

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



Identifying Human Phenotype Terms in Text Using a Machine Learning Approach

Manuel Stapleton Garcia De Vasconcelos Lobo

MESTRADO EM BIOINFORMÁTICA E BIOLOGIA COMPUTACIONAL
ESPECIALIZAÇÃO EM BIOINFORMÁTICA

Dissertação orientada por:
Francisco M. Couto

2017

Acknowledgements

I want to thank **Francisco M. Couto**, my teacher and supervisor, for bringing me into this project and giving me the support and the opportunity to work in such an interesting field.

I also want to thank **André Lamúrias**, the developer of IBEnt, who always offered a helping hand in the numerous times I needed help.

Finally, I want to thank my family for all the help and support they provided.

Resumo

Todos os dias, uma grande quantidade de informação biomédica está a ser criada sob a forma de artigos científicos, livros e imagens. Como a linguagem humana tem uma natureza não-estruturada (texto com baixo nível de organização), torna-se necessário a criação de métodos de extração de informação automáticos para que seja possível converter esta informação de modo a ser legível por uma máquina e para que seja possível automatizar este processo. Os sistemas de extração de informação têm melhorado ao longo dos anos, tornando-se cada vez mais eficazes. Esta informação extraída pode depois ser inserida em bases de dados para que seja facilmente acessível, pesquisável e para que seja possível criar ligações entre diferentes tipos de informação.

O Processamento de Linguagem Natural (PLN) é uma área da informática que lida com linguagem humana. O seu objetivo é extrair significado de texto não-estruturado, de forma automática, utilizando um computador. Utiliza um conjunto de técnicas como tokenization, stemming, lemmatization e part-of-speech tagging para desconstruir o texto e torna-lo legível para máquinas. O PLN tem várias aplicações, entre as quais podemos encontrar: coreference resolution, tradução automática, Reconhecimento de Entidades Mencionadas (REM) e part-of-speech tagging.

Os métodos de aprendizagem automática têm um papel muito importante na extração de informação, tendo sido desenvolvidos e melhorados ao longo dos anos, tornando-se cada vez mais poderosos. Estes métodos podem ser divididos em dois tipos: aprendizagem não-supervisionada e aprendizagem supervisionada. Os métodos de aprendizagem não-supervisionada como o Clustering, não necessitam de um conjunto de treino anotado, sendo isso vantajoso pois pode ser difícil de encontrar. Estes métodos podem ser usados para encontrar padrões nos dados, o que pode ser útil quando as características dos dados são desconhecidas. Por sua vez, os métodos de aprendizagem supervisionada utilizam um conjunto de treino anotado, que contém exemplos para os dados de input e de output, com o qual é possível criar um modelo capaz de classificar um conjunto de dados não anotado. Alguns dos métodos de aprendizagem supervisionada mais comuns são os Conditional Random Fields (CRFs), Support Vectors Machines (SVMs) e Decision Trees.

Os CRFs são utilizados nesta tese e são modelos probabilísticos geralmente usados em sistemas de REM. Estes modelos apresentam vantagens em relação a outros modelos, permitindo relaxar as hipóteses de independência que são postas aos Hidden Markov Models (HMM) e evitar os problemas de bias (preconceito) existentes nos SVMs.

O REM é um método que consiste na identificação de entidades em texto não-estruturado. Os sistemas REM podem ser divididos em três vertentes: métodos de aprendizagem automática, métodos baseados em dicionários e métodos baseados em regras escritas. Hoje em dia, a maioria dos sistemas de REM utilizam métodos de aprendizagem automática. As vertentes que utilizam apenas métodos de aprendizagem automática são flexíveis, mas precisam de grandes quantidades de dado, tendo a possibilidade de não produzir resultados precisos. Os métodos baseados em dicionários eliminam a necessidade de grandes quantidades de dados e conseguem obter bons resultados. No entanto, estes métodos são limitativos pois não conseguem identificar entidades que não estão dentro do dicionário. Finalmente, métodos que usam regras escritas podem produzir resultados de alta qualidade. Não tendo tantas limitações como os métodos baseados em dicionários, têm a desvantagem de ser necessário uma grande quantidade de tempo e trabalho manual para obter bons resultados.

O objetivo desta tese é o desenvolvimento de um sistema REM, o IHP (Identifying Human Phenotypes) para a identificação automática de entidades representadas na Human Phenotype Ontology (HPO). A

HPO é uma ontologia com o objetivo de fornecer um vocabulário standardizado para defeitos fenotípicos que podem ser encontrados em doenças humanas. O IHP utiliza métodos de aprendizagem automática para o processo de identificação de entidades e uma combinação de métodos baseados em dicionários e métodos baseados em regras escritas para o processo de validação das entidades identificadas.

O IHP utiliza duas ferramentas de benchmarking específicas para esta ontologia, apresentadas num trabalho anterior (Groza T, 2015): O Gold Standard Corpora (GSC), que consiste num conjunto de abstracts com as respetivas anotações de termos do HPO, e os Test Suites (TS), que consistem num conjunto de testes específicos divididos em categorias diferentes. Estas ferramentas têm o propósito de testar diferentes propriedades dos anotadores. Enquanto que o GSC testa os anotadores de uma forma geral, avaliando a capacidade de identificar entidades em texto livre, os TS são compostos por um conjunto de testes que avaliam as possíveis variações linguísticas que as entidades do HPO podem ter. Groza et al. também apresenta os resultados do anotador BioLark-CR, o qual é utilizado como baseline para os resultados do IHP.

O IHP utiliza o IBent (Identification of Biological Entities) como o sistema de REM base, tendo sido modificado para aceitar entidades do HPO. Este sistema usa o Stanford CoreNLP em conjunto com CRFs, sob a forma de StanfordNER e CRFSuite, de modo a criar um modelo a partir de um conjunto de treino. Este modelo pode depois ser avaliado por um conjunto de teste.

Para a criação de um modelo é necessário selecionar um conjunto de características (features) que se ajuste ao conjunto de dados utilizados. O StanfordNER e o CRFSuite apresentam conjuntos de features diferentes. Para o StanfordNER, uma lista de features existente foi utilizada, aplicando um algoritmo para selecionar as features que trazem maiores benefícios. Para o CRFSuite, foi criado um conjunto de features (linguísticas, morfológicas, ortográficas, léxicas, de contexto e outra) com base em trabalhos prévios na área do REM biomédico. Este conjunto de features foi testado e selecionado manualmente de acordo com o desempenho.

Além da utilização das features, um conjunto de regras de pós-processamento foi desenvolvido para pesquisar padrões linguísticos, utilizando também listas de palavras e stop words, com o propósito de remover entidades que tenham sido mal identificadas, identificar entidades que não tenham sido identificadas e combinar entidades adjacentes.

Os resultados para o IHP foram obtidos utilizando os classificadores StanfordNER e o CRFSuite. Para o StanfordNER, o IHP atinge um F-measure de 0.63498 no GSC e de 0.86916 nos TS. Para o CRFSuite, atinge um F-measure de 0.64009 no GSC e 0.89556 nos TS. Em relação ao anotador comparativo BioLarK CR, estes resultados mostram um aumento de desempenho no GSC, sugerindo que o IHP tem uma maior capacidade do que o BioLarK CR em lidar com situações reais. Apresenta, no entanto, um decréscimo nos TS, tendo uma menor capacidade em lidar com estruturas linguísticas complexas que possam ocorrer. No entanto, apesar de haver um decréscimo nos TS, as estruturas linguísticas avaliadas por estes testes ocorrem naturalmente em texto livre (como os abstracts do GSC), sugerindo que os resultados do GSC sejam mais significativos do que os resultados dos TS.

Durante o desenvolvimento da tese, alguns problemas foram identificados no GSC: anotação de entidades superclasse/subclasse, número de vezes que uma entidade é anotada erros comuns. Devido a estas inconsistências encontradas, o IHP tem o potencial de ter um desempenho melhor no GSC. Para testar esta possibilidade, foi efetuado um teste que consiste em remover Falsos Positivos que se encontram tanto nas anotações do GSC como também na base de dados do HPO. Estes Falsos Positivos, estando presentes no GSC e no HPO, provavelmente deveriam ser considerados como bem anotados, mas, no entanto, o GSC não identifica como uma entidade. Estes testes mostram que o IHP tem o

potencial de atingir um desempenho de 0.816, que corresponde a um aumento considerável de cerca de 0.18 em relação aos resultados obtidos.

Com a análise destas inconsistências encontradas no GSC, uma nova versão, o GSC+, foi criada. GSC+ permite uma anotação dos documentos mais consistente, tentando anotar o máximo número de entidades nos documentos. Em relação ao GSC, ao GSC+ foram adicionadas 881 entidades e foram modificadas 4 entidades. O desempenho do IHP no GSC+ é consideravelmente mais alta do que no GSC, tendo atingindo um valor de F-measure de 0.863. Esta diferença no desempenho é devido ao facto do GSC+ tentar identificar o máximo número de entidades possível. Muitas entidades que eram consideradas como erradas, agora são consideradas corretas.

Palavras Chave: **Reconhecimento de Entidades Mencionadas, Prospeção de texto, Aprendizagem Automática, Ontologias, Fenótipos**

Abstract

Named-Entity Recognition (NER) is an important Natural Language Processing task that can be used in Information Extraction systems to automatically identify and extract entities in unstructured text. NER is commonly used to identify biological entities such as proteins, genes and chemical compounds found in scientific articles. The Human Phenotype Ontology (HPO) is an ontology that provides a standardized vocabulary for phenotypic abnormalities found in human diseases. This article presents the Identifying Human Phenotypes (IHP) system, tuned to recognize HPO entities in unstructured text. IHP uses IBent (Identification of Biological Entities) as the base NER system. It uses Stanford CoreNLP for text processing and applies Conditional Random Fields (CRFs) for the identification of entities.

IHP uses of a rich feature set containing linguistic, orthographic, morphologic, lexical and context features created for the machine learning-based classifier. However, the main novelty of IHP is its validation step based on a set of carefully crafted hand-written rules, such as the negative connotation analysis, that combined with a dictionary are able to filter incorrectly identified entities, find missing entities and combine adjacent entities.

The performance of IHP was evaluated using the recently published HPO Gold Standardized Corpora (GSC) and Test Suites (TS), where the system Bio-LarK CR obtained the best F-measure of 0.56 and 0.95 in the GSC and TS, respectively. Using StanfordNER, IHP achieved an F-measure of 0.646 for the GSC and 0.869 for the TS. Using CRFSuite, it achieved an F-measure of 0.648 for the GSC and 0.895 for the TS.

Due to inconsistencies found in the GSC, an extended version of the GSC, the GSC+, was created, adding 881 entities and modifying 4 entities. IHP achieved an F-measure of 0.863 on GSC+. Both the GSC+ and the IHP system are publicly available at: <https://github.com/lasigeBioTM/IHP>.

Contents

1	Introduction	1
1.1	Motivation.....	1
1.2	Objective.....	2
1.3	Thesis Organization	3
2	Related Work.....	4
2.1	Natural Language Processing.....	4
2.2	Machine Learning.....	5
2.3	Named-Entity Recognition	6
2.3.1	Biomedical Named-Entity Recognition	7
2.3.2	Previous Machine Learning Systems and Feature Selection	7
2.3.3	Feature Selection for Named-Entity Recognition	9
2.3.4	Post-Processing	12
2.4	Human Phenotype Ontology	12
2.4.1	Benchmarking Tools	13
2.4.2	Results from other Annotators	14
2.5	IBEnt (Identification of Biological Entities).....	14
2.5.1	Stanford CoreNLP	14
2.5.2	StanfordNER.....	15
2.5.3	CRFSuite	15
2.5.4	Evaluation Methods	15
3	Identifying Human Phenotypes.....	17
3.1	Methods	17
3.1.1	Corpus Loading	18
3.1.2	Training.....	19
3.1.3	Testing.....	23
3.1.4	Validation	23
3.1.5	Evaluation.....	25
3.2	GSC+.....	25
4	Results and Discussion	27
4.1	Overall Performance	27
4.1.1	Gold Standard Corpora Performance.....	28
4.1.2	Test Suites Performance.....	28
4.2	Feature Performance	28

4.2.1	CRFSuite Features.....	29
4.2.2	StanfordNER Feature Selection	30
4.2.3	Overall feature performance	33
4.3	Validation Rules Performance	33
4.4	Problems Faced During the Annotation	35
4.4.1	Problems Faced with the Ontology	35
4.4.2	Inconsistencies in the Corpora	36
4.5	GSC+.....	38
5	Conclusions	39
5.1	Limitations and Range of Validity	39
5.2	Significance to the area	40
5.3	Contributions.....	40
5.4	Future Work.....	40
6	References.....	42

List of Tables

Table 2.1 Benchmark Performance of External Annotators. Performance of NCBO Annotator, OBO Annotator and Bio-LarK CR with the benchmarking tools: Gold Standard Corpora and Test Suites.	14
Table 3.1: Linguistic features used in the annotator	20
Table 3.2: Orthographic features used in the annotator	21
Table 3.3: Morphologic features used in the annotator.....	21
Table 3.4: Context features used in the annotator	22
Table 3.5: Lexical features used in the annotator	22
Table 3.6: Other type of features used in the annotator	22
Table 4.1: Comparative performance of IHP and Bio-LarK CR in the Gold Standard Corpora and Test Suites.....	28
Table 4.2: The performance of the different types of used features for CRFSuite.	29
Table 4.3: Performance of IHP's validation rules on the Gold Standard Corpora using CRFSuite.....	34
Table 4.4: Performance of IHP's validation rules on the Gold Standard Corpora using StanfordNER.	34
Table 4.5: Potential Performance of IHP in the Gold Standard Corpus	38
Table 4.6: Performance of IHP with the GSC+.	38

List of Figures

Figure 3.1: Layout of the annotation procedure	17
Figure 4.1 Iterative selection of StanfordNER's Boolean Features.....	31
Figure 4.2: StanfordNER's Boolean Features.....	31
Figure 4.3 Brown Clusters selection process.....	33

1 Introduction

1.1 Motivation

The quantity of scientific information that is created every day is enormous, and although some of this information can be found in the form of structured data, that can be easily read by computers, a great portion remains as unstructured text like scientific articles, which are only meant for human reading. This, of course, means that unless the information is analyzed by a human being, it would be meaningless. Due to the lack ability of a computer to understand texts written in a human language, Information Extraction (IE) is an area that has been developed to extract information from text. IE is capable of reducing the amount of human resources needed to extract that information, and, at the same time, automatize this process. This area has had considerable growth in the past years with the development of new technologies, techniques and approaches. One of the main aspects of this evolution is the increasing number of systems with the purpose of storing structured data. More and more databases and ontologies are being created for the purpose of organizing all the information that is created, making information more available and connected.

