IT UNIVERSITY OF COPENHAGEN

Named Entity Recognition and Disambiguation in Danish Electronic Health Records

Authors:
Mathias RASMUSSEN (matra@itu.dk)
Nichlas BERGGREIN (nimb@itu.dk)

Supervisor:
Leon DERZCYNSKI

A report submitted in fulfilment of the requirements for the MSc. Thesis

in the

MSc Software Development - Design Track
Course code: KISPECI1SE

June 3, 2019

IT UNIVERSITY OF COPENHAGEN

Keywords

Natural Language Processing, Machine Learning, Medical sector, Danish, Electronic Health Records, Information Retrieval, Named Entity Recognition, Named Entity Disambiguation, Sentence Boundary Disambiguation, Conditional Random Fields, International Classification of Diseases

Abstract

In recent times, actors from the private, and the public sector in Denmark are ambitious to make the data within the Health sector more accessible for processing and analysis. This is because the databases storing clinical texts are perceived as a mine of information. In relation, previous appliances of machine learning have led to an optimisation of processes in the Health sector in Denmark. In this study, we propose a coherent pipeline of **N**atural **L**anguage **P**roccesing (NLP)-components that serves as a baseline approach to process medical text in Danish. We collect 9,974 abbreviations specific to Danish medical texts to facilitate the segmentation of sentences in Danish **E**lectronic **H**ealth **R**ecord (EHR). Our approach succeeds in recognising disease entities from the health records by use of a Conditional Random Fields model. To provide a foundation for training the model, we manually annotate 92,485 words to capture the structure of Danish medical texts. Also, we discuss the performance with state-of-the-art practises from the CoNLL 2003 shared task and the i2b2 2010 challenge. Finally, we disambiguate the recognised disease mentions by linking them to a specific concept in a disease knowledge base. In the construction of this knowledge base, we use the **I**nternational **C**lassification of **D**iseases (ICD) and the Danish **S**undhedsvæsenets **K**lassifikations **S**ystem (SKS). Moreover, we present the practical implications of implementing a disambiguation process in the Danish Health Sector.

Acknowledgements

The accomplishment of this thesis benefits from the invaluable and constructive inputs from our supervisor, Leon DERZCYNSKI. It would not have been possible to achieve a thesis of this quality without his great expertise, and persisting enthusiasm in Natural Language Processing. We wish to thank Leon for guiding us and helping the development of the system that is proposed in this paper. Moreover, thank you for always being available, and the willingness to engage in our project.

Contents

List of Abbreviations

AMI	**A**verage **M**utual **I**nformation
BFGS	**B**royden- **F**letcher-**G**oldfarb- **S**hanno
BILOU	**B**eginning- **I**nside- **L**ast- **O**utside- **U**nit-length
BIO	**B**eginning- **I**nside- **O**utside
CRF	**C**onditional **R**andom **F**ields
EDA	**E**xploratory **D**ata **A**nalysis
EHR	**E**lectronic **H**ealth **R**ecord
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
GloVe	**G**lobal **V**ectors for Word Representation
HMM	**H**idden **M**arkov **M**odel
ICD	**I**nternational **C**lassification of **D**iseases
L-BFGS	**L**imited Memory- **B**royden- **F**letcher-**G**oldfarb- **S**hanno
LSTM	**L**ong **S**hort **T**erm **M**emory
MAP	**M**ean **A**verage **P**recision
MEM	**M**aximum **E**ntropy **M**odels
MEMM	**M**aximum **E**ntropy **M**arkov **M**odels
NED	**N**amed **E**ntity **D**isambiguation
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**roccesing
NLTK	**N**atural **L**anguage **T**ool **K**it
PMI	**P**ointwise **M**utual **I**nformation
POS	**P**art **O**f **S**peech
RNN	**R**ecurrent **N**eural **N**etwork

SBD	**S**entence **B**oundary **D**isambiguation
SKS	**S**undhedsvæsenets **K**lassifikations **S**ystem
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
WLM	**W**ikipedia **L**ink-based **M**easure
WSD	**W**ord **S**ense **D**isambiguation

