also displays the document retrieved from the chosen provider and annotated by our system.

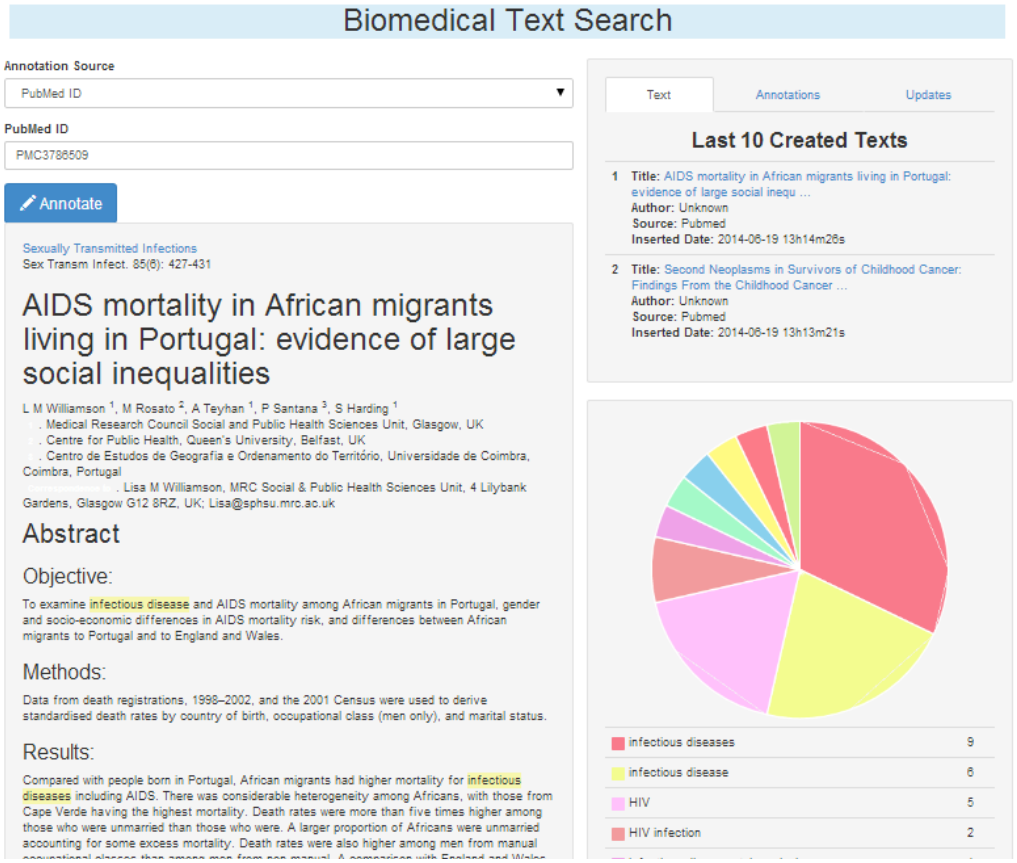

Figure 3.7: Overview of the web-system prototype.

- Top Right Pane: Contains three tabs showing real time information regarding the users' activity. Notifications for the last added documents and the last created and modified annotations are provided.

- Bottom Right Pane: Contains a circular graph along with a list showing the most frequent concepts presented in the current document.

The annotated entities are highlighted within the documents for an easy access to the information. As Figure 3.8 illustrates, users are able to check the unique identifier assigned to each of the annotated entities and also correct this information manually by overwriting the initial information.

Figure 3.7 illustrates a use case where a user introduces a unique PubMed identifier to retrieve the respective abstract. Figure 3.8 on the other hand, illustrates the same use case, but now retrieving a specific tweet.

3.7.2 Architecture

The Web application follows the common architecture composed by a front-end, back-end and Web services.

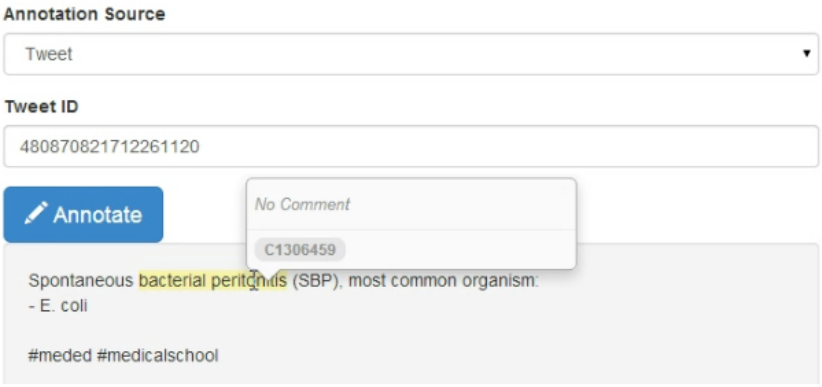

Figure 3.8: Tweet retrieval user interface.

Back-end is the component responsible for storing and processing the retrieved documents. This component incorporates the system described in this work. This component also comprises the needed interfaces to consume the Twitter and PubMed APIs. These APIs were used to retrieve documents based on their unique identifier.

Web services consists on an abstraction layer which allows the communication between the back-end and the front-end. This layer offers a set of functions for the front-end to access the back-end information.

Front-end is the only component that the user will interact with. It has the purpose to present all the information to the user and receive his feedback.

3.7.3 Technologies

To develop the web system we leverage on the following technologies:

- All the technologies used by the system developed for this thesis, such as Stanford NER, Apache Lucene, etc.
- Apache Tomcat web server. This specific web server was chosen as all software and scripts used in this work were developed using the Java programming language.
- MySQL databases. This was the DBMS (Database Management System) chosen to store the information produced in the back-end.
- Twitter API. This API was used to allow the users to recognized and normalize entities in Twitter messages. We intended to use a social network provider in this work.
- PubMed API. This API was used to allow, similarly to Twitter, the users to introduce a simple PubMed identifier to import the whole abstract. Along with the social network provider, we also wanted to support a biomedical provider.

- Web development technologies such as HTML5, JSON, JQuery, Chart.js[1], Bootstrap and Annotator.js[2].

3.7.4 Features

The following features are included in this prototype. Some are illustrated in Figures 3.7 and 3.8:

- Recognition of disease disorders within texts introduced by the user. The text may be introduced using the Tweet's and PubMed's unique identifiers, or by manually introducing the text.

- Crowd-sourcing. The users are able to annotate new entities or even correct entities annotated by the system. Figure 3.8 illustrates this feature.

  The information provided by the users is shown in real time for the remaining users. For example, if two users are reading the same text retrieved from the PubMed database, and if one of the users annotates a new entity, then the other user will be able to see that new recognized entity immediately.

  This information is stored to be used for future iterations where it can be added to the existing knowledge base.

- Real time contextual information. Next to the document, three distinct information items are showed to the users

  - The top ten last submitted texts.
  - The top ten last created annotations by a user
  - The top ten last modified annotations by a user

  This information has a link for the affected document, allowing the users to navigate to them.

- Statistics related to the text being read. The top ten annotations are showed in a circular graphic showing their total frequency. This information was introduced with the intention to allow a user to quickly understand the context of the document that is about to read.

3.8 Summary

This chapter presents the work developed for this thesis. The system developed to accomplish the objectives proposed in Section 1.2 was presented, being each system's module

[1] http://www.chartjs.org/
[2] http://annotatorjs.org/

detailed. The system is composed by two independent modules: the first one is responsible for addressing the recognition task by employing a machine learning algorithm with a rich set of features including Brown clusters. The second one addresses the normalization task and was built based on a novel pipeline architecture which leverages on several strategies for disambiguating the entities, such as semantic similarity, information content and pattern matching algorithms. An extended Levenhstein pattern matching algorithm was also developed to improve the pipeline's performance. The system is a result of two previous iterations that were submitted to the international competition SemEval 2014 and 2015.

A prototype of a user interface was developed, allowing users to annotate any free text, PubMed abstracts and Tweets. The system also produces relevant statistic information for the user, and allows the user to correct the automatic annotation, improving the system future results.

In the next chapter, the system described is evaluated following non-official evaluations performed to each individual module and to the system as a whole. The official evaluations performed to the previous system's iterations will also be described.

Chapter 4

Experimental Results

This section describes the assessment of the developed system followed along with the results obtained. Two types of evaluations were performed: official evaluations resulting from the SemEval 2014 and 2015 participation, and non-official evaluations.