The Identification of Biological Entities (IBEnt)^[1] (Lamurias A., 2015) is one of these information extraction systems and it relies on Stanford CoreNLP (Manning, 2014), along with Conditional Random Fields (John D. Lafferty, 2001) in the form of StanfordNER (Jenny Rose Finkel, 2005) and CRFSuite (Okazaki, 2007), to perform Named-Entity Recognition (NER) tasks, with the purpose of identification of biological entities.

The development of information extraction systems becomes increasingly important to automatically organize information, especially since there are a number of ways to analyze the same piece of information, depending on the perspective of the analysis. For example, from a single scientific text that focuses on the study of a single disease, it is possible to extract information on the causes, affected subjects, involved proteins and chemical compounds, symptoms, etc. All of these could be inserted into a particular database.

The Human Phenotype Ontology (HPO) (Sebastian Köhler, 2017) has the purpose of providing a standardized vocabulary for phenotypic abnormalities that can be found in human diseases. Each HPO entity can be described as a phenotypic abnormality. The information found on this ontology can be useful in a lot of medical areas and it could facilitate the understanding of medical texts such as scientific articles and patient reports, and could even bring different insight for certain observations.

For the purpose of creating a standard to evaluate the performance of HPO annotators, Groza et al. (Groza T, 2015) created two benchmarking tools: the Gold Standard Corpora (GSC), which contains a group of abstracts with the respective exact matching annotation of HPO terms, and the Test Suites (TS), which consist of a group of tests to evaluate the linguistic portion of annotators. Groza et al. applied the benchmarking tools to several annotators, of which Bio-LarK CR, a NER system created by Bio-LarK, was the best performing annotator and is considered the baseline performance for this work.

^[1] <https://github.com/AndreLamurias/IBEnt>

1.2 Objective

It can be challenging to annotate free text with entities from the HPO because their nature. This ontology contains a variety of entities that span from simple to extremely complex entities, that range from 1 to 14 words and contain complex linguistic patterns. Since this ontology is relatively new, having been created in 2008^[2], there is not a lot of work done around it, compared to other ontologies (like the Gene Ontology created in 2000 (Ashburner, 2000)), which usually translates as a diminished ability to annotate entities from an ontology.

The objective of this work is to develop an automatic annotator for the HPO called IHP (Identifying Human Phenotypes), that achieves better results than the state-of-the-art HPO annotators (like the Bio-LarK CR annotator). In order to evaluate the IHP's performance, IHP relies on GSC for both training and testing, using a 10-fold cross-validation method, as well as a group of TS that contain specific tests to determine an annotator's strengths and weaknesses.

Although IHP uses IBent as the information extraction system, this thesis focuses on the selection of a particular feature set for the machine learning-based NER system and specific validation methods adapted for the identification of HPO entities. IHP uses a different feature set for StanfordNER and CRFSuite. StanfordNER has a list of existent features^[3], which were selected by trial-and-error, using a developed algorithm that selects the features that better fit the data. CRFSuite, on the other hand, uses IBent to create a set of handcrafted feature functions. A set of feature functions were created for the different types of features (linguistic, morphologic, orthographic, lexical, context and others) having all been manually tested. The validation stage uses a dictionary-based approach to filter out false positive entities, find missed entities and combine adjacent entities. Specific handwritten rules were used in combination with the dictionary-based approach in order to overcome the complex linguistic nature of HPO entities.

1.3 Results and contributions

IHP was evaluated using Bio-LarK's benchmarking tools, using both StanfordNER and CRFSuite due to similarity in performance. Bio-LarK CR is the annotator used as a baseline, and it achieved an F-measure of 0.56 for the GSC, and an F-measure of 0.95 for the TS. Using StanfordNER, IHP achieved an F-measure of 0.63498 for the GSC and 0.86916 for the TS. Using CRFSuite, it achieved an F-measure of 0.64009 for the GSC and 0.89556 for the TS. When compared to Bio-LarK CR, IHP shows an increase in performance in the GSC, which suggests that it performs better in real-world situations. IHP does, however, underperform in the TS which suggests that it cannot deal with complex linguistic structures as well as the comparative annotator. However, since complex linguistic structure should occur normally in the GSC (since they are actual abstracts), it suggests that the GSC results are more relevant.

During the development of this thesis, some problems were identified in the GSC: annotation of superclass/subclass and nested entities, number of times an entity is annotated and common mistakes. Considering these issues, IHP has the potential of achieving a better performance in the GSC. To test this possibility, a test was created which consists of removing false positives that can be found in the GSC annotations and in the HPO database. These false positives, being present in the GSC and the HPO, should be considered as HPO entities, however, the GSC doesn't consider them. These tests show

^[2] The Human Phenotype Ontology started in 2008 at the Charité, Berlin, Germany. The major contributors were Peter Robinson and Sebastian Köhler

^[3] <http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html>

that IHP has the potential of achieving an F-measure of 0.816, which corresponds to a considerable increase of around 0.18 in F-measure, compared to the obtained results.

With the analysis of these inconsistencies found in the GSC, a new version, the GSC+, was created, providing a more consistent annotation of the documents. GSC+ attempts to provide as many instances of HPO entities as possible, having added 881 new entities and modified 4 entities. IHP's performance on GSC+ is much higher than the performance on the previous GSC, having achieved an F-measure of 0.863. This difference in performance is due to IHP's attempt to identify as many entities as possible. Many entities that were considered false positives in the previous GSC are now considered as true positives in GSC+. Both the GSC+ and IHP source code are available at <https://github.com/lasigeBioTM/IHP>.

1.3 Thesis Organization

This thesis is organized as follows:

Chapter 2 presents an overview of the field of Natural Language Processing and Machine Learning, as well as some of techniques and methods in text mining. It will explore biomedical Named-Entity Recognition and the Human Phenotype Ontology, along with Groza et al. work (Groza T, 2015), exploring benchmarking tools and some linguistic issues faced in this ontology. Finally, it will explore the particular Named-Entity Recognition system used in this thesis, IBEnt (Identification of Biological Entities).

Chapter 3 presents the developed HPO Named-Entity Recognition, exploring the methods, techniques and software that were used to achieve these results, focusing on the selected feature set and post-processing rules.

Chapter 4 provides the results obtained using the developed system in both the Gold Standard Corpora and Test Suites, as well as the results from the feature selection process and contribution of hand-written rules.

Chapter 5 will discuss the results obtained in Chapter 4, the problems faced, surpassed and those that could not be surpassed, as well as what can be done in the future to improve.

2 Related Work

This chapter will start by providing an overview of Natural Language Processing, some Machine Learning techniques and Named-Entity Recognition (NER), followed by an analysis of commonly used features in biomedical NER tasks. After this, the HPO and Groza et al. work with this ontology is presented. Finally, an overview of IBEnt, the used information extraction system, is presented focusing on the techniques and tools that it uses to give a better understanding of their roles. It is important to highlight the use of IBEnt (the NER system), a specific set of carefully chosen features and the two HPO benchmarking tools created by Groza et al.

2.1 Natural Language Processing

Natural Language Processing (NLP) is an area of computer science that deals with human languages, with the final objective of having a computer derive meaning from unstructured text written and understood by humans. It focuses on a number of tasks, each with a specific setting and evaluation method, as well as a standard corpus to be evaluated on. Common tasks found in NLP include coreference resolution, machine translation, NER, Part-of-Speech (POS) tagging and parsing.

Human language is extremely complex and, for that reason, special algorithms need to be devised in order to perform tasks involving the human language. The main problem found during these tasks is ambiguity. There are different types of ambiguities to consider such as morphologic, syntactical or semantical, which are a result of the multiple alternative linguistic structures of the human language (Jurafsky, 2009).

Structured data (found in databases and ontologies) is information that has a high degree of organization that can be easily processed by data mining tools. Unstructured data (found in scientific articles and books), on the other hand, has a low degree of organization that does not allow it to be easily processed. Since text written in human languages is unstructured, there are several techniques that are applied to try to turn this text into structured data. Some of the important techniques that are commonly used include tokenization, stemming, lemmatization, POS tagging, parse trees and coreference resolution (Lamurias, 2014).

Tokenization is one of the most important techniques and is one of the first ones that should be applied. It is a process that consists in breaking down the input text into tokens so that they can be easily processed. There are different techniques depending on the type of text. For simple texts, the regular tokenization using whitespaces and punctuation can be enough. However, if the text contains uncommon entities like chemical entities or symbols, the tokenization process can become very difficult and specialized techniques need to be developed.

Stemming has the purpose of reducing the variability in natural language by normalizing the variations of the same concept and it consists in reducing a word to its base form by chopping off the ends of the word and often removing derivational affixes (Christopher D. Manning, 2008). **Lemmatization** has a similar purpose as stemming but it uses the context of the word, along with a vocabulary and morphologic analysis, to determine the canonical/dictionary form of the word, also known as the lemma (Christopher D. Manning, 2008).

Part-of-speech tagging is a classification task with the objective of categorizing each word according to its grammatical context, resolving existing ambiguities in words. Identifying the class of a word provides information about likely neighbor words, according to a particular syntactic structure. POS tags are useful features in NER and information extraction tasks, and they can even help with stemming

(Jurafsky, 2009). There are different POS tagsets available, being the Penn Treebank Tagset^[4] one of the most commonly used. Since words can have different meanings depending on the context, it is important to choose a tagset that works well in the specific field at hand.

Parse trees represent the syntactic structure of a sentence and similarly to POS tags, they can provide additional information about a word in a sentence. Parse trees are a compact representation of derivations, where several derivations may come from the same parse tree. In a parse tree there are several nodes, each of those containing a label. The top node of the tree is the root, and the bottom nodes are the leaves. From the parse tree, we determine the *yield* which is defined as the concatenation of the labels found on the leaves, from left to right (Plaisted, 2016).

Coreference resolution is a task used to determine which referring expressions are meant for which entities and avoid misclassification of an entity. A reference is defined as a linguistic expression used to denote an entity or an individual. Many times, we can use more than one referring expression for the same individual, and these expressions that refer to the same individual are said to corefer (Jurafsky, 2009).

These techniques and many more make natural language processing and information extraction possible.

2.2 Machine Learning

Machine learning consists in the development of algorithms that allow computers to perform automatic tasks. These can include a great variety of tasks such as classifying data after learning from a set of training data. There are different types of learning methods in machine learning and they can be divided into two main types: unsupervised learning and supervised learning.

Unsupervised learning has no need for labeled training data which can be very useful when it is difficult to come across enough data or when it requires too much manual work to generate the data. It does, however, require a large unlabeled training data for statistics. Unsupervised learning algorithms can be used to find patterns in data, which is especially useful when the characteristics of the data are unknown and need to be explored. An example of unsupervised learning is **Clustering**, where a set of feature functions define certain characteristics for each cluster, which allows the extraction of statistics from the unlabeled corpus.

Supervised learning requires the use of training sets that contain examples of the input data and expected output. The training process will generate a model that is able to classify new unlabeled data according to the generated model. There are different types of supervised learning algorithms used in the area of text mining, the same algorithms in which NER systems can rely on. Some of the most currently used supervised machine learning methods include Decision Trees, Support Vector Machines (SVMs) and CRFs (Jurafsky, 2009). Using these algorithms, models can be generated to be used in recognition systems.

Decision trees are popular inductive inference algorithms and they are used to represent learned functions that approximate discrete-value target functions. These trees classify instances using the nodes of the tree, from the top node (the root) to the bottom nodes (the leaves). Between the nodes there are branches that represent the possible values for the attribute of a certain instance. Instances are

^[4]https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

classified starting at the root where they are tested according to their attributes and descend down the tree to the branches where they pass the tests defined in each node (Mitchell, 1997). This automated group of tests allows a quick classification of instances and are a valuable algorithm in machine learning.

Support vector machines (SVMs) are one of the most successful classification methods in machine learning because they are easy to optimize and interpret, and they have a low computation cost. SVMs are computer algorithms that can assign labels to objects after training on a data set. They are algorithms used for the maximization of particular mathematical functions, according to a given data set (Noble, 2006). One of the main attractions of SVMs is the fact that it works well with non-linearly separable data by applying non-linear transforms to the data.

Conditional Random Fields are a type of model that use the Markov property which states that decisions about a certain state at a particular position in a sequence can depend only on a small local window (Jenny Rose Finkel, 2005). Biomedical NER uses traditional machine learning approaches which many times involves the introduction of features into sequential CRFs, having the ability of producing an output of annotated named entities. Since CRFs are used in IHP, a more detailed overview will be provided.

CRFs are a type of probabilistic model that is used for labeling sequential data, which can present advantages over Hidden Markov Models (HMMs) and SVMs. This is possible since CRFs can include a rich feature set that, given a sequence and the corresponding labels, can obtain the conditional probability and define a weight vector associated to a feature function (Hahn, 2016). CRFs have the advantage of being able to use any number of features functions to ask arbitrary questions about the input data (e.g. creating feature functions for suffixes and prefixes for the current word, as well as for the words around that word) (Li, 2003). CRFs can be used in situations like overlapping and non-independent features that HMMs cannot deal with. Due to the conditional nature of CRFs, they can relax the independence assumptions (unlike HMMs) and avoid label bias problems (unlike Maximum Entropy Markov Models) (Campos, 2012). Linear chain CRFs can be seen as an undirected graphical model version of HMMs (Zhu, 2007).

CRFs are used to calculate the conditional probabilities of values on designated output nodes, given the values on designated input nodes. CRFs, therefore, define the conditional probability of a state sequence (a label) given a certain input sequence. These models need to be trained on a training set. This training process consists on the adjustment of weights in order to maximize the conditional log-likelihood of a labeled sequence in the training set. After having created and trained a model, CRFs are able to label sequences of tokens with the most probable labels, according to that model (John D. Lafferty, 2001).