The non-official evaluations (performed locally) are first described, starting by the showing the evaluations carried out in each module individually and then in the system as a whole. Then, the official results obtained in the SemEval 2014 and 2015 competition are presented. In the end, a discussion about the overall improvement of system is provided.

4.1 Evaluation Data Sources

The SemEval 2014 and 2015 evaluations leverage on the data sources described in Sections 2.6.1.1 and 3.2.2, respectively. The SemEval 2015 data sources are also used to perform local evaluations, leveraging exclusively on the training and developing set of clinical notes.

4.2 Assessment

The systems used to participate in SemEval 2014 and 2015 are officially evaluated according to each edition's performance assessment metrics and corpora described in Sections 2.6.1 and 3.5.2 respectively. The system's non-official evaluations were performed according to the metrics of precision, recall, F-score and accuracy, using both the SemEval 2014 and 2015 performance assessment. Both strict and relaxed evaluations were performed. Although taken into consideration, the relaxed evaluations are not as relevant as the strict evaluation.

4.3 Recognition

For the recognition module two distinct approaches were developed.

Run	Strict Evaluation			Relaxed Evaluation		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
1	**0.290**	0.660	0.403	0.551	0.884	0.679
2	0.016	0.676	0.031	0.150	**0.999**	0.261
3	0.287	0.660	0.400	0.554	0.889	0.683
4	**0.290**	0.661	0.403	0.551	0.885	0.679
5	0.286	**0.684**	**0.404**	**0.567**	0.897	**0.695**

Table 4.1: Evaluation results obtained in the recognition task by using a dictionary based approach.

4.3.1 Dictionary Based Approach

This approach was described in Section 3.3.1. It leverages on the annotated data belonging to the training set to generate a dictionary. This dictionary distinguishes two types of entities: the continuous and the non-continuous. Based on this dictionary, and keeping in mind the different entities, I developed several algorithms to accomplish the recognition task.

The dictionary based approach was evaluated through five distinct runs:

- Only continuous entities
  - Run 1: Without single character entities
  - Run 2: With single character entities
- Both continuous and non-continuous entities (Section 3.3.1.2).
  - Run 3: Direct match as a whole
  - Run 4: Relevant direct match
  - Run 5: Relevant direct match with irrelevant window (value used: 50)

4.3.1.1 Results

Table 4.1 shows the results obtained in each run, highlighting the best ones. Runs 1 and 2 only address continues entities, which correspond to 90% of the entities within the clinical notes.

Run 1 represents the baseline results, achieving an F-score of 40.3%. This run recognized more than 60% of the entities within the development clinical notes, with a precision of 29%.

Run 2, unlike Run 1, did not exclude single character entities. These specific entities do not contain almost any valuable information, and without using the context is impossible to distinguish the relevant characters from the irrelevant ones. The precision value drop to an insignificantly 1.6% with a small increase in recall. The relaxed evaluation

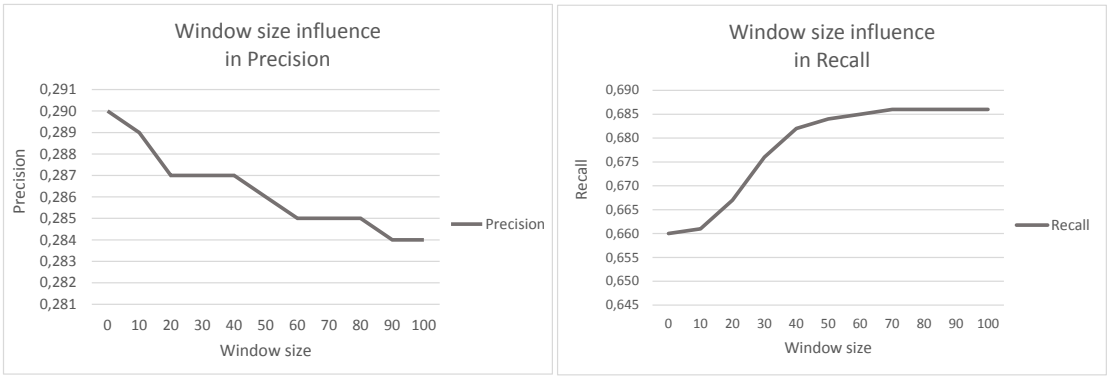

Figure 4.1: Window size precision and recall performance influence in dictionary based approach.

achieved almost 100% recall, as any word that contains the single character entity will be considered a valid match according to the relaxed evaluation. This situation can be validated by observing the number of False Positives. In Run 1 about 13,000 entities were wrongly identified while in the Run 2, this value was over 333,000.

In Runs 3, 4 and 5, non-continuous entities were addressed. The continuous entities in this three new runs were addressed by following the Run 1 approach.

In Run 3, the non-continuous entities are converted into continuous entities by considering both relevant and irrelevant tokens within the relevant entity. The results show that in the strict evaluation, the recall value remains the same while the precision has decreased along with the F-score measure. This indicates that few non-continuous entities were correctly identified.

Run 4 only takes in consideration the relevant tokens to generate a continuous entity. This approach achieved similar results to the ones presented by Run 1. Again, since the irrelevant tokens are not considered, few matches were found, resulting in a small increase of the recall. The overall performance measure by the F-score did not change.

The previous runs failed to address non-continuous entities as they address the irrelevant tokens' information incorrectly. Run 5 employs a more complex approach by leveraging on an *irrelevant window*. This window identifies the irrelevant characters between each identified relevant token. Therefore, an entity is considered relevant if the clinical note contains the relevant tokens separated by a given amount of irrelevant characters defined by the size of the window.

To understand the window size influence, the system was evaluated using distinct window's sizes. The chosen values were limited by the minimum and maximum number of irrelevant characters that separate relevant tokens within the training data. In this case, the training data used has the minimum of 1 and a maximum of 105 irrelevant characters, resulting on an average window of size 50.

Three charts are presented to show the impact of the chosen window value in the

Figure 4.2: Window size F-score performance influence in dictionary based approach.

system's precision, recall and F-score. Eleven window sizes were evaluated by ranging the window size from 0 to 100. The approach with a window of size 0 is identical to the one used by Run 4.

The charts presented in 4.1, show a positive correlation between the window size and the recall value. On the other hand, the precision shows an inverse relation. Figure 4.2 shows the F-score measure variation with the window size, where it is possible to observe that this measure is maximum when a window with an average value is used. Extreme window values have lower results, being the window of size 20 the lowest recorded.

The results showed in the Table 4.1, referring to the Run 5, are obtained by using a window of size 50 (average). A window of size 40 could also have been employed. It is possible to see that this is the highest performance approach to address non-continuous entities.

4.3.2 Machine Learning

A more complex approach, commonly used to address the recognition task, consists on using machine learning algorithms. As described in Section 3.3.2, the Stanford NER software [17] was used to generate CRF recognition models [35] based on a set of features also described in the same section.

This module was evaluated through several runs, each one trained with baseline features (identical to all runs) and a combination of three specific features: model's order, Brown clusters and domain lexicons. It is intended to explore the influence of these features in the module's performance.

The features are added following a specific sequence: (i) model's of first, second and third order, (ii) Brown clusters of different sizes and (iii) domain lexicons. The evaluations

Model's Order	Strict Evaluation			Relaxed Evaluation		
	Precision	Recall	F-score	Precision	Recall	F-score
1	0.779	**0.728**	**0.753**	0.913	**0.867**	0.889
2	**0.780**	0.727	**0.753**	**0.916**	0.866	**0.890**
3	**0.780**	0.727	0.752	0.915	0.866	**0.890**

Table 4.2: Influence of CRF model's order in the performance of the recognition task.

Model's Order	Time (seconds)	Time (hh:mm:ss)
1	5177	01:26:17
2	20142	05:35:42
3	73914	20:31:54

Table 4.3: Time spent training each model's order.

are cumulative, meaning that each feature leveraged on the best run results obtained by the previous feature evaluations.

4.3.2.1 Model's Order

A CRF model's order is defined by the size of the window (context) which represents the number of tokens, in the left and right of the target's token, to be kept in memory and used to infer new features. For example, a second order's model will keep in memory the two tokens at left and right of the target's token.

Results

Three models were trained using the clinical notes from the training set, where the only difference was the window used.