These algorithms all have different parameters that can be modified to optimize the performance of the system, however, these parameters must be carefully chosen in order to avoid overfitting. Overfitting occurs when the classifier fits the data too well. This happens when the parameters are not correctly chosen and, therefore, instead of only fitting the data, it also fits the noise in the data, which then leads to unexpected results in different data sets.

2.3 Named-Entity Recognition

This section will provide an overview of NER, Biomedical NER and some of the previous works conducted in this area. It will provide a detailed view of features and post-processing techniques used in these works.

Named-Entity Recognition is an extremely important task in information extraction systems. NER tasks consist of finding Named Entities in unstructured text and classifying them into specific

categories. NER systems can be used in a lot of different fields, being common in the biomedical field, where it can be used to recognize proteins, genes, drugs, chemical compounds, and many other types of entities. In general, NER systems identify entities in the text and return their position (the absolute character position of the first letter in the input text) and their offset (the number of characters in the entity). However, in some cases, only the information about the presence of the entity in a document is necessary.

Most of the current approaches to NER systems use machine learning. Pure machine learning approaches are very flexible, having the ability of identifying a variable and dynamic vocabulary of entities like proteins and genes (Campos, 2012). However, these approaches do not use entities from curated sources (like a dictionary) and may not produce very precise results. Usually, a large quantity of labeled data is required to yield good results.

In order to eliminate the need for large quantities of data, dictionary-based approaches can be used and still maintain good results. Dictionaries can be created from all types of sources (e.g. Wikipedia). Dictionary-based approaches are meant for strictly defined vocabulary entities like diseases or species (Campos, 2012) and usually yield high quality results. However, these approaches have some limitations since they cannot recognize entities that are not in the dictionary.

The use of hand-written rules and regular expressions can also be used to obtain high quality results, and they are not as limited as the dictionary approach. They have the advantage of identifying strongly defined orthographic and morphologic structures like chemical compounds (Campos, 2012). The disadvantage of hand-written rules and regular expressions is the amount of time and manual effort needed to accomplish good results (Lamurias, 2014).

It is possible to improve the overall performance of a NER system by using a combined approach of machine learning, dictionaries and hand-written rules.

2.3.1 Biomedical Named-Entity Recognition

In a NER system, it's important to know the type of entities that are being identified so that an appropriate feature set can be chosen in order to identify the correct patterns. Biomedical entities are difficult to identify in unstructured text because they don't follow particular nomenclatures, having a lot of variations. These entities are usually long, descriptive and often contain pre-modifiers (e.g. "normal thymic epithelia cells") (Campos, 2012), and can have several synonyms. They also have different capitalization, spelling, hyphenation and symbol patterns, as well as non-standard and highly ambiguous abbreviations (e.g. "TCF" may refer to "T cell factor" and "Tissue Culture Fluid") (Zhou, 2004). Finally, two or more entities can share a single head noun, making it hard to identify the several entities.

2.3.2 Previous Machine Learning Systems and Feature Selection

This section provides an overview over some previous works in NER which shows the benefits of using machine learning techniques in NER, as well as the influence of specific feature sets. The actual performance of the specific features will be discussed in the next section.

Zhou et al. (Zhou, 2004) developed a HMM-based NER system (PowerBioNE) that doesn't rely on the use of a dictionary but integrates a rich set of features which include word formation patterns, morphologic patterns (prefix and suffix), POS tagging, head noun triggers, special verb triggers and

name alias. The system was evaluated on GENIA^[5] 3.0 and GENIA 1.1, achieving an F-measure of 66.2 and 62.2, respectively. It achieves an F-measure of 75.8% for the protein class with GENIA 3.0.

Lee et al. (Lee, 2003) developed a two-phase (identification and classification) SVM-based NER system. A dictionary was constructed using the training corpus and is used in a post-processing step that follows the SVM classifier. The identification phase had an F-measure of 79.9% in the GENIA 3.0 corpus, using 10-fold cross-validation. The system used orthographic and context (range of 2) features, prefix, suffix and lexical of each token, POS tagging and “presence-in-dictionary” features.

Lin et al. (Lin, 2004) developed a Maximum Entropy-based machine learning system with dictionary-based and rule-based methods for post-processing. The system was tested on the GENIA 3.01 corpus using a 10-fold cross-validation and achieved an F-measure of 0.525 and 0.72, before and after post-processing, respectively. The post-processing stage was applied to extend partially identified entities. The system uses orthographic and morphologic features, head nouns and POS tagging.

Tsai et al. (Tsai, 2014) developed a Maximum Entropy-based NER system that uses a set of features: orthographic, context, POS tags, word shape and prefix/suffix features. The system is evaluated on the GENIA 3.02 corpus, achieving an F-measure of 70.0% without using a dictionary and achieved an F-measure of 74.3% for proteins.

Hahn et al. (Hahn, 2016) developed a CRFs-based NER system using CRFSuite (L-BFGS method) with focus on sentence and token-level features. It uses a conjunction of orthographic and contextual features, affixes, n-grams, POS tagging and word normalization. This system was evaluated on the NCBI Disease^[6] corpus using a 10-fold cross-validation technique, achieving an F-measure of 88%. The work focuses on the challenges of boundary detection and entity classification.

Campos et al. (Campos, 2012) developed two CRFs-based machine learning systems (BioEnEx and BANNER) that can identify disorder names, using linguistic, orthographic, morphologic, context and lexicon features. It used the Arizona Disease Corpus^[7] and BioText^[8] to train the system to identify disorder names. It achieved an F-measure of 81.08% in the Arizona Disease Corpus and 54.84% in BioText.

Tkachenko et al. (Tkachenko, 2012) developed a feature-rich CRFs-based NER system. The system achieves an F-measure of 91.02% on the CoNLL 2003 dataset^[9], 81.4% F-measure on the OntoNotes version 4 CNN Dataset^[10] and 74.27% on NLPBA 2004 dataset^[11]. The system explores the use of local knowledge features (features extracted from a single token), external knowledge features (POS tags, dictionaries), non-local dependencies of names entities and specific features that do not fall into previous categories.

^[5] <http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/>

^[6] <https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/>

^[7] http://diego.asu.edu/downloads/AZDCAnnotationGuidelines_v013.pdf

^[8] <http://biotext.berkeley.edu/>

^[9] <http://www.cnts.ua.ac.be/conll2003/ner/>

^[10] <http://www.aclweb.org/anthology/N09-4006>

^[11] <http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html>

2.3.3 Feature Selection for Named-Entity Recognition

Extraction of features from data is essential in machine learning, and selecting the best features is one of the great challenges in machine learning. These features represent properties in the data that make labels distinguishable from each other. The feature choice needs to be specific enough so that it fits the data but not too specific so that the features fit a large quantity of data (Lamurias, 2014). Careful feature selection is especially important in CRFs because these are extremely sensitive to the selected feature set.

Features are usually selected according to observational tests and then combined in certain user-defined patterns. Since the number of possible tests is enormous it can be computationally expensive, and, therefore, features that prove to be irrelevant should be removed (Li, 2003).

The next sections will present different types of features, along with their relevance in different NER systems. These features are divided into Linguistic, Orthographic, Morphologic, Context, Lexicon and other features, based on the feature classification of Campos et al. (Campos, 2012). Orthographic, morphologic and context features are used in almost every NER system.

2.3.3.1 Linguistic Features

These features represent the most basic feature, the token, and consider the variations found in that token due to the context. Linguistic features include normalization techniques like stemming and lemmatization, as well as POS tagging. Linguistic features can help in the recognition of highly variable and low standardized entity names (Campos, 2012). In the work of Hahn et al. (Hahn, 2016) the use of stemming features resulted in an F-measure increase of 0.21.

The quality of the used POS tagger is crucial for tasks such as NER and even high quality taggers can yield bad results. It is important to choose a tagger that works well in a particular field, because the same way languages have a different structure, different fields have different meanings for certain words. According to various authors (Hahn, 2016; Zhou, 2004; Tsai, 2014), POS tags are helpful for boundary detection and for this purpose, Tsai et al. uses POS tags with a window size of 5.

In the work of Zhou et al. (Zhou, 2004), the use of POS tags increased the F-measure by 0.228 (on top of word formation and morphologic features); in the work of Hahn et al. (Hahn, 2016), increased by 0.12 (on top of orthographic and word normalization features); in the work of Lee et al. (Lee, 2003), increased by 0.028; and in the work of Tkachenko et al. (Tkachenko, 2012) it increased by 0.01.

2.3.3.2 Orthographic Features

Orthographic features capture knowledge about the word formation. These include the presence of uppercase/lowercase characters, presence of symbols and counting the number of digits (Campos, 2012). Orthographic features as well as POS tagging are strongly related to the identification of an entity (Lee, 2003) and are generally used as the baseline features for NER systems, just like in the work of Zhou et al. (Zhou, 2004) and Hahn et al. (Hahn, 2016).

In the work of Lin et al. (Lin, 2004) and Tsai et al. (Tsai, 2014), the most useful orthographic features include 'Full Capitalization', 'Mixed Capitalization' and 'Initial Capitalization'; In the work of Hahn et al. (Hahn, 2016) Orthographic Features were used for effective for boundary detection, and these included 'Dash in the token', 'More than 2 dashes in the token', 'Full Capitalization', 'Mixed Capitalization', 'Initial Capitalization', 'No Capitalization', 'Both numeric and text characters',

‘Parenthesis in a Multi-word’, ‘Brackets in a token’, ‘Greek letters in a token’ and ‘Slash in a Multi-word’.

2.3.3.3 Morphologic Features

Morphologic features reflect common structures found in different tokens. Three types of morphologic features are usually considered: Affixes (suffixes and prefixes), which can be used to distinguish between certain types of entities (e.g. the suffix “ases” for proteins); character n-grams, which represent sub-sequences of a token (unigrams, bigrams and trigrams); and word shape patterns, which generate character sequences that reflect the organization nature of letters, digits and symbols in a token (Campos, 2012).

Since biomedical entities are long and descriptive, the use of different ranges may have a big impact on the NER system. Lin et al. (Lin, 2004) considered morphologic features with a range of at least three, while, Hahn et al. (Hahn, 2016) considers different possible prefixes and suffixes for a certain entity. To show this, an example given by Hahn et al. is provided: for the word “tumour”, the extracted prefixes are “t”, “tu”, “tum” and “tumo” and the extracted suffixes are “r”, “ur”, “our” and “mour”.

Affixes showed an increase in F-measure of 0.02 in the work of Hahn et al. (Hahn, 2016), 0.01 (0.004 for prefixes and 0.006 for suffixes) in the work of Lee et al. (Lee, 2003) and 0.027 In the work of Zhou et al. (Zhou, 2004). In the work of Tkachenko et al. (Tkachenko, 2012), the addition of affixes and word shape increases the F-measure by nearly 0.05 and 0.004 on the baseline performance (single token feature), respectively.

In the work of Hahn et al. (Hahn, 2016), unigrams and bigrams showed an increase of 0.05 in F-measure.

2.3.3.4 Context Features

CRFs-based systems tend to work poorly when using information only about the current token, and therefore, the tokens around the current token should be considered, using features that rely on the previous and next words, within a certain range (Tkachenko, 2012). Context features consist of grouping features of surrounding tokens, in order to reflect the local context of single tokens (Campos, 2012).

Tkachenko et al. (Tkachenko, 2012) shows the importance of a careful choice of context features, focusing on the range. The system was tested using only the previous and next tokens as well as the current token and there was an increase of about 0.04 in F-measure. However, the F-measure decreased by 0.005 when the range was increased by 1, showing the importance of taking the range into account. A sliding window of 3 tokens showed the best performance in the choice of range, increasing almost 0.05 over the baseline performance. Tkachenko et al. applied context features on top of morphologic features and also considered the word shape of neighboring tokens. This was able to increase the baseline performance by about 0.052.

Lin et al. (Lin, 2004) used the 960 most common unigram and bigram head nouns to improve the performance. Tsai et al. (Tsai, 2014) used with a window size of 5 for the context features which helped determine the category of the token.

Context features can have a positive impact on the performance of a NER system, having demonstrated an increase of 0.01 in F-measure in the work of Hahn et al. (Hahn, 2016) and an increase of 0.077 in F-

measure in the work of Zhou et al. (Zhou, 2004) in the form of head noun triggers. Zhou et al. obtained negative results when applying special verb triggers.

2.3.3.5 Lexicon Features

Lexicon features represent dictionary knowledge that can be added to the set of features and potentially improve the performance (Campos, 2012). The chosen dictionary is matched against words, resulting in tags that can be used as features. There are two types of dictionaries: dictionaries containing target entity names, using exact matching and dictionaries containing trigger names that indicate the presence of specific names in surrounding tokens.

Encyclopedic knowledge like dictionary is a simple method to recognize named entities in a phrase and it can greatly improve the performance. It is common to use these techniques in conjunction with machine learning approaches for better results. Look-up systems that use a large entity list work well when those entities are not ambiguous. This is shown by Tkachenko et al. (Tkachenko, 2012), in which using a dictionary without disambiguation caused a decrease of around 0.8 in F-measure while the use of the same dictionary with disambiguation increased the performance by around 0.16 in F-measure. These results are highly dependent on the quality of the dictionary that is used. Tkachenko et al. conducted a second test using a different dictionary which showed an increase in F-measure of 0.14 without disambiguation and even showed a decrease when the disambiguation was applied.

2.3.3.6 Other Types of Features

This section will present features that do not particularly fit in the division above. These features include document/corpus-level features, as well as Word Clustering in the form of Brown Clustering.

2.3.3.6.1 Document/Corpus-level Features

Zhang et al. (Li Zhang, 2004) explores a different approach to NER, which applies a “focused” NER approach where entities are defined as “focused” or not. Zhang et al. applies a different set of features that reflect the properties of individual important entities, either at document-level or corpus-level. These features include:

- *Entity Type*, considers a vector of existing classes;
- *'In Title'*, checks whether the entity appears in the title or not;
- *Entity Frequency*, corresponds to the number of times an entity occurs in the document. A higher frequency tends to mean that the entity has a greater importance;
- *Entity Distribution*, considers the appearance of the entity in sub-sections of a document. The more sections an entity appears in, the more likely it is to be important;
- *Entity Neighbor*, considers the context of the entity by counting the neighboring entity types. If there are several entities of the same type side-by-side, it is likely that it serves the purpose of enumeration and therefore the entities are irrelevant;
- *First Sentence Occurrence*, is the number of occurrences of the entity in the beginning of a paragraph;
- *Total Entity Count*, is the total number of entities in the document. It reflects the importance of an entity;
- *Corpus-Level Document Frequency*, considers the number of times an entity appears in a corpus. If the entity has low frequency in the corpus but high frequency in the document, then it is likely to be a focused entity.

2.3.3.6.2 Word Clustering

Word clustering can improve the quality of a NER system by including word representations as features. In this thesis, Brown Clustering is used since it is a common technique and, therefore, there are available algorithms to work with. Brown clustering has shown success in NER systems (Tkachenko, 2012). It relies on the fact that we can find similar words in similar contexts, although, to do this, a large unlabeled data set is necessary. There are two types of brown clustering algorithms: partition of the input into word clusters, in which similar words are clustered together (e.g. weekdays, months are in the same cluster), and hierarchical word clustering, in which words are clustered according to the mutual information of bigrams (Brown P.F., 1992).

Since bigram pairs have different levels of similarity, a distance can be defined. This distance is given by a bit string and it is assigned to each word on the vocabulary (words in the input text). These bit strings can then be used as features for a NER system to improve the performance. In the work of Tkachenko et al. (Tkachenko, 2012), brown clusters increased the baseline performance by 0.02.

2.3.4 Post-Processing

Post-processing is an important step in NER that can resolve some of the issues that the classifier could not face automatically. Post-processing rules can identify missed entities, remove misidentified entities, as well as combine adjacent entities. There are several post-processing techniques for achieving better results. Some of these techniques are discussed below.

Lin et al. (Lin, 2004) used a post-processing method that consisted on the detection of boundaries in partially recognized entities. For nested entities, a list was created for the leftmost context words of the entities in the training corpus and another list for the rightmost. The boundaries are extended depending if the entity is followed by another entity or if there is a valid POS tag. This boundary method showed an improvement of 0.177 in F-measure.

Zhou et al. (Zhou, 2004) presented a pattern based post-processing for cascaded entity name resolution which showed an increase of 0.039 in performance. This technique is based on six entity construction patterns:

- Entity + Head Noun (e.g. <PROTEIN> binding motif → <DNA>);
- Entity + Entity (e.g. <LIPID> <PROTEIN> → <PROTEIN>);
- Modifier + Entity (e.g. anti <Protein> → <Protein>);
- Entity + Word + Entity (e.g. <VIRUS> infected <MULTICELL> → <MULTICELL>);
- Modifier + entity + head noun;
- Entity + entity + head noun.

Campos et al. (Campos, 2012) uses parentheses processing and abbreviation resolution can be essential tasks and used independently of the entity type in hand.

2.4 Human Phenotype Ontology

This section will provide an overview of the HPO, presenting some of the issues in the NER process, as well as two benchmarking tools developed by Groza et al. (Groza T, 2015) that are used in this thesis

and which are meant to test the performance of HPO annotators. The results from three annotators (NCBO Annotator, OBO Annotator and Bio-LarK CR) are presented from the work of Groza et al.

The HPO has the purpose of providing a standardized vocabulary for phenotypic abnormalities that can be found in human diseases. Each HPO entity can be described as a phenotypic abnormality. The information found on this ontology can be useful in a lot of medical areas and it could facilitate the understanding of medical texts such as scientific articles.

Since HPO entities, like many biomedical entities, have specific characteristics which do not follow a particular nomenclature, they can be difficult to identify in free text. HPO entities:

- are long, ranging from 1 to 14 words;
- are highly descriptive;
- contain pre-modifiers
- contain several synonyms for words in an entity
- contain specific capitalization, spelling, hyphenation and symbol patterns
- contain ambiguous abbreviations
- share a single head noun with two or more entities

2.4.1 Benchmarking Tools

The two benchmarking tools developed by Groza et al. include the GSC and the TS.

The GSC is the first HPO-specific corpus, and was created with the purpose of being used as a standard for phenotype NER. It consists of 228 individually annotated abstracts with HPO entities, containing a total of 1933 annotations, which cover 460 unique HPO entities.

The TS consist of 32 different types of test cases, each type containing singular entity identification tests for that specific type, reaching a total of 2164 HPO entities. These tests are manually crafted by experts and cover a broad range of entity types, and are meant to be used as a standard manner to perform error analysis. The TS package offers a stratified approach to data sampling that is based on several different criteria, creating a framework that is able to characterize the strength of the linguistic patterns used in each of the entity recognition systems. Each of these criteria is focused on a set of entities that have a particular property in common such as length in tokens, presence of punctuation and coordination. On average, for each criteria there are around 70 test cases and they can be grouped into different categories:

- *Length-based tests*, characterize the ability of a recognition system to identify entities of different word lengths. HPO terms range from 1 to 14 words.
- *Tests accounting for the presence of certain types of tokens*, include tests for punctuation, isolated numerals (Arabic, Roman) and stop words (IN, OF, TO, BY, FROM, WITH)
- *Lexical variation tests*, test the ability to deal with transformations from singular to plural or from nouns to adjectives and vice versa
- *Token ordering tests*, test the ability to deal with transformations that oppose canonical and transformed canonical ordering
- *Synonym tests*, include tests where ontology synonyms are replaced with the original entity labels
- *Other specialized tests*, include tests for non-English canonical structures, metaphoric constructs and composite terms created by the conjunction of several atomic entities.

2.4.2 Results from other Annotators

With these benchmarking tools, Groza et al. tested Bio-LarK's annotator, Bio-LarK CR, as well as two other annotators: NCBO annotator and OBO annotator. For both the GSC and the TS, the alignment strategy used was exact matching, which is the default strategy. Each case in the TS were treated as free text, and were used as the input for the recognition systems. The used evaluation metrics are Precision, Recall and F-score. The performance of the three annotators in the work of Groza et al. are presented in [Table 2.1](#):

Table 2.1 Benchmark Performance of External Annotators. Performance of NCBO Annotator, OBO Annotator and Bio-LarK CR with the benchmarking tools: Gold Standard Corpora and Test Suites.

	Gold Standard Corpora			Test Suites		
	Precision	Recall	F1	Precision	Recall	F1
NCBO Annotator	0.54	0.39	0.45	0.95	0.84	0.89
OBO Annotator	0.69	0.44	0.54	0.54	0.26	0.35
Bio-LarK CR	0.65	0.49	0.56	0.97	0.93	0.95

As presented in [Table 2.1](#), Bio-LarK CR is the best performing entity recognition system, using both the Gold Standard Corpora and the Test Suites, with an F-measure of 0.56 and 0.95, respectively.

2.5 IBent (Identification of Biological Entities)

This section will provide a look on the NER system used in this thesis, IBent, along with the tools and software incorporated into IBent.

IBent is a NER system developed by André Lamurias ([Lamurias A., 2015](#)). It relies on Stanford CoreNLP and CRFs in the form of StanfordNER or CRFSuite and has the purpose of identifying different types of biological entities. IBent's functioning can be divided into 5 main stages: Corpus loading, Training, Testing, Validation and Evaluation.

The sections below will give an overview of Stanford CoreNLP, StanfordNER, CRFSuite and the used evaluation methods.

2.5.1 Stanford CoreNLP

Natural Language Processing is a field that is in constant development and certain groups have developed specialized software meant to perform NLP tasks. One of the most popular tools for NLP tasks is Stanford's CoreNLP ([Manning, 2014](#)), which contains a set of packages that are able to perform different number of task such as POS tagging, NER, coreference resolution and many more.

Stanford's CoreNLP tools come in the form of annotators which hold the analysis information for a particular text, stored as a heterogeneous map. Annotators can be used together in a pipeline and their

behavior can be controlled by certain properties (Manning, 2014). The following annotators are used by IBEnt to process and enrich the text, in order to improve the quality of the recognition system:

- *tokenize*, tokenizes text into a sequence of tokens, recording their exact matching offset
- *ssplit*, splits a sequence of tokens into sentences
- *pos*, labels tokens with their POS tag
- *lemma*, generates lemmas for the tokens
- *ner*, recognizes named and numerical entities (makes NER possible using StanfordNER)
- *parse*, provides full syntactic analysis

2.5.2 StanfordNER

StanfordNER is a NER that is able to label sequences of words in a text in specific classes. It provides a general implementation of linear chain CRFs sequence models, which allows training of a model using labeled data. It also provides a big quantity of developed features, making it easier to select the best features for a particular recognition system.

2.5.3 CRFSuite

CRFSuite^[12], just like StanfordNER, implements CRFs for labeling sequential data (Okazaki, 2007). CRFSuite offers different training methods like Gradient descent using L-BFG (default), stochastic gradient descent, average perceptron and others. According to Li et al. (Li, 2003), methods like L-BFGS are more efficient than traditional iterative gradient and it can be simply treated as a black-box optimization procedure that only requires the first-derivative of the function for optimization.

2.5.4 Evaluation Methods

A Gold Standard is a corpus that is annotated with every entity of interest and corresponding position. It is then used as an unlabeled corpus to evaluate a NER system, checking which entities were correctly identified. To evaluate a recognition system, the positive and negative results need to be defined. A positive result, a True Positive (TP), is considered when an entity is correctly identified according to the Gold Standard, while a negative result is considered either as a False Positive (FP) when an entity is incorrectly identified, or as a False Negative (FN) when an entity in the Gold Standard is not found.

Since data sets have different sizes and different number of entities, a relative measure has to be defined in order to compare results obtained in the different cases. In most systems, three measures are used to evaluate the system's performance: Precision, Recall and F-measure.

Precision is the fraction of positive results that were correctly classified:

$$Precision = \frac{TP}{TP+FP} \quad (2.1)$$

^[12] <http://www.chokkan.org/software/crfsuite/>

Recall represents how often the results are correct, measuring how many positive results were identified relative to the total number of positive results in the Gold Standard:

$$Recall = \frac{TP}{TP+FN} \quad (2.2)$$

The F-measure is often used to express the performance of extraction systems with a single value, combining the results of the precision and recall. It is given by the harmonic mean between the precision and recall:

$$F - measure = \frac{2 \times precision \times recall}{Precision + Recall} \quad (2.3)$$

3 Identifying Human Phenotypes

This chapter will focus on the HPO annotator I developed, Identifying Human Phenotypes (IHP). It will explore the incorporation of the HPO and benchmarking tools into IHP and will focus on the description of the selected feature set, as well as the validation techniques that were used to improve the performance.

IHP is a NER system tuned to recognize HPO entities in unstructured text. IHP uses Stanford CoreNLP for text processing and applies Conditional Random Fields trained with a rich feature set containing linguistic, orthographic, morphologic, lexical and context features created for the machine learning-based classifier. However, the main aspect of IHP is the validation step based on a set of carefully crafted hand-written rules that combined with a dictionary can filter incorrectly identified entities, find missed entities and combine adjacent entities.

3.1 Methods

The figure below shows a diagram of IHP's annotation process, using IBEnt as the base NER system.

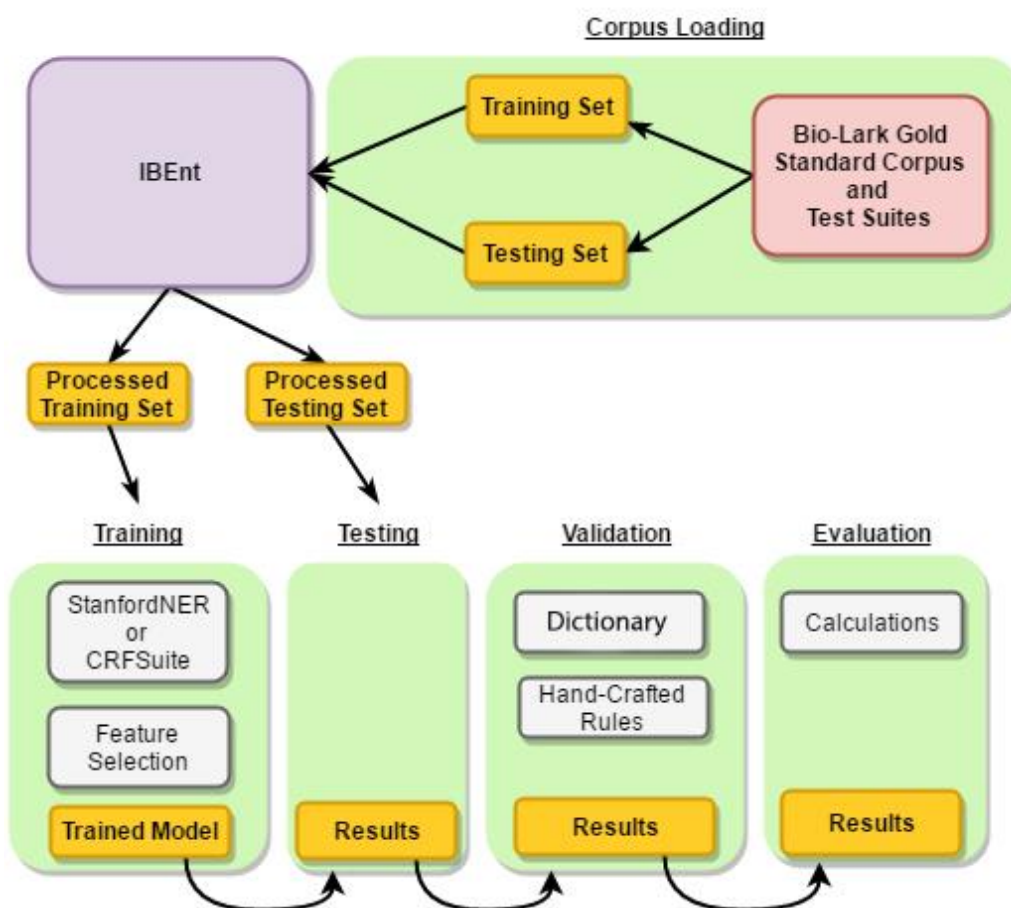


Figure 3.1: Layout of the annotation procedure. The procedure consists of 5 stages: Loading corpus, Training, Testing, Validation and Evaluation.