The results obtained are described in Table 4.2, highlighting the best ones for each measure. Table 4.3 shows the amount of time required to train each model, being the results presented an average of three executions.

The results show that the model's order does not have much influence in the performance. As it is possible to see in the Table 4.2, all three models have almost identical F-score measures.

The first order model achieved the highest recall measure but also the lowest precision. As for the second and third order models, their results are almost identical with a small decrease in performance in the third order model. The expansion of the context maintained in memory by the model seems to have resulted in a decrease of precision, without any changes in the recall. The results also suggest that first order models are more likely to achieve higher recall values in detriment of the precision, while the results obtained by higher order models suggest the opposite.

#Clusters	Strict Evaluation			Relaxed Evaluation		
	Precision	Recall	F-score	Precision	Recall	F-score
0	0.779	0.728	0.753	0.913	0.867	0.889
50	**0.781**	0.730	**0.755**	0.914	0.866	0.889
100	**0.781**	0.729	0.754	**0.915**	0.866	0.890
150	0.780	0.729	0.754	0.914	0.869	**0.891**
200	0.780	**0.731**	**0.755**	0.913	**0.870**	**0.891**

Table 4.4: Influence of Brown clusters in the performance of the recognition task leveraging on first order models.

Although the results obtained by the three models are almost identical, Table 4.3 shows that the time needed to generate each model is very distinct. The amount of the context needed to maintain memory to infer composed features dramatically increases the time needed to train the model. Table 4.3 shows that a first order model requires around one hour and a half, almost 4 time less than the time required to train a second order model, and around 14 times less than a third order model.

Based on this evaluation, it is clear to say that the third order models require a huge amount of time to train, without having any performance improvement. As for the first and second order models, their overall performance is similar. Since there is no significant improvement, future evaluations will use first order models, reducing the time required to execute the training procedure.

4.3.2.2 Brown Clusters

To evaluate the influence in adding Brown clusters as a feature during the training process, first order models were generated using Brown clusters with 50, 100, 150 and 200 clusters. The clusters were generated using as knowledge, all the unlabelled notes available, comprising a total of 404k clinical notes (Section 3.2.2.2).

Results

The results obtained are presented in Table 4.4. Each entry consists on a first order model generated by using a given amount of Brown clusters. The first entry represents the baseline results obtained in the previous evaluation (Section 4.3.2.1).

The addition of Brown clusters to the model generation increases the recognition performance independently of the number of clusters chosen. In these evaluations, using 200 Brown clusters allowed me to achieved the highest results, increasing the F-score measure in two percentage points. This increase was mainly due to the increase of the recall value.

Table 4.5 allows one to understand the time effort required to use this feature. Both the time required to generate each amount of clusters (one time event) and the time required to train the recognition model using the respective clusters as a feature is presented.

#Clusters	Generate (seconds)	Generate (hh:mm:ss)	Train (seconds)	Train (hh:mm:ss)
0	-	-	5177	01:26:17
50	1281	00:32:31	4160	01:09:20
100	6086	01:41:26	4294	01:11:34
150	12455	03:27:35	4244	01:10:44
200	21635	06:00:35	3907	01:05:07

Table 4.5: Time spent generating each set of Brown cluster and training the respective model.

Analyzing the values present in the Table 4.5, it is easy to see a pattern where the more clusters are generated, the more time is required. For example, generating 50 Brown clusters takes only 6% of the time required to generate 200 Brown clusters.

Table 4.5 shows that the addition of Brown clusters does not increase the amount of time required to train the model. For instance, it is observed that the addition of these features allows one to slightly decrease the training time, being the model with higher performance the one which took less time to train.

Domain Lexicon

The following runs consist on a first order model trained with 200 Brown clusters, along with one or more of the domain lexicons described in Section 3.3.2.1.

The lexicons were generated with basis on the two main knowledge sources (SNOMED CT and Golden corpora). Each domain lexicon was first individually evaluated to understand their individual impact. Then, all domain lexicons were combined to understand their impact as a whole. The OBO and the DBPedia lexicons were not evaluated individually because they are external knowledge and, therefore, I assumed they have less confidence.

Results

The best results obtained are highlighted in Table 4.6. These evaluations show that the addition of domain lexicons as features results on an overall decrement of the system's performance. The Golden domain lexicon is the only lexicon that does not affect the system's performance when used individually. This result was expected since the information is already represented through the SBIEON classification also used as a feature.

Although the results got worse by using these lexicons, a pattern is observable. First, the precision measure has decreased drastically while the recall have slightly increase. These results were expected but never with a decreasing of the F-score measure. Some evaluations show a slightly improvevement in the relaxed F-score measure, but I ignored this improvement as it implies lower strict evaluations.

Comparing the evaluations without taking into consideration the performance loss, it

Domain Lexicons	Strict Evaluation			Relaxed Evaluation		
	Precision	Recall	F-score	Precision	Recall	F-score
None	**0.780**	0.731	**0.755**	**0.913**	0.870	0.891
Golden	**0.780**	0.731	**0.755**	**0.913**	0.870	0.891
All-SNOMED	0.770	0.733	0.751	0.907	**0.878**	**0.892**
Separated-SNOMED	0.769	**0.734**	0.751	0.906	0.877	**0.892**
{All-SNOMED | OBO | DBPedia | Golden }	0.771	**0.734**	0.752	0.907	0.877	**0.892**
{Separated-SNOMED | OBO | DBPedia | Golden }	0.771	**0.734**	0.752	0.907	0.876	0.891

Table 4.6: Influence of domain lexicons in the performance of the recognition task leveraging on first order models trained with 200 Brown clusters.

is possible to observe that using the entities comprised within the SNOMED CT ontology as a single lexicon or as separated lexicons, produces the same results. A slightly variation in recall is observed but it is not consistent to be taken into consideration.

The models which used a conjugation of all available lexicons have showed to perform better than the ones where each domain lexicon is used independently. Even with a not so significantly performance increase, it is possible to verify that the use of a separated SNOMED domain lexicon achieved better relaxed evaluations but again, the performance was not significant enough to result in a definitive conclusion.

4.3.3 Discussion

Two distinct strategies were evaluated to address the recognition task.

Addressing the recognition task with a dictionary based approach achieved poor results with an F-score of 40.4%, mainly due to the system's poor precision (less than 30%). The amount of information retrieved by the system is considerable, being retrieved 68% of the existing entities within the clinical notes.

The second strategy employed a machine learning technique. Using just the basic features to generate CRF models, proved to achieved better results than the ones established by the dictionary approach. Tehse results were improved by adding Brown clusters as features. Domain lexicons must only be used when it is intended to increase the recall value, knowing that the precision and the overall system performance will be degraded. With this approach, 73% of the entities are recognized with a 78% of precision, resulting on a 75.5 strict F-score.

4.4 Normalization

The normalization module follows the pipeline framework described in Section 3.4. To evaluate the normalization task independently of the recognition module's performance, the entities present in the development dataset are used as input, simulating a perfect recognition module.

For each component of this pipeline (Section 3.4.3.4), several approaches were eval-

Algoritm	Precision	Recall	F-score	Component's Accuracy	Time (s)
Non-Ambiguous Entities (1st section)					
Abbreviation dictionary lookup	0.977	0.060	0.113	**97.70%**	2.1
Golden dictionary lookup	0.965	0.639	0.769	**96.34%**	2.2
SNOMED dictionary lookup	0.921	0.753	0.829	67.88%	5.3
SNOMED dictionary lookup (check disorders)	0.952	0.753	0.841	**88.23%**	5.7
Ambiguous Golden Entities					
Ambiguous Golden Entities (frequency)	0.933	0.798	0.860	60.97%	6.9
Ambiguous Golden Entities (frequency * IC{Seco} ; CUI-less: 0.5)	0.934	0.798	0.860	61.54%	7.0
Ambiguous Golden Entities (frequency * IC{Seco} ; CUI-less: 1)	0.934	0.798	0.860	61.54%	7.0
Ambiguous Golden Entities (frequency * IC{Seco} ; CUI-less: 0)	0.935	0.798	0.861	**63.25%**	7.0
Ambiguous SNOMED Entities					
Ambiguous SNOMED Entities (SemanticSimilarityJian)	0.932	0.801	0.862	35.71%	7.8
Ambiguous SNOMED Entities (SemanticSimilarityLin)	0.932	0.801	0.862	35.71%	7.8
Ambiguous SNOMED Entities (SemanticSimilarityResnik)	0.932	0.801	0.862	46.43%	7.8
Ambiguous SNOMED Entities (SemanticSimilarityJian + check disorders)	0.933	0.801	0.862	57.14%	8.0
Ambiguous SNOMED Entities (SemanticSimilarityLin + check disorders)	0.933	0.801	0.862	57.14%	8.0
Ambiguous SNOMED Entities (SemanticSimilarityResnik + check disorders)	0.934	0.801	0.862	71.43%	8.0
Ambiguous SNOMED Entities (SemanticSimilarityResnik + check disorders + Limited base knowledge)	0.934	0.801	0.862	**75.00%**	8.0
Similarity Search					
Similarity Search ({Golden+SNOMED} + {Lev: 1 | ExtLev: 0 | Ng5: 0})	0.864	1.0	0.927	58.45%	77
Similarity Search ({Golden+SNOMED} + {Lev: 0.15 | ExtLev: 0.15 | Ng5: 0.7})	0.861	1.0	0.925	57.06%	80
Similarity Search ({Golden+SNOMED} + {Lev: 0.7 | ExtLev: 0.15 | Ng5: 0.15})	0.865	1.0	0.928	**58.64%**	80
Similarity Search ({Golden+SNOMED} + {Lev: 0.15 | ExtLev: 0.7 | Ng5: 0.15})	0.866	1.0	0.928	**59.46%**	80