As shown in **Figure 3.1**, the annotation process can be divided into five main stages: Corpus loading, Training, Testing, Validation (Post-processing) and Evaluation. During the corpus loading stage, a corpus needs to be loaded into IHP so that it can be used to create a model. A second corpus needs to be loaded as the testing set so that the model can be tested. After the training/testing stage, there is a validation stage, in which a combination of a dictionary and handwritten rules will remove false positives, identify missed entities and combine adjacent entities. In the final stage (evaluation stage), the results are calculated, returning the precision and recall of the annotation process.

In the following section, the five stages are going to be presented. In each of these sections I will show the work I have done in detail.

3.1.1 Corpus Loading

IBEnt has the ability to accept different types of corpora, however, a reader must be created in order to read a particular corpus. The created reader has to match the organization and format of the available files. I created a reader for both GSC^[13] and TS^[14].

3.1.1.1 Gold Standard Corpora

For each document file in the GSC, there is a corresponding annotation file. Each document file consists of one line without a title and each annotation file contains as many lines as the number of annotated entities. The annotation is given by three columns: the exact matching offset, the HPO accession number and the annotation text. An example of a few lines is presented below:

[27::42]	HP_0000110 renal dysplasia
[171::183]	HP_0000365 hearing loss
[353::368]	HP_0000110 renal dysplasia
[375::393]	HP_0000006 autosomal dominant
[442::459]	HP_0004467 preauricular pits

Each of these documents is processed by IBEnt, being first divided into multiple lines using a specific algorithm for processing biomedical text, the GeniaSS (GENIA Sentence Splitter)^[15] algorithm, and then divided into tokens using Stanford CoreNLP Tokenizer annotator^[16]. Since the annotations are provided with the exact matching, the system can identify the tokens that correspond to the entities. In the end, the processed text and annotations are saved in a Stanford CoreNLP format to be used for training and testing.

^[13] https://github.com/lasigeBioTM/IHP/tree/master/src/reader/hpo_corpus.py

^[14] https://github.com/lasigeBioTM/IHP/tree/master/src/reader/test_suite.py

^[15] <http://www.nactem.ac.uk/y-matsu/geniass/>

^[16] <http://stanfordnlp.github.io/CoreNLP/tokenize.html>

3.1.1.2 Test Suites

Test Suites are the other benchmarking tool created by Bio-LarK in order to provide a standard manner to evaluate the performance of HPO annotators. Since, just like for the GSC, a reader is necessary, I also created a reader for the TS. Each TS file contains a series of tests that span the different categories explored in section 2.4, followed by the entities to be identified in that category. An example of a few lines are presented below:

```
# Lenght: 9
- HP:0003248=Gonadal tissue inappropriate for external genitalia or chromosomal sex
- HP:0003513=Reduced ratio of renal calcium clearance to creatinine clearance

# Containing punctuation
- HP:0002050=Macroorchidism, postpubertal
- HP:0008730=Female external genitalia in individual with 46,XY karyotype
- HP:0012321=D-2-hydroxyglutaric aciduria
- HP:0005575=Hemolytic-uremic syndrome
- HP:0003126=Low-molecular-weight proteinuria

# Containing stop words: OF
- HP:0008697=Hypoplasia of the fallopian tube
- HP:0000079=Abnormality of the urinary system
- HP:0008651=Uric acid urolithiasis independent of gout
```

For each of the files, only the text of each entity is loaded into IBEnt for testing. Since there is no exact matching information, the offset was calculated for each entity so that the system understands when an entity is correctly identified.

3.1.2 Training

During the training stage, a model is trained using the GSC. For the creation of the model, a classifier and a list of features have to be chosen. IBEnt offers the option of using either StanfordNER or CRFSuite as a classifier to train a model. These two classifiers have their own set of features to choose from. Both of these classifiers were used for the annotation process since they achieved similar results.

A 10-fold validation technique was applied for the GSC, since it has been proven to provide a good estimative of the quality of a system (Mohri, 2012). I created a python script^[17] to apply the cross-validation. It divides the documents randomly into ten groups, loading nine of these into IBEnt to be trained and create a model using either StanfordNER or CRFSuite, and loading the remaining group to be tested on the created model. This process is repeated ten times for each part. The precision and recall values are recorded and in the end the average of these values is obtained.

^[17] https://github.com/lasigeBioTM/IHP/tree/master/src/other/cross_validation.py

For the TS, a model is created using StanfordNER or CRFSuite and training with the whole GSC. The TS are then tested on the created model.

3.1.2.1 StanfordNER Features

StanfordNER has an available set of prepared features to choose from. I divided these features into Boolean features (features that can be set as True or False) and numerical features (features that can be assigned different numerical values). Then I created a python script^[18] to determine which features better fit the corpus. The script iterates over the several StanfordNER features and sets different values, using a greedy approach (gives priority to speed instead of performance) to accumulate features that improve the results. For the Boolean features, the script iterates these features, activating one feature at the time and checking if there is an improvement in the performance. For each of the numerical features, the default value (defined in the list of features) was used as a base value to create around 10 different value tests. The numerical features were tested after the Boolean features were chosen.

Since this consumes a lot of processing power and time to test in the best conditions, I tested the features on a single cross-validation iteration, meaning that the same training and testing corpuses were used for testing all of the features. This might not provide the best results and can even create some sampling bias and overfitting, but it is enough to understand which features have a positive impact on the performance.

3.1.2.2 CRFSuite Features

CRFSuite has a set of options to choose from^[19]. These options were tested manually to achieve the best results for the classifier. CRFSuite was applied with a l2sgd algorithm (Stochastic Gradient Descent with L2 regularization term), with the following settings:

- **L1 Coefficient** - '0.9833'
- **L2 Coefficient** - '1'
- **feature.possible_states** - '1'
- **feature.possible_transitions** - '1'

Apart from these options, CRFSuite also relies on a set of features that can be defined using feature functions in IBEnt. Some features like POS tagging, lemmatization and affixes were already defined in IBEnt. I created a new set of feature functions which include linguistic, orthographic morphologic, context, lexicon and other features, that were selected based on the works of (Zhou, 2004; Lee, 2003; Lin, 2004; Tsai, 2014; Hahn, 2016; Campos, 2012; Tkachenko, 2012), as well as based on their contribution in the identification of HPO entities. The used features are going to be detailed in the next section, along with a discription and an example of how the feature works.

Table 3.1: Linguistic features used in the annotator

<i>Linguistic</i>	Description	Example
<hr/>		

^[18] https://github.com/lasigeBioTM/IHP/tree/master/src/other/feature_selection.py

^[19] <http://www.chokkan.org/software/crfsuite/manual.html>

<i>Lemma</i>	The lemma of current token	"abnormalities" -> "abnorm"
<i>Postag</i>	The POS tag of current token	"abnormalities" -> NNS

Table 3.2: Orthographic features used in the annotator

Orthographic	Description	Example
<i>Word Case</i>	Classification of tokens according to case	"LOWERCASE", UPPERCASE"
<i>Digit</i>	Presence of a digit in the token	"xxx12"
<i>Bracket (Left)</i>	Presence of a left bracket in the token	"[" "[x"
<i>Bracket (Right)</i>	Presence of a right bracket in the token	"]" "x]"
<i>Slash</i>	Presence of a slash in the token	"/" "x/" "/x"
<i>Dash</i>	Presence of a dash in the token	"-" "x-" "-x"
<i>Quote</i>	Presence of a quote in the token	" ' " " 'x " " 'x "
<i>Double-quote</i>	Presence of a double-quote in the token	" " " 'x ' ' x " "
<i>Parenthesis (Left)</i>	Presence of a left parenthesis in the token	"(" "(x"
<i>Parenthesis (Right)</i>	Presence of a right parenthesis in the token	")" "x)"

Table 3.3: Morphologic features used in the annotator

Morphologic	Description	Example
<i>Prefix-2</i>	Prefix with length of 2	t[:2] "tumour" -> "tu"
<i>Prefix-3</i>	Prefix with length of 3	t[:3] "tumour" -> "tum"
<i>Suffix-1</i>	Suffix with length of 1	t[-1:] "tumour" -> "r"
<i>Suffix-2</i>	Suffix with length of 2	t[-2:] "tumour" -> "ur"
<i>Suffix-3</i>	Suffix with length of 3	t[-3:] "tumour" -> "our"
<i>Suffix-4</i>	Suffix with length of 4	t[-4:] "tumour" -> "mour"
<i>Word Shape</i>	Representation of a token	"SYM1-like" -> "XXXD-xxxx"
<i>Bigram</i>	Token and previous token	t-1/t

Table 3.4: Context features used in the annotator

Context	Description	Example
<i>Lemmas</i>	Lemma with range of 2	l-2, l-1, l, l+1, l+2
<i>Part-of-Speech</i>	POS tags with range of 4	p-4, p-3, p-2, p-1, p, p+1, p+2, p+3, p+4
<i>Word Shape</i>	Word Shape with range of 2	s-2, s-1, s, s+1, s+2
<i>Prefixes-1</i>	Prefix with length 1 with range of 1	t-1[:1], t[:1], t+1[:1]
<i>Prefixes-2</i>	Prefix with length 2 with range of 1	t-1[:2], t[:2], t+1[:2]
<i>Prefixes-3</i>	Prefix with length 3 with range of 1	t-1[:3], t[:3], t+1[:3]
<i>Prefixes-4</i>	Prefix with length 4 with range of 1	t-1[:4], t[:4], t+1[:4]
<i>Suffixes-1</i>	Suffix with length 1 with range of 1	t-1[-1:], t[-1:], t+1[-1:]
<i>Suffixes-2</i>	Suffix with length 2 with range of 1	t-1[-2:], t[-2:], t+1[-2:]
<i>Suffixes-3</i>	Suffix with length 3 with range of 1	t-1[-3:], t[-3:], t+1[-3:]
<i>Suffixes-4</i>	Suffix with length 4 with range of 1	t-1[-4:], t[-4:], t+1[-4:]

Table 3.5: Lexical features used in the annotator

Lexical	Description	
<i>stopwords</i>	Presence of stop words with a range of 4	sw-4, sw-3, sw-2, sw-1, sw, sw+1, sw+2, sw+3, sw+4

Table 3.6: Other type of features used in the annotator

Other	Description	Example
<i>brown_cluster</i>	Brown Clustering representation of words	"tumour" -> 00010011
<i>Length</i>	Classification according to number of characters	"tum" -> A; "tumor" -> B ; "Tumorous" -> C

3.1.2.3 Brown Clusters

For the implementation of Brown clustering, I used one of the most popular brown clustering algorithms, the Liang's Brown clustering algorithm^[20]. This algorithm creates word clusters from an input text divided into separate sentences. I processed the whole GSC and divided it into sentences using the GeniaSS algorithm.

The Brown clustering algorithm allows different options for the creation of the clusters, which include the 'number of clusters', 'number of collocations with most mutual information', 'maximum length of a phrase to consider' and 'minimal number of occurrences of a phrase'. Since HPO terms can range from 1 to 14 words, the 'maximum length of a phrase to consider' parameter was excluded.

I created a python script^[21] in order to test the variation of the different parameters. I tested the 'number of clusters' with values between 50 and 500, the 'number of collocations with most mutual information' with values between 50 and 750, and the 'minimal number of occurrences of a phrase' with values between 1 and 10.

After selecting the best combination of parameters, the output file can be inserted into both StanfordNER and CRFSuite's features. StanfordNER already comes prepared with a specific feature for distance similarity measures (such as Brown clusters) and IBEnt's feature extractor allowed the creation of a feature function for Brown clusters.

3.1.3 Testing

During the testing stage, the loaded corpus for testing is tested using the model created during the training phase. IHP will annotate the text according to the model and save the results so that they can be validated and evaluated.

3.1.4 Validation

During the validation stage, IHP will remove false positives, identify missed entities and combine adjacent entities. The process consists of revaluation of every sentence using a combination of the hand-written rules and a dictionary containing HPO terms.

3.1.4.1 Dictionary

I built the dictionary using all the terms and term synonyms from the HPO database^[22], as well as the training set annotations (different in each cross-validation iteration). To try to identify more entities, each entity that has the form "abnormalities of the kidney" is converted to "kidney abnormalities", and vice-versa. All the entities are then processed into a text file containing one entity per line, to be used along with hand-written rules.

^[20] <https://github.com/percyliang/brown-cluster>

^[21] <https://github.com/Hellsbath/IBEnt/blob/master/src/other/features/tests.py>

^[22] <http://compbio.charite.de/jenkins/job/hpo.annotations.monthly/lastStableBuild/> (downloaded on 25/January/2016)

3.1.4.2 Hand-written Rules

The validation process is extremely important for NER systems and one of the most important processes for IHP. For this thesis, a combination of handwritten rules, a dictionary and lists of words²³ (e.g. stop words, common words) was used in order to address some of the issues created by the machine learning classifier. This process is able to remove false positives, identify missed entities and combine adjacent entities. I created a script that receives the results from the machine learning-based annotator and applies different validation rules.

Handwritten rules can also play a big role in the identification of entities in a NER system. These rules have the advantage of being highly specific to the training set in hand and they can take advantage of specific structural patterns found in the text. These rules help find entity boundaries making it possible to identify bigger and smaller entities. The hand-written rules I developed for this thesis try to take advantage of the HPO entities in the dictionary, as well as the entities that were previously identified by the machine learning-based annotator. The developed handwritten rules give priority to the recall and therefore to the identification of a bigger number of GSC terms. The validation rules developed for IHP can roughly be divided into two categories: identification of entities and removal of entities. These rules start by adding as many entities as possible, adding word phrases that match a certain structural pattern. This can have a negative impact on the precision since not all word phrases that have a certain structural pattern found in entities, are actually entities. To remove the misidentified entities, and therefore, improve the precision of IHP, a group of rules are applied to remove entities with specific patterns. The two types of rules will now be presented.

3.1.4.2.1 Identification of entities

- **Dictionary entities** - The most basic rule consists on the identification of entities that exist in the dictionary, using exact matching.
- **Entity Variations** - This technique finds specific entity structures. Starts by considering a set of nouns commonly present in HPO entities (e.g. “abnormalities”, “malformations”) and possible variations of the next tokens in the sentence (e.g “of”, “of the” and “in the”). Using these structures, it then tries to match nouns (e.g. “abnormalities of the ear”) or a group of adjectives (e.g. “defects of the outer, middle and inner ear”).
- **Longer Entities** - It works similarly to the previous rule, however, instead of finding structures in the sentence, it uses entities from the results set and the dictionary as the base point. After finding these entities in the sentence, it tries to expand the entity boundaries (to the left or to the right) by identifying certain words and POS tags. For example, if “rib anomalies” was previously identified, it would identify “spine and rib anomalies” by expanding to the right, identifying of the word “and” and the noun “spine”.

Smaller Entities - This technique for identifying smaller entities checks if an entity can be separated into more entities by searching for specific words and commas (e.g. identification of the entity “pits of the palms” inside “pits of the palms and soles”).

Second Validation - A second validation process is performed, providing the list of previously identified entities. Providing this list allows the rules to perform on a bigger number of entities leading to the identification of a bigger number of entities.

^[23] The various lists of words used during the validation can be found in:
<https://github.com/Hellsbath/IBEnt/blob/master/src/other/dictionary.py>

3.1.4.2.2 Removal of Entities

- **General Errors** - There are several techniques used to remove misidentified entities. General rules are used to remove obvious mistakes such as entities that are formed only by digits, entities that contain only a single quote/parenthesis, entities that are smaller than a total character length of 3 and entities that contain more than one specific type of noun like “abnormalities” or “malformations”. The latter example is an unlikely scenario because an entity should not have more than one of these specific nouns.
- **Incorrect Structure** - Using a combination of POS tags of the last tokens of an entity, it is possible to determine if it has a correct syntactical structure, removing entities with unlikely scenarios such as entities ending in commas, dots, prepositions and determiners.
- **Negative Connotation Analysis** - This is similar to the NLP task of Sentiment Analysis to determine the polarity of subjects in a set of documents. Since HPO entities refer to diseases and irregularities, they have a negative connotation. It is therefore possible to judge if an entity is an HPO entity by the connotation of the words used. For words like “development”, used in cases like “cognitive development”, this could only become a negative entity by the addition of a word such as “impairment” (“cognitive development impairment”). For this reason, a simple rule removes entities that follow two conditions: the entity has 2 tokens and the entity contains a noun with positive connotation. Following these conditions, entities with less than 3 tokens would never have a negative connotation and can therefore be removed. With more practical experience, it would be possible to improve this technique so that it could determine the connotation for bigger entities and possibly improving the performance of the annotator.
- **Stop Words** - Stop words are an integral part of most NER systems aiding in the removal of unwanted words. Two types of stop word lists²⁴ are used with different levels of exclusion. One list contains word phrases that remove entities using exact matching and the other list contains word phrases that remove entities that contain that word phrase in any part of the entity.

3.1.5 Evaluation

During the evaluation stage, IHP receives the results from the testing and validation stage and calculates the precision and recall values for the particular testing set. The calculation mechanism is already built into IBEnt and no changes had to be done. The evaluation also provides further information on the performance of the system such as the most common false positives and false negatives.

3.2 GSC+

As it will be discussed in detail in the next section, some inconsistencies have been found in the GSC. These inconsistencies can cause confusion to annotators and may lead to uncertain results. IHP was developed in a way that tries to identify all instances of HPO entities (normal, nested and subclass/superclass entities), independently of the number of times they appear in the text. Having the correct number of times entities appear in a document can be useful for calculating important values such as the Term Frequency, which is used to determine the importance of a term in a document. Since IHP tries to annotate as many entities as possible, if the GSC does not have a consistent annotation, IHP

²⁴ https://github.com/lasigeBioTM/IHP/tree/master/src/other/word_lists

will identify a lot of entities that are not in the GSC. This, of course, will cause a decrease in precision and therefore in the overall performance.

Since the inconsistencies found in the GSC may have a negative impact on the performance of HPO annotators, the GSC+ was created, providing a more consistent annotation of documents. The GSC+ was created by adding new instances of HPO entities that were automatically identified by IHP. The GSC+ attempts to provide as many instances of HPO entities as possible, using the annotations in the GSC which has been annotated by experts, as well as directly from the HPO database.

4 Results and Discussion

The objective of this thesis is the development of a NER system for the recognition of HPO entities in free text. IHP relies on IBEnt to perform NER tasks. IBEnt was modified and adapted for HPO entities, having selected a specific feature set, as well as a group of validation rules. This is the main focus of this work.

The feature set and validation rules have been selected depending on which features and rules provided the best performance. IHP is a result of this selection process.

To test the performance of IHP, the GSC and TS are used as benchmarking tools, which provide a standard for HPO annotators. The GSC contains a group of abstracts annotated by experts, reflecting real-world situations. The TS contain specific tests, separated in different categories, meant to evaluate the linguistic strengths and weaknesses of an HPO annotator.

IHP achieved good results, being able to outperform the baseline annotator (Bio-LarK CR) in the GSC. This suggests that it could perform better in real-world situations. IHP underperforms in the TS, suggesting that it can't deal with complex linguistic structures as well as the baseline annotator. However, since these linguistic structures occur naturally in the GSC, in general, IHP still performs better.

In this chapter several aspects of this thesis will be discussed along with the obtained results:

- **Section 4.1 Overall Performance** will present the overall performance of IHP in the two benchmarking tools and compare these results with the baseline performance. The performance for both CRF-based classifiers used in IHP – CRFSuite and StanfordNER – are presented.
- **Section 4.2 Feature Performance** will show the difference between CRFSuite and StanfordNER (the available classifiers), presenting the selection of the feature set used for each of these classifiers, as well as provide an overview of the performance of some of the selected features.
- **Section 4.3 Validation Performance** will present the validation technique used for IHP. It will provide an overview of the handwritten rules and the dictionary used to increase the performance.
- **Section 4.4 Problems Faced During the Annotation** will present some of the issues found during the annotation process. It will discuss some of the difficulties found in the HPO, as well as some of the inconsistencies in the GSC.
- **Section 4.5 GCS+** will present the potential performance of IHP in the GSC when the inconsistencies are taken into account and present an updated version of the GSC, the GCS+, along with the results of IHP in this new version.

4.1 Overall Performance

The final performance for IHP was evaluated using Bio-LarK's benchmarking tools. Although CRFSuite is the chosen base CRF algorithm for IHP, since both StanfordNER and CRFSuite have a similar performance, both results are presented. These findings are the result of a combinations of several factors such as the classifier, the selected features and the validation process. These results can be found in [Table 4.1](#).

Table 4.1: Comparative performance of IHP and Bio-LarK CR in the Gold Standard Corpora and Test Suites.

	Gold Standard Corpora (F-Measure)	Test Suites (F-Measure)
CRFSuite	0.648	0.896
StanfordNER	0.646	0.869
Bio-LarK CR	0.560	0.950

4.1.1 Gold Standard Corpora Performance

The GSC is a benchmarking tool that consists of a group of abstracts and the corresponding annotations. In this thesis, the GSC is used for the training stage and the testing stage. It uses a 10-fold cross-validation.

The performance of the GSC depends on the performance of the feature selection and validation process. The combination of the machine learning annotator with the validation rules make it possible for IHP to outperform the Bio-LarK CR. As it can be seen on [Table 4.1](#), IHP achieved an F-measure of 0.646 and 0.648 using CRFSuite and StanfordNER, respectively. These results show that IHP outperforms Bio-LarK CR in the GSC, with an increase of 0.088 in performance.

4.1.2 Test Suites Performance

The TS is the other benchmarking tool used to test IHP's performance. They consist of different types of tests meant to evaluate the linguistic abilities of an annotator. In this thesis, the TS were tested using a model created from the GSC.

As we can see in [Table 4.1](#), IHP achieved an F-measure of 0.896 and 0.869 using CRFSuite and StanfordNER, respectively. Although, both classifiers underperformed when using the TS, CRFSuite slightly outperforms StanfordNER. The TS are meant to test the linguistic capacities of a recognition system and therefore, CRFSuite's selected features could be better suited for linguistics patterns than the StanfordNER features. These results show that IHP underperformed in the TS compared to Bio-LarK CR, corresponding to a decrease of 0.054 in performance. However, although IHP underperformed in the TS, these tests evaluate complex structures that may occur in natural text such as the abstracts in the GSC. This suggests that the GSC is more significant than the TS.

I believe that the reason why IHP underperformed in the TS is due to lack of experience with the linguistic field and in the future, with more experience in the area, it may be possible to improve the results to levels comparable to Bio-LarK CR, which has an outstanding performance.

4.2 Feature Performance

The selection of a feature set is an important step in NER systems. Choosing a rich feature set can have a big impact on the recognition of a specific type of entities. IHP has the option of using CRFSuite and StanfordNER as CRF-based classifiers. The performance of these classifiers is affected by the internal algorithm implementation (different software generally have different implementations) and by the choice of the feature set. In this work, different feature sets are used for the two classifiers. StanfordNER provides a list of available features that can be easily selected and combined in different ways. CRFSuite does not provide such a list and therefore the features have to be created.

In this section, the CRFSuite feature performance is presented, followed by StanfordNER features and the contribution of Brown clusters.

4.2.1 CRFSuite Features

Since CRFSuite does not provide a list of already crafted features to use, a feature set had to be created by developing a set of feature functions. IBEnt provides the option of developing features so that they can be used during the training stage. A group of feature functions were developed according to the work done by several authors (Zhou, 2004; Lee, 2003; Lin, 2004; Tsai, 2014; Hahn, 2016; Campos, 2012; Tkachenko, 2012).

To study the feature performance in CRFSuite, each feature was tested individually to check its impact. The feature performance results are divided into 6 categories: linguistic, orthographic, morphologic, context, lexical and others. The presented results show both the results for individual performance in each category, as well as the contribution of each category to the final performance. The baseline feature corresponds to the results when only the current token text is considered. Each feature was tested on a single cross-validation iteration. This greedy approach may create some sampling bias and overfitting, however, the objective is determining which features are more important and not in measuring the overall performance of IHP.

Table 4.2: The performance of the different types of used features for CRFSuite: Linguistic (L), Orthographic (O), Morphologic (M), Context (C), Lexical (LE) and others (X). These features were tested in a single cross-validation iteration.

Features	Precision	Recall	F-Measure
NO FEATURES	0.452	0.594	0.514
L	0.463	0.720	0.564
O	0.452	0.594	0.514
M	0.457	0.766	0.573
C	0.469	0.783	0.587
Le	0.453	0.606	0.518
X	0.428	0.697	0.530
L + O	0.458	0.720	0.560
L + O + M	0.451	0.760	0.566
L + O + M + C	0.478	0.800	0.598
L + O + M + C + Le	0.478	0.806	0.600
L + O + M + C + Le + X	0.482	0.823	0.608

As it can be seen in Table 4.2, the performance of IHP improves as features are added to the feature set. Although the performance improves steadily in F-measure, this is mainly due to an increase in recall since the precision values stay practically the same. The increase in recall, but not in precision, means that although more entities are being correctly identified, there are also incorrect word phrases being considered entities. A possible explanation for this is that IHP is focusing too much on the structure and less on the context.

Before the addition of the features, IHP made mistakes such as the identification of the word phrases "36 schwannomas" (instead of "schwannomas"), "ear malformations" (instead of "external ear malformations") and "jerky movements" (instead of "atactic jerky movements"). After the addition of the features it was able to identify the previous mistakes "schwannomas", "external ear malformations" and "atactic jerky movements". It also identified some incorrect word phrases such as "neuroanatomy" and "hippocampus", probably due to the fact that they use words used in other HPO entities.

By observation of [Table 4.2](#), it is possible to see that linguistic, morphologic and context features are responsible for the best performance individually but context features show the best individual performance between the three. A possible explanation for this is due to the fact that these features take the neighbor tokens into account and therefore gather more important information, allowing the system to perform better using context features over the other types. Although the lexical features and 'other' features seem to have a low impact on the system, a slight increase in performance is always relevant in NER systems. The used features can be found from [Table 3.1](#) to [Table 3.6](#).

In general, as the features are combined, the performance increases. However, as more features are added, the increments become smaller. This suggests that entities that are easy to identify have already been identified, and only more complex entities remain.

4.2.2 StanfordNER Feature Selection

Using the available list of StanfordNER features, it is possible to easily define values for each of the used features. A few base features that were already selected in IBEnt were kept. These base features are the following:

- useClassFeature
- useWord
- maxNGramLeng
- entitySubclassification=SBIEO
- wordShape=chris2useLC

The rest of the features were selected using an iterative script. The script I created uses a greedy approach to iterate over Boolean and numerical features. A greedy approach is used since it would require a great amount of time and computational power to iterate all the possible combination of features. The script iterates the list of Boolean features, determines whether a certain feature improves the performance and adds the feature to the base features. Using the base features and Boolean features as a feature set, the script then iterates the list of numerical features, using a range of numbers around the default value.

Before applying the features to IHP, IHP had an F-measure of 0.60960. [Figure 4.1](#) shows the selection process of all StanfordNER Boolean features. [Figure 4.2](#) shows the increase in performance as the final selected Boolean features are added to IHP.

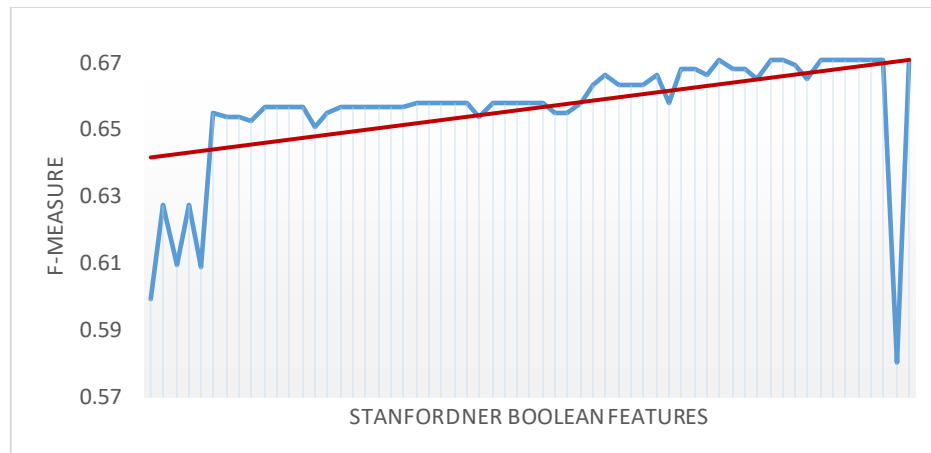


Figure 4.1 Iterative selection of StanfordNER's Boolean Features. This line graph shows the iterative process of selection of the features. The blue line represents the F-measure for each of the features throughout the selection process. The red line represents the performance trendline. The features were tested on a single cross-validation iteration.