Table 4.7: Normalization module's components evaluation results.

uated leveraging on the previous components' best results. Each approach is evaluated according to their precision, recall, F-score and accuracy (Section 4.2). These measures allow one to better understand the impact of each component and respective approach in the overall module's performance.

4.4.1 Results

Table 4.7 presents the results obtained in each module's components evaluation. The table is split into four sections, one for the non-ambiguous entities (first pipeline's section), and the other three for each of the remaining components (second pipeline's section).

Since the pipeline is organized so that the higher confidence entities are first addressed, it is expected that the first evaluations have a higher precision value with lower recall. As the pipeline is executed, the precision decreases in detriment of the recall, resulting in a higher F-score measure. The table also shows the accuracy of each component, i.e. the percentage of entities which were correctly normalized by that component using a given strategy. The time required to execute each approach is also taken into consideration to understand its impact during the pipeline execution.

Non-Ambiguous Entities

In this first evaluation, non-ambiguous entities belonging to three distinct dictionaries are addressed.

Abbreviation dictionary lookup (first component) is the approach with the highest confidence from all that are present in the system, with a 97.7% of accuracy. On the other hand it has also the lowest recall since the dictionary is limited.

The Golden dictionary lookup (second component) achieves almost the same levels of precision of the previous component (96.5%), with a much higher recall (63.9%). This component addresses a larger amount of entities, being able to maintain the normalization quality. With this component, the F-score measure increases largely.

Finally, the SNOMED dictionary lookup (third component) is the one with the poorest results, where two approaches were evaluated. When assigning the identifier associated in the SNOMED dictionary without any further evaluation, a 67.88% accuracy is achieved. With this first approach, the entities are always normalized with a CUI even when the CUI-less identifier should be assigned. By evaluating if the normalized entity belongs to the disease disorder semantic type using the method proposed in Section 3.4.3.3, the accuracy increased to 88.23%, sresulting on a 20% improvement.

By leveraging exclusively on these approaches to normalize entities, a 81.8% F-score is achieved with a 95.2% precision value. Since only non-ambiguous entities are considered, only 71.7% of the existing entities were addressed.

Ambiguous Golden Entities

This is the fourth component in the pipeline and also the first one addressing ambiguous entities. This component leveraged on the annotated data available, to calculate the frequency of a given concept. By leveraging only on this measure, 60.97% of the entities were correctly normalized.

A higher F-score measure was achieved by generating a new measure resulting from the linear combination of two measures: information content (according to Seco's algorithm) and the frequency of a given concept. Since some entities within the Golden dictionary may have the CUI-less identifier assigned, three evaluations were performed where this identifier had the highest, lowest and average information content value. According to Seco's algorithm, the information content values are comprised within the values of 0 and 1.

The results show that any strategy increased the precision value and thus the overall accuracy. Giving less relevance to the CUI-less identifier was the strategy which results on higher performance, increasing the accuracy in almost 3%.

Ambiguous SNOMED Entities

The fifth component in the pipeline. It leverages on the SNOMED dictionary (ambiguous) and uses a semantic similarity approach to disambiguate entities (Section 3.4.3.2). A total of seven approaches were evaluated, comprising the three known semantic similarity algorithms: Jiang and Conrath [29], Lin [41] and Resnik [58], with and without checking if the chosen CUI belonged to the disease disorders semantic type.

All semantic similarity algorithms leverage on Seco's information content algorithm. The results show that the algorithm from Lin and Jiang achieved the same results. Resnik's

algorithm, on the other hand, exceeded by almost 15% the results achieved by both Jiang and Lin when validating the assigned identifier. Validating if the identifier belongs to the disease disorder semantic group improves largely the results.

This approach leveraged on a knowledge base containing all the normalized entities during this pipeline. In the previous evaluations, entities normalized by any component were added to the knowledge. Since not all components have the same confidence, a new evaluation was performed, leveraging on the previous best evaluation and limiting the addition of new normalized entities to four components: the three components which addresses non-ambiguous entities and this component (Ambiguous SNOMED Entities). The results show that this limitation allowed me to increase the confidence of the knowledge base and thus increasing the accuracy in almost 4%.

The recall measure has not increased significantly from the previous component, showing that few entities were normalized by this component.

Similarity Search

This is the last component of the pipeline. It addresses all entities which are not comprised in any dictionary by using a pattern matching algorithm to retrieve the most suitable candidate from the SNOMED and Golden dictionary. The abbreviation dictionary was not used because it only contains abbreviations which should only be found using a direct match approach. The retrieved candidate is then normalized by leveraging on the most suitable approach used by any of the previous components.

Since the chosen entity will be normalized by any of the previous components, the performance of this component will be related to the quality of the identified candidates. For that, the formula described in Section 3.4.3.4 was developed. This formula leverages on three distinct pattern matching techniques: Levenshtein (fine grain), NGram5 (higher grain) and a new measure which I named ExtendedLevenshtein. This distance is based on the Levenshtein distance but does not take into consideration the token's order (Section 3.4.3.1).

The results show that leveraging only on the Levenshtein distance to retrieve the most suitable candidate achieves a reasonable accuracy, which was considered as a baseline. Next, the formula composed by the three measures is evaluated, ranging their weights. One of the algorithms will have a 0.7 weight and the remaining two 0.15. It was intended that one of the algorithms had more relevance than the other two added. The following results were observed:

- Giving the highest weight to the Levenshtein distance slightly improved the performance, showing that the combination of algorithms brings additional knowledge to the component.

- Giving the highest weight to the NGram5 algorithm, produces the lowest results.

This shows that giving a higher relevance to higher granularity algorithms reduces the performance.

- Giving the highest weight to the ExtendedLevenshtein distance achieved the highest performance. This means that the performance increase results on identifying new candidates which were considered distinct, and thus drop, due to their token's order.

Running Time

The elapsed time of each component's approach was measured, being the results presented in Table 4.7. For each run, three executions were performed, and the results that are presented correspond to an average of the time required for each execution to end. As expected, the time required for each component scales through the pipeline. This time increase is a result of the employment of more complex strategies and the employment of extra knowledge sources.

Without taking into consideration the last component, the pipeline requires a reasonable 8 seconds, being this value not far away from the 2 seconds required to execute the first component individually. Adding the last component to the pipeline increased the execution time to 80 seconds. This is by far the most time consuming component as it not only uses a recursive approach to retrieve the most similar candidate using a similarity search approach, but also leverages on the previous components to finish the normalization.