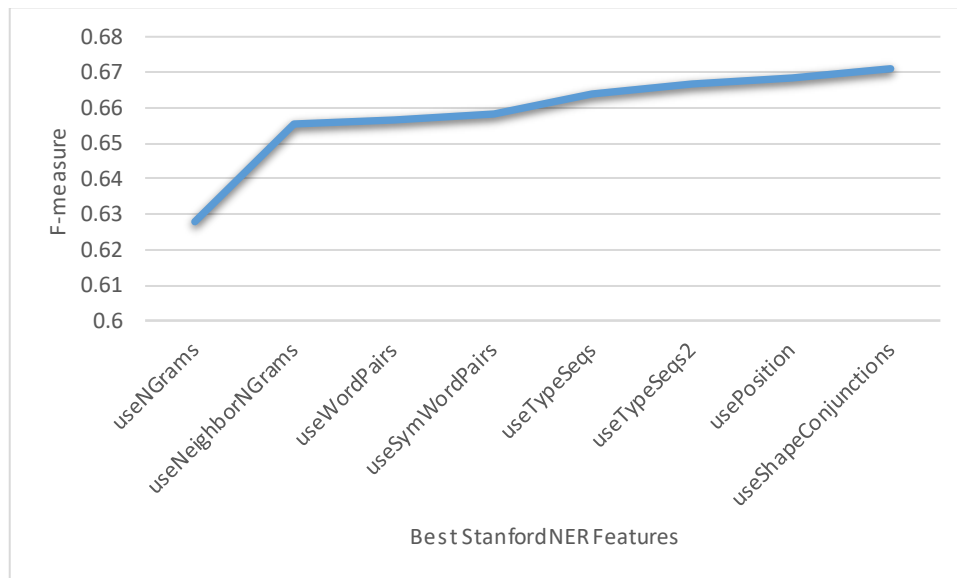


Figure 4.2: StanfordNER's Boolean Features. This is compact version of the previous graph, showing only the features that improved the results.

We can see by observation of [Figure 4.1](#) and [Figure 4.2](#) that there was an increase in the performance. The final value for the F-measure is 0.67099, corresponding to an increase of 0.0613 over the initial performance. The final Boolean StanfordNER features were added in addition to the base features. These added features are:

- useNGrams
- useNeighborNGrams
- useWordPairs
- useSymWordPairs
- useTypeSeqs
- useTypeSeqs2

- usePosition
- useShapeConjunctions

For StanfordNER's numerical features, although more than 100 different tests were conducted varying the numerical feature values, these didn't improve the performance with the currently selected features. It is probable that the numerical features did not have an impact in the performance due to the specific feature set that was used. This feature set could have features that were causing a conflict between the two types of features.

In the future, with more time, it would be possible to attempt choosing a better feature set as it is time consuming and arduous.

4.2.2.1 Brown Clusters

Word clustering features like Brown clustering (Brown P.F., 1992) help determine what words are more likely to be together, allowing the identification of specific patterns in the entities, contributing to finding new entities. Word clustering features have the potential to help the identification of new entities. This type of feature has to be carefully crafted in order to perform well. To use Brown clusters as features for a NER system there are several aspects to take into consideration. An important aspect is the text from which the Brown clusters are formed. The text used to create the clusters should be related to the text used for the identification of entities because words have specific relationships depending on the field they are applied to.

The Brown clustering algorithm that was used allowed the selection of different options. The choice of the best Brown Clustering was done using a script that tested variations between three parameters: number of clusters, number of collocations and minimal number of occurrences. After some preliminary tests, it was observed that the number of collocations did not affect the results and therefore this parameter was not considered. The clusters were tested on a single cross-validation iteration.

The Brown clusters are applied to StanfordNER and CRFSuite. The initial F-measure value for CRFSuite, before the use of Brown Clustering, is 0.66519, and for StanfordNER is 0.75789. A heatmap plot was used to present the F-measure values according to the number of clusters and the minimal number of occurrences.

By looking at Figure 4.3, it is possible to see that the selection of the right number of clusters, as well the minimum number of occurrences of a phrase, has an impact on the performance of the Brown clusters. The maximum value presented in the heatmap plot can be found when there are 95 clusters and a minimal number of occurrences of a word of 3. This maximum value corresponds to an F-measure value of 0.68421 for CRFSuite and 0.76115 for StanfordNER. This corresponds to an increase of F-measure of 0.019 for CRFSuite and 0.0033 for StanfordNER. Although the use of Brown clustering did not cause a great improvement in the performance, a small increase in performance is still a considerable improvement in a NER system.

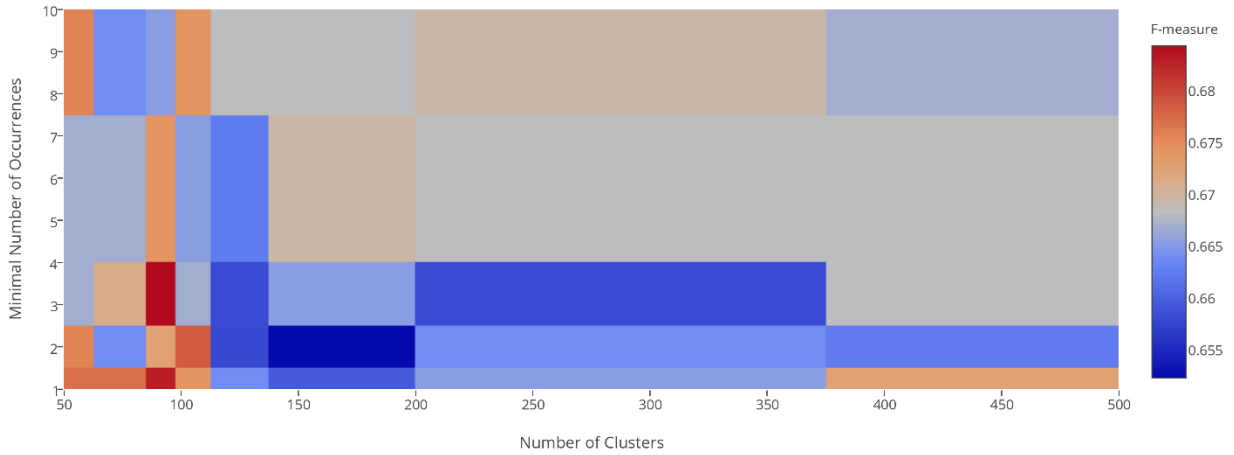


Figure 4.3 Brown Clusters selection process. Heatmap representation of the F-measure according to the number of clusters and the minimal number of occurrences. The clusters were tested on a single cross-validation iteration.

4.2.3 Overall feature performance

As it can be seen in [Table 4.1](#), there is not a great different in performance between CRFSuite and StanfordNER, although CRFSuite has a slightly better performance. The similarity in performance in both classifiers is probably due to the complex linguistic nature of HPO entities. It is possible that the machine learning algorithm does not perform as well with these complex entities and therefore reaches a point where entities become harder to identify.

There are some advantages and disadvantages when using one classifier over the other. While CRFSuite uses features that were manually crafted and selected according to previous works, StanfordNER uses features that were selected from the list of already available features. Having a list of available features can be extremely useful, cutting back on the manual labor and time dedicated to the creation of the features. The fact that different users can apply the same features means that it is easier to get feedback and more information about what specific feature combinations work best for some specific situations.

Overall it is clear that the choice of the correct combination of features can have a positive impact on the performance. As suggested by Tkachenko et al. ([Tkachenko, 2012](#)), features such as word features, Brown clusters and POS tagging work well for NER systems. General features like linguistic, morphologic, orthographic, lexical and context features should be considered for NER systems since they can have a good impact on the performance of an annotator.

In the future, it would be interesting to test some corpus-level features presented by Zhang et al. ([Li Zhang, 2004](#)), where entities are labeled as “focused”. These types of features that give more importance to some type of words could help improve the performance.

4.3 Validation Rules Performance

The validation process is extremely important for NER systems and one of the most important processes for IHP. For this thesis, a combination of handwritten rules and a dictionary was used in order to address some of the issues created by the machine learning classifier. This process is able to remove false positives, identify missed entities and combine adjacent entities.

The dictionary used in conjunction with the handwritten rules contains a list of HPO entities, along with possible synonyms for each of the entities. I created a script that receives the results from the machine learning-based annotator and applies different validation rules. The validation rules developed for IHP can roughly be divided into two categories: identification of entities and removal of entities.

The validation process gives priority to the recall, trying to identify as many entities as possible. Since this leads to the identification of incorrect entities, a removal process occurs afterwards to remove the misidentified entities, improving the precision of IHP. The results for the validation process for CRFSuite and StanfordNER are now presented:

Table 4.3: Performance of IHP's validation rules on the Gold Standard Corpora using CRFSuite. It was evaluated with no validation rules, only identification rules, only removal rules and all validation rules.

	Precision	Recall	F-measure
No Validation Rules	0.672	0.614	0.642
With Identification Rules	0.442	0.797	0.568
With Removal Rules	0.754	0.609	0.674
With Validation Rules	0.549	0.791	0.648

Table 4.4: Performance of IHP's validation rules on the Gold Standard Corpora using StanfordNER. It was evaluated with no validation rules, only identification rules, only removal rules and all validation rules.

	Precision	Recall	F-measure
No Validation Rules	0.705	0.642	0.672
With Identification Rules	0.448	0.775	0.568
With Removal Rules	0.762	0.639	0.695
With Validation Rules	0.555	0.772	0.646

IHP gives priority to the recall since it leads to the identification of a bigger quantity of the GSC entities. As it can be seen in both [Table 4.3](#) and [Table 4.4](#), although the F-measure results remain practically the same before and after the rules, there is a clear increase in recall and decrease in precision. The use of the identification validation rules leads to an increase of 0.19 in recall in comparison with no validation rules and remains relatively the same after the removal validation rules. The reason for the low precision values is due to the inconsistencies found in the GSC and due to IHP's attempt to identify as many HPO entities as possible. The removal process leads to an increase of 0.08 in precision in comparison with no validation rules and an increase of 0.9 in precision after the use of identification rules. It is possible to see that the identification rules are mainly responsible for the increase in recall, while the removal rules are mainly responsible for the increase in precision.

4.4 Problems Faced During the Annotation

During the course of this work, there were several problems that made it difficult to correctly annotate the terms in the GSC. Some of the problems that may have influenced the results include issues in the HPO, inconsistencies in the GSC and inexperience in the field.

This section will present a discussion of specific problems encountered.

4.4.1 Problems Faced with the Ontology

Some of the issues faced during the thesis involved the identification of long entities, lack of synonyms for dictionary-based NER systems and ambiguity of the Human language.

4.4.1.1 Entity length

One of the biggest barriers in NER is developing an annotator capable of identifying long descriptive terms. This becomes especially hard in cases like the HPO that has entities that range from 1 to 14 words. Since many of the long terms in this ontology are formed by the combination of atomic terms, it is important to use techniques that have this in consideration. For this purpose, in this thesis, several attempts to improve the performance with context features were made. In general, using context with a range of more than 4 did not improve the performance, and in some cases, it decreased the performance. In the end, context features were used with varying range, depending on the feature. Word shape and lemma were used with a range of 2, affixes were used with a range of 1 and POS tagging was used with a range of 4.

Since long entities can have a word length of 14, the used range should probably be longer. I believe that the reason why using a longer range was not beneficial is because the training set is not big enough. Maybe with a bigger training set, IHP would be able to be better prepared for a bigger variation of entity structures. It would be interesting to test longer ranges in the future if a bigger training set is available.

Since IHP often misses long entities, handwritten rules were developed to extend existing entities. These rules look for certain linguistic cues and check a dictionary to try to identify missed entities.

4.4.1.2 Synonyms

A common issue in the Biomedical field is finding a consensus on the words that are used to describe certain observations. For this reason, nowadays, many databases and ontologies use special fields to include possible synonyms for a certain entity. This can be extremely useful, especially for NER systems that are based on dictionaries.

The HPO, like many of these ontologies, contain a synonyms field. Although HPO contains a great number of synonyms, sometimes some essential ones are missing. An example of this happens in cases like the word “neoplasm”. While in entities like “brain neoplasm” there is a synonym “brain tumor”, the same is not true for the entity “neoplasm”, which does not contain the synonym “tumor” (it contains other synonyms like “neoplasia” and “cancer”). The addition of more synonyms to the ontology could be extremely beneficial for NER systems that use dictionaries to help improve the performance.

4.4.1.3 Ambiguity

The Human language can be extremely ambiguous. A common way to deal with ambiguities in a NER system is to use POS tags since POS taggers are trained with texts to help determine the correct syntactical tag of a word. Some ambiguities can, however, be hard to overcome, especially if the POS tagger is not trained for a specific field. The issue with the Human language and the choice of a POS tagger is that even high quality taggers can yield bad results depending on the situation.

A particular case that shows the difficulty of identifying the correct syntactical context, can be found with the entity “tumor suppressor”. This entity is composed by two nouns in which the first noun is being used as an adjective. Therefore, the correct way to tag this entity would be with Adjective-Noun. Several POS taggers (online annotators, NLTK tagger and Stanford POS tagger) were tested using this phrase as an input. Of all the annotators, only the Stanford POS tagger was able to correctly identify the context as Adjective-Noun (the others identified Noun-Noun).

4.4.2 Inconsistencies in the Corpora

Being unexperienced in the Linguistics and Text Mining field, there could be an ulterior motive for the way the GSC was annotated. The GSC seems to contain a few inconsistencies that could bring confusion to the annotator. I have divided the found inconsistencies in the GSC in four different types:

- Number of annotations
- Entity meaning
- Nested entities
- Superclass/subclass entities

These inconsistencies are going to be discussed below with examples from specific documents. Since the potential performance is affected by these inconsistencies, it will also be discussed in this section.

4.4.2.1 Number of Annotations

The number of times an entity is identified in a document seems to be inconsistent. For example, the entity “medulloblastoma”^[25] is used three times during the text but it is only annotated twice. In another situation the entity “preauricular pits”^[26] is also used three times during the text but is annotated all three times.