4.4.2 Discussion

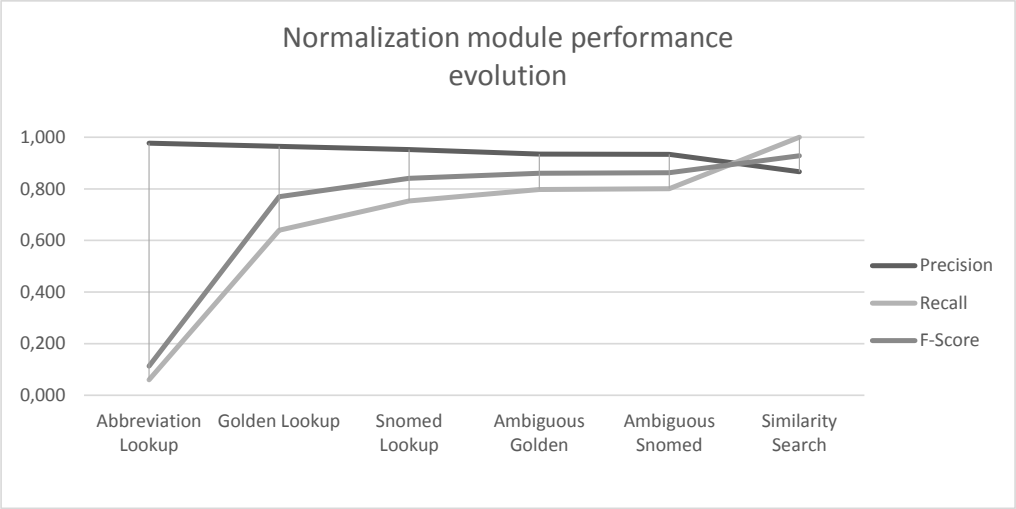

Figure 4.3: Normalization module performance evolution.

To better understand the normalization module's performance, I created a chart (Figure 4.3) comparing the precision, recall and F-score measures obtained in each compo-

nent. Leveraging on this graphic and on the evaluation results that were showed and discussed on the previous section (Table 4.7) the following conclusions can be reached:

- The component's order was suitable. In Figure 4.3 it is possible to observe a slightly decrease of the precision value resulting from the normalization of entities by components with lower confidence.
- Although the precision has decreased, the recall value increased along with the F-score, showing that the pipeline is normalizing correctly most of the entities.
- The first section of the pipeline, which addresses non-ambiguous entities, achieved considerable results with high precision standards. Important to highlight the second component's performance, which normalized almost 70% of the entities maintaining the precision value. This is one of the most crucial components in the pipeline.
- The semantic similarity approach achieved a reasonable accuracy. However, the recall variation suggests that this component did not addressed a considerable amount of entities and thus it is the one with the least impact in the pipeline.
- Validating if the chosen CUI belongs to the disease disorder semantic type is critical for the system performance. Approaches which leverage on the SNOMED dictionary, and that validate the chosen CUI, achieved higher results.
- The last component resulted in a noticeable and dramatically decrease of the precision value, in favor of reasonable increase in both recall and F-score value. It shows that this component addresses several entities but with a low confidence.
- Finally, the evaluations performed showed that the Extendedlevenshtein and the formula by me created resulted on higher accuracy values, which lead to a better system's performance.

The module's final evaluation consists on a promising 92.8% F-score with a 86.6% precision. This means that from all existing entities, 86.6% were correctly normalized. This results are considerably higher when comparing to the ones obtained by the recognition module.

4.4.3 System

In the previous sections, the individual module's results were presented. I leveraged on the knowledge obtained with these evaluations to define which strategies are going to be employed on the last system iteration. The top performance strategies were the ones chosen. The recognition module leverages on a first order model trained using 200 Brown

Evaluation	Strict Evaluation			Relaxed Evaluation			Strict	Relaxed
	Precision	Recall	F-score	Precision	Recall	F-score	Accuracy	
SemEval2014	0.780	0.731	0.755	0.913	0.870	0.891	0.668	0.913
SemEval2015	0.712	0.668	0.689	0.746	0.699	0.722	-	-

Table 4.8: Performance on development data for the last system iteration.

clusters. The normalization module leverages on the pipeline presented in the previous section.

Table 4.8 presents the results obtained when evaluating this system using both the SemEval 2014 (Section 2.6.1.2) and SemEval 2015 (Section 3.5.2.2) performance assessment. The former allows to evaluate each system's modules individually while the latter evaluates them as a unique task.

The recognition module correctly recognized around 73% of the available entities with a 78% precision. Leveraging on the relaxed evaluation, these values increase to 87% of recall and 91.3% of precision. The result's discrepancy suggests that several entities are only being partially recognized.

The normalization module achieved 66.8% and 91.3% accuracy in strict and relaxed evaluation respectively. The former indicates that around 67% of the existing entities within the evaluated clinical notes were correctly normalized. The latter is more related to the module's performance, showing that it was able to correctly normalize more than 90% of the correctly recognized entities.

In the SemEval 2015 evaluation, strict and relaxed results are only distinguished by the recognition task. On the previous evaluation, a considerable amount of partial recognized entities were recognized when using the relaxed evaluation. This pattern is no longer visible in this evaluation, being the relaxed and strict results very similar. This suggests that the normalization module is not being able to correctly normalize partial identified entities.

Overall, the system achieved considerable results, being the normalization module the one with the highest performance.

4.5 Official Evaluations

The system presented in this thesis is a result of several versions and improvements. The first two versions were used to participate in the SemEval 2014 and SemEval 2015 competitions respectively, where they were officially evaluated along with other teams across the globe. In this section these two first system prototypes are briefly described along with the official results obtained.

Place	Team ID	Strict Evaluation			Relaxed Evaluation			Data
		Precision	Recall	F-score	Precision	Recall	F-score	
1	UTH_CCB	0.843	0.786	0.813	0.936	0.866	0.900	T+D
2	UTH_CCB	0.808	0.805	0.753	0.916	0.907	0.911	T+D
3	UTU	0.765	0.767	0.752	0.886	0.899	0.893	T+D
4	UWM	0.787	0.726	0.752	0.911	0.856	0.883	T+D
5	UTH_CCB	0.680	0.849	0.752	0.838	0.935	0.884	T+D
14	ULisboa	0.753	0.663	0.705	0.914	0.815	0.862	T
15	ULisboa	0.752	0.660	0.703	0.909	0.806	0.855	T
16	ULisboa	0.752	0.660	0.703	0.909	0.806	0.855	T
40	QUT_AEHRC	0.387	0.298	0.337	0.906	0.709	0.795	T+D
41	SZTE_NLP	0.571	0.205	0.302	0.918	0.325	0.480	T

Table 4.9: Performance on test data for participating systems on the recognition task of SemEval 2014 [56].

4.5.1 SemEval 2014

The system used to participate in the SemEval 2014 competition was described in Section 2.6.3.1. The following subsections present the most relevant systems' results in each task assessed.

4.5.1.1 Recognition

Table 4.9 shows the top five and bottom two systems' results, comparing them to the runs submitted by my team. This table contains both the strict and relaxed evaluation along with the datasets used as training set. The UTHC_CCB team system, described in Section 2.6.3.2, was the one with the best approach, managing to have the three submitted runs among the top five. Our system on the other hand only managed to achieve the 14º position. Although it seems a poor result, it is important to take into consideration that the amount of training data used was reduced. The top five recognition systems have used the clinical notes from both the training and developing datasets as training data. Our system, on the other hand, only leveraged on the training set, still being able to overtaken some systems which used a larger amount of clinical notes for training.

The submission with higher performance, achieving the 14º position, was the only submission which leveraged on the SBIEON encoding, showing that it can be used to address non-continuous entities.

The organization, to allow a uniform comparison among all systems, including the best system in the ShARe/CLEF 2013 competition, organized a new evaluation where all the systems leveraged exclusively on the training set to train and the development set to evaluate.

In this evaluation (Table 4.10), our system achieved the 8th best F-score measure, showing that the amount of data used to train the system is critical to achieve higher recognition results. The top system' results have decreased as a result of using this training set

Place	Team ID	Strict Evaluation			Relaxed Evaluation			Data
		Precision	Recall	F-score	Precision	Recall	F-score	
1	TMU	0.687	0.922	0.787	0.952	1.000	0.975	T
2	UTH_CCB	0.877	0.710	0.785	0.962	0.789	0.867	T
3	UTH_CCB	0.828	0.747	0.785	0.941	0.853	0.895	T
-	Best ShARe/CLEF Performance	0.800	0.706	0.750	0.925	0.827	0.873	T
4	UWM	0.827	0.675	0.743	0.958	0.799	0.871	T
5	ezDI	0.813	0.670	0.734	0.954	0.800	0.870	T
6	ezDI	0.809	0.667	0.732	0.954	0.801	0.871	T
7	UTH_CCB	0.657	0.790	0.717	0.806	0.893	0.847	T
8	ULisboa	0.803	0.646	0.716	0.954	0.871	0.858	T
15	IITP	0.467	0.440	0.453	0.812	0.775	0.793	T
16	IITP	0.493	0.410	0.448	0.828	0.706	0.762	T

Table 4.10: Uniform evaluation of performance on development data for participating systems on the Recognition task of SemEval 2014 [56].

Place	Team ID	Strict	Relaxed	Data
		Accuracy (%)	Accuracy (%)	
1	UTH_CCB	74.1	87.3	T+D
2	UTH_CCB	70.8	88.0	T+D
3	UTH_CCB	69.4	88.3	T+D
4	UWM	66.0	90.9	T+D
5	RelAgent	63.9	91.2	T+D
25	ULisboa	40.5	61.5	T
26	ULisboa	40.4	61.2	T
27	ULisboa	40.2	60.6	T
34	CogComp	25.3	47.9	T+D
35	CogComp	24.8	47.7	T+D

Table 4.11: Performance on test data for participating systems on the normalization task of SemEval 2014 [56].

with a lower amount of clinical notes. Comparing to the best ShARE/CLEF system, our system achieved higher precision but a considerable lower recall.

The relaxed results obtained in both evaluations show that all systems had similar results. With the exception of the bottom system, the remaining have achieved above 80% of precision and 70% of recall. Strict results on the other have a larger distribution, having systems with around 80% of precision and recall while others have values around the 30%.

4.5.1.2 Normalization

The second task in this competition consisted on the normalization of the recognized entities. Both strict and relaxed evaluation were performed. The first one evaluates the task considering all the existing entities within the test dataset, while the second one considers only the correctly recognized entities. Again, as not all the systems have used

Place	Team ID	Strict	Relaxed	Data
		Accuracy (%)	Accuracy (%)	
1	TMU	71.6	77.7	T
2	TMU	71.6	77.7	T
3	UTH_CCB	71.3	90.3	T
4	UTH_CCB	68.0	91.0	T
5	UTH_CCB	64.7	91.0	T
-	Best ShARe/CLEF Performance	58.9	89.5	T
12	ULisboa	38.5	60.1	T
16	IITP	31.2	72.5	T
17	IITP	29.9	73.0	T

Table 4.12: Uniform evaluation of performance on development data for participating systems on the normalization task of SemEval 2014 [56].

the same amount of clinical notes for training, a second evaluation was performed in order to compare the systems uniformly.

Table 4.11 shows the official results obtained in this task. The top five and bottom two systems, along with our three submissions, are presented. Table 4.12 shows the uniform evaluation which also includes the best ShARe/CLEFF system results.

Our best system's submission achieved a 25th position among 37 submissions. These results show that the approach taken was unreliable, achieving a relaxed accuracy close to 60% and a strict accuracy of 40%.

Table 4.12 shows the system's ranking when evaluated uniformly. Our system was able to achieve the 12º place, showing that some top performance systems are dependent of the amount of training data used to achieve better results. UTH_CCB was the only top 5 system that achieved similar results, indicating that their strategy it is not entirely dependent of the training data. Our results are still way above the ones achieved by the best ShARE/CLEF system.

A special mention for the UTH_CCB team which, achieved one of the highest relaxed accuracy scores. The 90% relaxed accuracy shows that their strategy to normalize entities is effective when the entities are well recognized.

4.5.1.3 Discussion

In the previous sections, I showed that the first system version did not achieve the best results. This participation on SemEval allowed me to understand the weaknesses of the system developed. Although the recognition component achieved some promising results, the normalization was poor, resulting on a weak system. The normalization component was employing a wrong approach, which was required to be changed in future iterations. As for the recognition, the results were not perfect but they can be improved by tweaking the machine learning algorithm features.

Place	Team ID	Run	Strict Evaluation			Relaxed Evaluation		
			Precision	Recall	F-score	Precision	Recall	F-score
1	ezDI	1	0.783	0.732	0.757	0.815	0.761	0.787
2	ULisboa	3	0.779	0.705	0.740	0.806	0.729	0.765
3	UTH_CCB	3	0.778	0.696	0.735	0.797	0.714	0.753
4	UWM	2	0.773	0.699	0.734	0.809	0.731	0.768
5	UTH_CCB	1	0.748	0.713	0.730	0.777	0.741	0.759
6	UTH_CCB	2	0.748	0.713	0.730	0.777	0.741	0.759
7	TAKELAB	1	0.761	0.696	0.727	0.794	0.727	0.759
8	ULisboa	2	0.749	0.681	0.713	0.780	0.709	0.743
9	Bioinformatics-UA	2	0.690	0.736	0.712	0.719	0.766	0.742
10	Bioinformatics-UA	3	0.691	0.735	0.712	0.720	0.765	0.742
11	ULisboa	1	0.748	0.676	0.710	0.782	0.706	0.742
12	CUAB	1	0.735	0.683	0.708	0.762	0.708	0.734
38	Sanj-TUM	3	0.098	0.110	0.104	0.444	0.496	0.469
39	Sanj-TUM	1	0.082	0.107	0.093	0.425	0.552	0.481

Table 4.13: Official SemEval 2015 results [16].

4.5.2 SemEval 2015

The 2015 competition, as described in Section 3.5.2, is very similar to the SemEval 2014 edition. The system I developed for this competition leveraged on the existing SemEval 2014 system, being the main differences pointed out in Section 3.6.

A total of three runs, described in Section 3.5.3 were submitted and officially evaluated in this competition.

4.5.2.1 Results

Table 4.13 summarizes the official results obtained in SemEval 2015 *Analysis of Clinical Text* task. This table presents the results obtained by the first twelve systems and the bottom two, contemplating my three submissions.

My best run achieved the second best F-score measure. This run had the particularity to be identical to the one which reach the eleventh place, but leveraging on a larger set of clinical notes for training. The increased performance was noticeable, increasing both the system's precision and recall in about 3%.

As described in Section 4.3.2.2, the addition of domain lexicons decreased the overall system performance. This effect was also observed in the evaluations, with Run 1 achieving worst results than Run 2.

Some of the teams which participated in this year's edition also took part in the SemEval 2014 edition. For instance, the team UTH_CCB was the one that achieved the best F-score measure in 2014's edition. These results show that my best run was able to overtake all the UTH_CCB team submissions, showing a positive evolution of the system.

My best submission strictly recognized and normalized with success 70.5% of the available entities with a 77.9% precision. These results are not very distinct from the

System/Evaluation	Strict Evaluation			Relaxed Evaluation			Strict	Relaxed
	Precision	Recall	F-score	Precision	Recall	F-score	Accuracy	
SemEval 2014 Evaluation								
SemEval 2014 System	0.771	0.735	0.753	0.907	**0.878**	**0.892**	0.443	0.602
SemEval 2105 System	0.771	**0.736**	0.754	0.907	**0.878**	**0.892**	0.665	0.905
Final System	**0.780**	0.731	**0.755**	**0.913**	0.870	0.891	**0.668**	**0.913**
SemEval 2015 Evaluation								
SemEval 2014 System	0.464	0.443	0.453	0.473	0.450	0.461	-	-
SemEval 2105 System	0.698	0.665	0.681	0.732	0.698	0.715	-	-
Final System	**0.712**	**0.668**	**0.689**	**0.746**	**0.699**	**0.722**	-	-

Table 4.14: Evaluation results comparison between SemEval 2015 system and the last system iteration.

71.4% recall, 79.7% precision and 76.5% F-score obtained in the relaxed evaluation. Besides the second best F-score, the system also held the second best precision.

4.5.2.2 Discussion

These results show that this second system iteration represents a considerable improvement when compared to the first one. Although it is not possible to perform a direct comparison between the results presented, it is possible to infer that the normalization module was likely the main reason for the performance improvement, as the recognition module leverages on a similar strategy and the normalization module was built from the scratch based on a novel pipeline architecture.

4.6 System Comparison

The system developed for this work was described, and compared to its first versions in Section 3.2. In this section, the results obtained by each of these iterations are compared, aiming to understand the positive evolution of the system through this three iterations.

The systems' results presented in the previous sections cannot be compared directly as they were trained and evaluated using distinct clinical notes. Therefore, new evaluations were performed, were each system was trained and evaluated using the SemEval 2015 training and development sets respectively.