The NER system I developed tries to identify all instances of HPO entities, no matter how many times it appears in a text. I think it is important to extract as much information as possible, even if it is repeated. Having the correct number of times entities appear in text can be useful for calculating important values such as the Term Frequency, which is used to determine the importance of a term in a document.

The issue with having a NER system that annotates all the instances of entities means that the entities that were not considered on the GSC are considered as false positives. This means that there is a decrease in the precision and, therefore, F-measure.

^[25] Document 19533801 in the GSC

^[26] Document 998578 in the GSC

4.4.2.2 Entity Meaning

In some situations, annotated entities do not exactly match their meaning in the ontology. Although these cases are possibly just simple mistakes, they can lead to some entities being misidentified. An example is the annotation of “calcium metabolism”^[27] instead of “disturbance of calcium metabolism”. The entity “calcium metabolism” by itself does not have any meaning in the HPO because it does not correspond to any abnormality.

4.4.2.3 Nested Entities

Nested entities are entities that are contained within other entities. In the GSC there is not a consistent annotation of nested entities. Sometimes the entities that are inside another entity are annotated while other times they are not. An example of this occurs for the entity “skin and genital anomalies”^[28] and “spine and rib anomalies”^[29]. The entity “spine and rib anomalies” is annotated in the GSC, along with the entity “rib anomalies”. However, the same is not true for the entity “skin and genital anomalies”. This entity is annotated in the GSC but the entity “genital anomalies”^[30], which exists in the HPO, is not.

The identification of nested entities in some cases but not others can lead to the confusion of the annotator. IHP tries to identify as many HPO entities as possible. This brings the same problem as in the first case: since the entity does not exist in the GSC, it will be considered a false positive and will decrease the precision and F-measure.

4.4.2.4 Superclass/Subclass Entities

The final type of inconsistency found has to do with superclass/subclass entities. This is closely related to nested entities because it also involves the identification of entities inside other entities. It is possible that some annotators identify only the most specific class of a certain entity, while other annotators try to identify all the possible classes. However, the GSC is not consistent with the annotation of these types of entities.

An example of this occurs with the superclass entity “tumours” and the subclass entities “tumours of the nervous system”^[31] and “intracranial tumors”^[32]. In the first case, the GSC annotates both “tumours” and “tumours of the nervous system” as entities. In the second case, only “intracranial tumors” is considered an entity.

Inconsistency issues like this one can cause confusion for the annotator. IHP tries to identify all instances of superclass and subclass entities. This means that the identification of “tumors” in the second case will be considered a false positive, reducing the precision and F-measure once again.

^[27] Document 6882181 in the GSC

^[28] Document 12219090 in the GSC

^[29] Document 9096761 in the GSC

^[30] Accession number: HP_0000078

^[31] Document 2888021 in the GSC

^[32] Document 3134615 in the GSC

4.5 GSC+

It is common that some entities in a document escape the manual annotation process, leaving a gold standard corpus incomplete. This under-annotation by the curators may lead to the automatic identification of many false positive which may, in fact be correct annotations that were not considered during the manual annotation process (Grego & Couto, 2013). As previously discussed, I found that there are some inconsistencies in certain aspects of the GSC, such as the number of times an entity is annotated and the simultaneous annotation of superclass and subclass entities. For this reason, I studied a potential situation where inconsistent false positives identified by the annotator were filtered from the corpus. These filtered false positives are word phrases that can be found either in the HPO database and in the GSC, and therefore, in an ideal situation would be considered as true positives. Taking the example given previously, since the entity "tumours" is not considered in the GSC but exists in the HPO, it would be removed from the results. Table 4.5 presents the results from a test performed where the false positives were filtered.

Table 4.5: Potential Performance of IHP in the Gold Standard Corpus by removing false positives found either in the HPO database or GSC.

	Precision	Recall	F-Measure
No Filter	0.549	0.791	0.649
With Filter	0.845	0.791	0.817

As it can be seen in Table 4.5, there is an increase in performance of about 0.17, which is a significant improvement. To address the issues found in the GSC, we updated the GSC, dubbed GSC+ (<https://github.com/lasigeBioTM/IHP/blob/master/GSC+.rar>), taking into account the inconsistencies found. The GSC+ adds new instances of HPO entities that were automatically identified by IHP. Using the list of identified entities, the entities were checked by exact matching to see if they exist either in the HPO database or in the GSC annotations. The entities that were identified by exact matching were added to the GSC+. The GSC+ provides the addition of 881 new entities and the modification 4 entities. Table 4.6 presents the new results of IHP with GSC+.

Table 4.6: Performance of IHP with the GSC+.

	Precision	Recall	F-Measure
IHP	0.872	0.854	0.863

5 Conclusions

Nowadays, there is a great amount of information that is not in a computer-readable format and, therefore, not easily accessible. For this reason, the development of extraction systems is extremely important. Making information available in a structured format allows the creation of algorithms that access, search and make connection between different types of information. The objective of this thesis was to develop one of these extraction systems.

In this thesis, I developed a NER system capable of identifying HPO entities in unstructured text. A system like this allows the annotation of HPO entities in any unstructured text which can then be paired with other types of information to derive meaning. IHP uses a machine learning-based approach coupled with a combination of dictionary-based and handwritten rules-based methods.

The choice of a rich feature set (linguistic, morphologic, orthographic, context, lexical and other features) that fits the data has a great impact in the development of machine learning-based NER systems. In this thesis, the feature set was selected according to a group of previous works in the area of Biomedical NER so that IHP could identify as many biomedical entities from the HPO as possible. Hopefully, the selected feature set could help guide the development of other NER systems.

The validation stage plays a big role in the performance of NER systems. For this thesis, in order to deal with the complex linguistic patterns found in the HPO, specific hand-written rules were developed. These rules try to identify specific variations found in HPO terms, as well as remove entities that were misidentified by the machine learning annotator.

The recognition system I developed is able to outperform (increase of 0.08 in F-measure) the comparative annotator Bio-LarK CR in the GSC. Although IHP underperforms (decrease of 0.054 in F-measure) in the TS, the fact that the evaluated structures should occur naturally in free text, and therefore in the GSC, suggests that the GSC results are more significant. The results were obtained using two different CRF-based classifiers: StanfordNER and CRFSuite. With StanfordNER, IHP achieved an F-measure of 0.63498 for the GSC and 0.86916 for the TS, while with CRFSuite IHP achieved an F-measure of 0.64009 for the GSC and 0.89556 for the TS.

These results do not take in consideration the potential inconsistencies that can be found in the GSC. For this reason, a specific test was performed to take into consideration the possible inconsistencies in the GSC. This test removes false positives that can be found as annotations in the GSC or that exist inside the HPO database. The removal of false positives improves the precision of IHP, allowing an F-measure of 0.82 in the GSC, which is an improvement of 0.18 compared to the achieved results.

The results from this thesis reinforce the importance of the use of certain techniques. The choice of a rich feature set is extremely important for the development of a machine learning-based NER system. Although machine learning-based NER systems can be great by themselves, the use of dictionary-based and handwritten rule-based methods should always be considered as an option. On the one hand, machine learning-based approaches are flexible, being able to identify entities that are not in a dictionary. On the other hand, these systems may miss extremely obvious entities that can easily be identified by the addition of a dictionary. The use of mixed approaches can compensate some of the weaknesses in pure approaches.

5.1 Limitations and Range of Validity

One of the hardest tasks in NER is finding a big set of labeled data. In general, the bigger the data set is, the better it is possible to train a model for an annotator. A good model can yield high quality results and can deal with a lot of real-world scenarios.

In this thesis, the GSC was used as both the training set and the testing set, using a 10-fold cross-validation approach. Since the GSC is a relatively small corpus, it is possible that IHP underperforms in real situations. However, although IHP relies on a small training set, it also uses external knowledge through the use of a dictionary-based approach coupled with handwritten rules, which may compensate for the lack of a large corpus. IHP has a good potential but only more tests will determine its real value.

5.2 Significance to the area

Text Mining is a very important field in the scientific area since there is an enormous amount of information that is not available due to the lack of tools to extract relevant information from unstructured text. For this reason, the development of different types of NER systems is crucial so that we can organize, search and connect different types of information.

Annotators such as the one I developed are the key to extract important information and make it available to the world. IHP provides an easy way to extract HPO entities from free text and hopefully could be used as a standard for HPO annotation.

This thesis also reinforces some important aspects in the development of NER systems. IHP uses a combination of machine learning, dictionaries and handwritten rules to perform the annotation of HPO terms. Machine learning systems can benefit greatly from adding dictionary-based methods as well as specific rules to improve the performance.

Another important aspect is the use of a specific feature set. In this thesis, a carefully crafted feature set was used. This feature set was created based on the work of previous authors. This thesis reinforces the idea that the features selected for the NER system can have a great impact.

5.3 Contributions

The developed annotator, along with the selected feature set and developed handwritten rules can be found at <https://github.com/lasigeBioTM/IHP>. The updated version of the GSC, the GSC+, containing an addition of 881 entities and the modification of 4 entities is available at <https://github.com/lasigeBioTM/IHP/blob/master/GSC+.rar>.

Along with this thesis, a paper was written presenting IHP and its results. This paper has been submitted to BioMed Research International.

5.4 Future Work

Everyday new methods and techniques are developed allowing better algorithms to be created. The area of information extraction is in constant growth working to deal with all this information that is available. In the future it would be interesting to explore some particular issues such as the use of a different feature set, the influence of a bigger annotated corpus or applying phenotypic similarity. More work in some of these areas could bring important contributions to the Biomedical field.

One type of features that could be interesting to test are the corpus-level features presented by Zhang et al. (Li Zhang, 2004), where entities are labeled as “focused”. These types of features give more importance to some type of words and have the potential to improve the performance of an annotator.

Although the performance in the GSC is superior to other annotators, it is crucial that IHP has a good performance in real-world situations. For this, gathering a larger annotated corpus would allow IHP to be prepared for more situations and make it reliable for general use.

Phenotypic similarity measures could help identifying potential misannotations that are not semantically related with the entities found in the text.

IHP achieves better results compared to other annotators and it can still be improved further. With more experience in these areas it could be possible to get a better understanding of the structural patterns found in the HPO and improve the performance.

Hopefully, IHP could be considered a standard tool for the HPO, working in conjunction with IBEnt.

6 References

- Ashburner, M. a. (2000). Gene Ontology: tool for the unification of biology. *Nature genetics*, 25, 25-29.
- Brown P.F., d. P. (1992). Class-based n-gram models of natural language. . *Computational Linguistics*, 467–479.
- Campos, D. M. (2012). *Biomedical Named Entity Recognition: A Survey of Machine-Learning Tools*. Theory and Applications for Advanced Text Mining, Prof. Shigeaki Sakurai (Ed.), InTech. Retrieved from <http://www.intechopen.com/books/theory-and-applications-for-advanced-text-mining/biomedical-named-entity-recognition-a-survey-of-machine-learning-tools>
- Christopher D. Manning, P. R. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Grego, T., & Couto, F. M. (2013). Enhancement of chemical entity identification in text using semantic similarity validation. *PloS one*, 8, e62984.
- Groza T, K. S. (2015). Automatic concept recognition using the human phenotype ontology reference and test suite corpora. *Database* .
- Hahn, T. R. (2016). *Disease Named Entity Recognition Using Conditional Random Fields*. SMBM, volume 1650 of CEUR Workshop Proceedings, CEUR-WS.org.
- Jenny Rose Finkel, T. G. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling-. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 363-370.
- John D. Lafferty, A. M. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*, Carla E. Brodley and Andrea Pohoreckyj Danyluk (Eds.). Morgan Kaufmann Publishers Inc., 282-289.
- Jurafsky, D. a. (2009). *Speech and language processing*. Upper Saddle River, N.J: Pearson Prentice Hall.
- Lamurias A., F. J. (2015). Improving chemical entity recognition through h-index based semantic similarity. *J Cheminfo*, 7, S13.
- Lamurias, A. (2014). *Identifying Interactions Between Chemical Entities in Text*. Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa.
- Lee, K. H. (2003). *Two-phase biomedical NE recognition based on SVMs*. PA, USA: In Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine - Volume 13 (BioMed '03), Vol. 13. Association for Computational Linguistics.
- Li Zhang, Y. P. (2004). Focused named entity recognition using machine learning. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04)*, 281-288.
- Li, A. M. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03)*, Vol. 4. Association for Computational Linguistics, 188-191.

- Lin, Y. T. (2004). *A maximum entropy approach to biomedical named entity recognition*. Springer-Verlag, London, UK: In Proceedings of the 4th International Conference on Data Mining in Bioinformatics (BIOKDD'04), Mohammed J. Zaki, Shinichi Morishita, and Isidore Rigoutsos (Eds.).
- Manning, C. D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55-60.
- Mitchell, T. M. (1997). *Machine Learning, First Edition*. New York, NY, USA: McGraw-Hill, Inc.
- Mohri, M. R. (2012). *Foundations of Machine Learning*. The MIT Press.
- Noble, W. S. (2006). *What is a support vector machine?* (Vol. 24(12)). Nature Biotechnology.
- Okazaki, N. (2007). CRFsuite: a fast implementation of Conditional Random Fields (CRFs). Retrieved from <http://www.chokkan.org/software/crfsuite/>
- Plaisted, D. A. (2016). *Parse Trees [Lecture Notes]*. Department of Computer Science, University of North Carolina at Chapel Hill. Retrieved from <https://www.cs.unc.edu/~plaisted/comp455/slides/pt3.2.pdf>
- Scholz, G. F. (2010). *Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement*. SIGKDD Explor. Newsl. 12, 1.
- Sebastian Köhler, N. V. (2017). *The Human Phenotype Ontology in 2017*. Nucl. Acids Res. (2017) doi: 10.1093/nar/gkw1039.
- Tkachenko, M. S. (2012). Named Entity Recognition: Exploring Features. *Proceedings of KONVENS 2012*, 118-127.
- Tsai, T. W. (2014). *Using Maximum Entropy to Extract Biomedical Named Entities*. Nankang, Taipei, Taiwan: Institute of Information Science, Academia Sinica.
- Zhou, G. Z. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics* 20, 7 (May 2004), 1178-1190. doi:DOI=<http://dx.doi.org/10.1093/bioinformatics/bth060>
- Zhu, X. (2007). *Advanced Natural Language Processing: Conditional Random Fields [Lecture Notes]*. Department of Computer Sciences, University of Wisconsin-Madison.