Table 4.14 presents the results obtained by each system when evaluated according to both the SemEval 2014 and 2015 assessments. The first assessment evaluates each module's performance individually while the second evaluates the system as a whole.

The recognition module was the one who suffered the least change, with the first two iterations presenting almost identical results. The slight increase in recall is due to the addition of golden named entities as a domain lexicon (Section 3.2). The third and last iteration shows a positive evolution although not significantly enough. A higher recall was obtained in detriment of the precision, resulting um a slightly higher F-score measure.

System/Evaluation	Recognition Time	Normalization Time	Total Time
SemEval 2104 System	125s	128s	253s
SemEval 2105 System	128s	270s	398s
Final System	**32s**	**80s**	**102s**

Table 4.15: Time required for each system iteration to execute the recognition and normalization task.

Unlike the recognition module whose strategy is very similar among the three iterations, the normalization module presents considerable changes. Each new system iteration resulted in increased performance, being the biggest improvement registered between the first and the second iteration. This variation results from the employment of the new pipeline architecture, allowing to correctly normalize more 22% of the existing entities within the clinical notes evaluated. It also increased the relaxed accuracy in 30%, resulting on a correct normalization of 90% of the correctly recognized entities. The final system leverages on this new architecture extended with new components and novel approaches, resulting on an increase of 1% over the relaxed accuracy.

The results of the system as a whole translate the individual results described above. The first iteration is the one with the lowest results, showing a precision and recall below the value of 50% in both strict and relaxed evaluations. These results were largely improved in the second iteration, by recognizing and normalizing 66.5% of the existing entities with a precision of 69.8%. The final system presented in this work increased these results to 66.8% recall with 71.2% precision. The relaxed evaluations are very alike to the strict evaluations in all three iterations, showing that all strategies fail to correctly normalize partial recognized entities.

Each system's running time was also taken into consideration. Table 4.15 presents the time required for each system iteration to perform the recognition and normalization task.

Both the first and second system iterations used similar recognition features resulting on similar execution times. The third iteration did not include any domain lexicon, which not only increased the performance but also reduced in 75% the time required to execute the recognition model.

As for the normalization module, the strategy used in the second iteration produced higher results, but also required almost twice of the time needed to execute the first system's strategy. The third iteration was a lot more efficient, requiring 38% less time than the first iteration and 70% less time than the second iteration, while delivering higher results.

These results show that the final system was not only a successful improvement in terms of performance, but also in efficiency. The final system overall execution when compared to the first iteration, resulted on an increase of 31% of relaxed accuracy and 23% in F-score, but also decreased in 60% the time required to execute the system.

Chapter 5

Conclusions

This work's major objective consisted on the development of a high performance system capable of accomplish two major tasks, namely to recognize and normalize disease disorders within clinical notes. These two tasks bring up several challenges related to the type of the data, being the recognition of disjoint entities and the normalization of ambiguous entities the most demanding, among several others.

A modular system was designed and developed to address these two tasks and the challenges that are associated. The system's recognition module is based on a supervised machine learning algorithm represented through CRF models, which are generated with the Stanford NER software. These models are generated using several features including Brown clusters, an unsupervised machine learning technique. A modified version of this software was used in order to classify each entity according to the novel SBIEON segment representation, allowing one to address disjoint entities.

The normalization leverages on a novel modular pipeline architecture composed by six distinct components, each one responsible for a specific set of entities. These components use a rule based approach and leverage on several strategies. To retrieve and normalize the most appropriate candidate, dictionary lookup and pattern matching algorithms are employed. The ambiguous candidates are resolved using different strategies according to the entity's type, which include using their information content, frequency and a semantic similarity approach. Theses techniques were proved to be effective in disambiguating entities, addressing this challenge. The pipeline developed was also capable of correctly assign the *CUI-less* identifier by using a validation algorithm which checks if the assigned CUI is a disease disorder.

This system is a result of two previous iterations that were used to participate in the SemEval 2014 and 2015 international workshop. These participations allowed me to assess the system officially, comparing it to the systems submitted by other teams across the globe. The SemEval 2015 participation was the most relevant for this work. The system achieved the second best F-score among 16 teams and 38 submissions, with a 77.9% precision and a 70.5% recall, resulting on an F-score of 74%. This participation

represents a major improvement when compared to the 14^o and 25^o position achieved in the SemEval 2014 edition, for the recognition and normalization task respectively. Note that these 16 teams represent the state-of-the-art on this field, and in my master thesis work I was able to outperform most of the systems.

The three system's versions were compared using the same datasets for training and evaluation. The second system iteration resulted on a performance increase of more than 20% when compared to the first one. This discrepancy is a result of a better normalization model which achieved a 90% accuracy, 30% more than the strategy used by the previous iteration. The third and last iteration consisted on a fine tuning of the system leading to an increase of almost 1% of the system's F-score. The system efficiency was largely improved, decreasing the amount of time required to execute the system in about 60%.

For the system described, an user interface prototype was developed comprising the basic features which consists on the recognition and normalization of medical text introduced by users. Additional features were developed including real time global statistics, crowd sourcing data retrieval and document retrieval from social and biomedical document providers.

5.1 Future Work

One of the challenges presented in the recognition task that was not addressed in this work consists on recognizing overlapped entities. This work does not provide any proper solution for this challenge, remaining an open issue that should be addressed to improve the recognition module's performance. This work also leveraged on a simple tokenizer algorithm, being the employment of more complex algorithms a future work that should improve the system performance. New features (e.g., part-of-the-speech, lemmas, etc.) can also be explore to enhance the set of features used during the training.

The pipeline architecture designed for the normalization module has proven to be efficient, achieving 91.3% accuracy. However, it leverages on a rule based approach which can be automated using a learning to rank approach, similar to the D-Norm system. This approach can be used for retrieving the most proper candidate based on a set of features, and thus replacing the pattern matching approach used in this work. Another open issue consists on the correct normalization of partially recognized entities, as the system fails in normalizing them with success.

Due to the amount of biomedical information available, namely annotated clinical notes and controlled vocabularies, this work leverage entirely on English native language data sources. Future work must be done to evaluate if this system is able to recognize and normalize biomedical entities within Portuguese clinical notes. This presents an ambitious challenge as fewer sources of Portuguese biomedical data are available, and the controlled vocabularies are scarce.

Finally, the user interface should be upgraded encompassing more social and biomedical document providers. The information available for each recognized and normalized entity may also be upgraded by showing fine details about the given entity, such as the description, similar entities, etc. The crowd sourcing data retrieval system may also be upgraded by employing more advance techniques to decide when a manual annotation has enough confidence to be presented to the remaining users of the system.

Bibliography

[1] *UMLS Reference Manual*. National Library of Medicine (US), Bethesda (MD), September 2009.

[2] ISO/TR 20514. Health informatics – Electronic health record – Definition, scope and context. ISO, 2005.

[3] Chid Apte, Fred Damerau, and Sholom Weiss. Text Mining with Decision Trees and Decision Rules. In *Proceedings of the Conference on Automated Learning and Discorery, Workshop 6: Learning from Text and the Web, Workshop 6: Learning from Text and the Web*, 1998.

[4] Alan R Aronson and François-Michel Lang. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.

[5] Christopher M Bishop et al. *Pattern Recognition and Machine Learning*, volume 4. springer New York.

[6] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based N-gram Models of Natural Language. *Computational linguistics*, 18(4):467–479, 1992.

[7] Markus Bundschus, Mathaeus Dejori, Martin Stetter, Volker Tresp, and Hans-Peter Kriegel. Extraction of semantic biomedical relations from text using conditional random fields. *BMC bioinformatics*, 9(1):207, 2008.

[8] Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, and Jun'ichi Tsujii. Named entity recognition with multiple segment representations. *Information Processing & Management*, 49(4):954–965, 2013.

[9] Peter Christen. A Comparison of Personal Name Matching: Techniques and Practical Issues. In *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 290–294. IEEE, 2006.

[10] Gari D Clifford, Daniel J Scott, and Mauricio Villarroel. User guide and documentation for the MIMIC II database. *MIMIC-II database version 2.6*, 2009.

[11] Aaron M Cohen and William R Hersh. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71, 2005.

[12] K Bretonnel Cohen and Lawrence Hunter. Natural Language Processing and Systems Biology. In *Artificial Intelligence methods and tools for systems biology*, pages 147–173. Springer, 2004.

[13] Ronald Cornet and Nicolette de Keizer. Forty years of SNOMED: A literature review. *BMC Medical Informatics and Decision Making*, 8(Suppl 1:S2):1–6, 2008.

[14] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995.

[15] Steven Dickman. Tough mining. *PLoS biology*, 1(2):e48, 2003.

[16] Noémie Elhadad, Sameer Pradhan, Sharon Gorman, Suresh Manandhar, Wendy Chapman, and Guergana Savova. SemEval-2015 Task 14: Analysis of Clinical Text. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 303–310, Denver, Colorado, June 2015. Association for Computational Linguistics.

[17] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[18] E Garcia. Cosine similarity and term weight tutorial. *Information retrieval intelligence*, 2006.

[19] Tiago Grego and Francisco M Couto. Enhancement of chemical entity identification in text using semantic similarity validation. *PloS ONE*, 8(5):1–9, 2013.

[20] Tiago Grego, Francisco Pinto, and Francisco Couto. LASIGE: using Conditional Random Fields and ChEBI ontology. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 660–666, 2013.

[21] Tiago Daniel Pereira Grego. Identifying chemical entities on literature: a machine learning approach using dictionaries as domain knowledge. 2013.

[22] Thomas R Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[23] Thomas R Gruber. Toward Principles for the Design of Ontologies used for Knowledge Sharing. *International journal of human-computer studies*, 43(5):907–928, 1995.

[24] Vishal Gupta and Gurpreet S Lehal. A survey of Text Mining Techniques and Applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76, 2009.

[25] Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304, 2008.

[26] Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravicius, and Martin Duneld. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of Biomedical Semantics*, 5(6), 2014.

[27] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[28] Matthew A. Jaro. Probabilistic linkage of large public health data files. *Statistics in medicine*, 14(5-7):491–498, 1995.

[29] Jay J. Jiang and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[30] Dipak Kalra. Electronic health record standards. 2006.

[31] Liadh Kelly, Lorraine Goeuriot, Hanna Suominen, Tobias Schreck, et al. Overview of the ShARe/CLEF eHealth Evaluation Lab 2014. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 172–191. Springer, 2014.

[32] Grzegorz Kondrak. N-gram Similarity and Distance. In *Proceedings of the 12th International Conference String Processing and Information Retrieval*, pages 115–126, 2005.

[33] Aleksandar Kovačević, Azad Dehghan, Michele Filannino, John A Keane, and Goran Nenadic. Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives. *Journal of the American Medical Informatics Association*, 20(5):859–866, 2013.

[34] Vijay Krishnan and Vignesh Ganapathy. Named entity recognition. 2005.

[35] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 2001.

[36] André Leal, Diogo Gonçalves, Bruno Martins, and Francisco M Couto. ULisboa: Identification and Classification of Medical Concepts. August 2014.

[37] André Leal, Bruno Martins, and Francisco Couto. ULisboa: Recognition and Normalization of Medical Concepts. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 829–834, Denver, Colorado, June 2015. Association for Computational Linguistics.

[38] André Leal, Bruno Martins, and Francisco Couto. Recognition and Normalization of Biomedical Entities Based on Ontologies. In *Proceedings of Bioinformatics Open Days*, page 12, 2015.

[39] Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. DNorm: Disease Name Normalization with Pairwise Learning to Rank. *Bioinformatics*, 29(22):2909–2917, 2013.

[40] Robert Leaman, Graciela Gonzalez, et al. Banner: an executable survey of advances in biomedical named entity recognition. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 13, pages 652–663, 2008.

[41] Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings ICML*, volume 98, pages 296–304, 1998.

[42] Haibin Liu, Tom Christiansen, William A. Baumgartner Jr, and Karin Verspoor. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal Biomedical Semantics*, 3(3):17, 2012.

[43] Henry J Lowe and G Octo Barnett. Understanding and Using the Medical Subject Headings (MeSH) Vocabulary to Perform Literature Searches. *Jama*, 271(14):1103–1108, 1994.

[44] Michael MacCandless, Erik Hatcher, and Otis Gospodnetić. *Lucene in Action*. Manning Publications Company, 2010.

[45] Christopher Manning, Tim Grow, Teg Grenager, Jenny Finkel, and John Bauer. Stanford Tokenizer, 2010.

[46] Bridget T. McInnes, Ted Pedersen, and Serguei V.S. Pakhomov. UMLS-Interface and UMLS-Similarity: Open Source Software for Measuring Paths and Semantic Similarity. In *Proceedings of the 2009 Annual Symposium of the American Medical Association*, pages 431–435, 2009.

[47] U.S. National Library of Medicine. Fact Sheet - Medical Subject Headings (MeSH®), 2013.

[48] U.S. National Library of Medicine. Fact Sheet - MEDLINE, PubMed, and PMC (PubMed Central): How are they different?, May 2014.

[49] U.S. National Library of Medicine. Fact Sheet - PubMed®: MEDLINE® Retrieval on the World Wide Web, 2014.

[50] U.S. National Library of Medicine. Unified Medical Language System® (UMLS®): Statistics - 2014AB Release, 2014.

[51] U.S. National Library of Medicine. Fact Sheet - MEDLINE®, 2015.

[52] U.S. National Library of Medicine. Unified Medical Language System® (UMLS®): SNOMED Clinical Terms® (SNOMED CT®), 2015.

[53] Naoaki Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). *URL http://www. chokkan. org/software/crfsuite*, 2007.

[54] Arzucan Ozgür. *Supervised and unsupervised machine learning techniques for text document categorization*. PhD thesis, Bogaziçi University, 2004.

[55] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[56] Sameer Pradhan, Noémie Elhadad, Wendy Chapman, Suresh Manandhar, and Guergana Savova. SemEval-2014 Task 7: Analysis of Clinical Text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 54–62, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[57] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[58] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[59] Mohammed Saeed, C. Lieu, G. Raber, R.G. Mark, et al. MIMIC II: a Massive Temporal ICU Patient Database to Support Research in Intelligent Patient Monitoring. In *Computers in Cardiology, 2002*, pages 641–644. IEEE, 2002.

[60] Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, et al. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[61] David Sánchez, Montserrat Batet, and David Isern. Ontology-based information content computation. *Knowledge-Based Systems*, 24(2):297–303, 2011.

[62] Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, et al. 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, 2010.

[63] Nuno Seco, Tony Veale, and Jer Hayes. An Intrinsic Information Content Metric for Semantic Similarity in WordNet. In *Proceedings of ECAI*, volume 16, page 1089, 2004.

[64] Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, et al. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 212–231. Springer, 2013.

[65] Ah-Hwee Tan et al. Text mining: The state of the art and the challenges. In *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, volume 8, pages 65–70, 1999.

[66] Kristina Toutanova and Christopher D Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

[67] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, et al. Developing a Robust Part-of-Speech Tagger for Biomedical Text. In *Advances in informatics*, pages 382–392. Springer, 2005.

[68] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, 2010.

[69] Joseph Turian, Lev Ratinov, Yoshua Bengio, and Dan Roth. A preliminary evaluation of word representations for named-entity recognition. In *Proceedings of the NIPS-09 Workshop on Grammar Induction, Representation of Language and Language Learning*, pages 1–8, 2009.

[70] Yu Usami, Han-Cheol Cho, Naoaki Okazaki, and Jun'ichi Tsujii. Automatic Acquisition of Huge Training Data for Bio-Medical Named Entity Recognition. In *Proceedings of BioNLP 2011 Workshop*, pages 65–73. Association for Computational Linguistics, 2011.

[71] Jonathan J Webster and Chunyu Kit. Tokenization as the initial phase in NLP. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*, pages 1106–1110. Association for Computational Linguistics, 1992.

[72] William E Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research of the American Statistical Association*, pages 354–359, 1990.

[73] Ian H Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005.

[74] Pak Chung Wong, Paul Whitney, and Jim Thomas. Visualizing Association Rules for Text Mining. In *Proceedings of Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pages 120–123. IEEE, 1999.

[75] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098, 2013.

[76] Yaoyun Zhang, Jingqi Wang, Buzhou Tang, Yonghui Wu, et al. UTH_CCB: A report for SemEval 2014 – Task 7 Analysis of Clinical Text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 802–806, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[77] Zili Zhou, Yanna Wang, and Junzhong Gu. A New Model of Information Content for Semantic Similarity in WordNet. In *Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on*, volume 3, pages 85–89. IEEE, 2008.