Chapter 1

Introduction

1.1 Medical data in an electronic era

In May 2019, the Danish think tank, Mandag Morgen and the fund, Tryg published a codex for communication regarding medical subjects (Mandag-Morgen and Tryg-Fonden, 2019). The ambition is to improve the mediation of medical knowledge both within and outside the sector. The publication originates from a belief that the current communication leads to greater confusion rather than clarification for parties inside- and outside the medical sector. Among other main objectives in the codex, the ambition is to improve the communication between medical personnel and patients (Mandag-Morgen and Tryg-Fonden, 2019). Other movements revolving around the medical sector have unified actors from both the public and private sector in the search for a more accessible healthcare system in Denmark. The most prominent actors include Rigshospitalet, Region Hovedstaden, Københavns Kommune, Novo Nordisk, LEO Pharma, and Pfizer (Reiermann and K.-Andersen, 2019). The ambition of this constellation has its roots in a research project from the year 2000 where the processing of medical data succeeded to analyse the mortality rate of patients with severe stomach ulcers. Doctors did at the current time believe that the mortality rate was 10 pct., but the project showed that the disease, in fact, caused the death of 30 pct. This resulted in The Danish Health Authority modifying the guidelines for the treatment of patients with a severe stomach ulcer to treat these patients faster (Reiermann and K.-Andersen, 2019). This example is part of a collection of 29 research projects in Denmark that prove a positive outcome of processing big data in the medical sector (Reiermann and K.-Andersen, 2019).

The recent movements in Denmark align with global movements where the medical sector is going through an era of digitisation. In particular, the use of EHR has gained grounds because it helps to create a resourceful environment giving a better foundation for making decisions, assessment of care, and research (Fan et al., 2011; Cui, Xie, and Shen, 2018). The conversion to storing medical records electronically gives the possibility to the computational tool of machine learning. Within this field, the branch of NLP is a widely used tool for mining text to understand the structures of a language. This yields the ability to elicit useful knowledge from a text by computational power, which has been extensively applied on EHR in English (Afzal et al., 2018; Hamid et al., 2013; Fan and Zhang, 2018; Tvardik et al., 2018). In November 2018, Amazon published a machine learning service, Amazon Comprehend Medical, allowing the processing of unstructured medical text in English to extract patient diagnosis, treatments, dosages and so forth (Amazon, 2018).
International movements thus indicate progress on the matter that is now gaining importance for many parties in Denmark. However, the exploration of NLP on EHR in Danish shows that no progress has been made on the subject yet. Therefore, we wish to explore the possibility of applying NLP in the analysis of EHR in Danish. Among many machine learning techniques, NLP is a vital first component in a solution such as the one published by Amazon. This is because NLP enables the extraction of specific information from unstructured text, and convert that information into a structured format. This is a prerequisite for other machine learning techniques and statistical

approaches to work with the data. NLP thus posses the potential for expanding the Danish Healthcare system such that it accedes the exact ambitions of the group of companies, think tanks, funds, and the public sector in Denmark.

1.2 Structure of this research paper

To investigate the appliance of NLP on Danish medical texts, we work with the ambition to propose a baseline approach of NLP-components that can be used to extract specific medical entities EHR. This study is based on a preliminary course at the IT University of Copenhagen where the general practices of NLP were explored. Therefore, we already have fundamental knowledge that gives direction for the focus in this study.

The first part of this study explores state-of-the-art practices related to NLP to create an optimal foundation for proposing a pipeline for processing Danish EHR. Having that foundation, the first challenge of finding sentences and words in the medical text will be approached. The following is to come up with a solution for eliciting disease names from the medical text. Hereafter, the ambition is to add meaning to the extracted entities by linking these to an existing knowledge base. The output of this practice is medically disambiguated words that are helping to determine the ability of the proposed solution to process medical text with a useful output. Finally, we discuss the performance with state-of-the-art practices and highlight the practical implications of the proposed solution.

1.2.1 Purpose statement

We wish to investigate the possibility of applying NLP on EHR in Danish. In doing so, our ambition is to develop a software solution that enables the recognition and disambiguation of Danish disease mentions.

1.2.2 Research questions

- What are the most suitable strategies for splitting Danish medical texts into sentences, and words?
- How are words that mention a disease recognised and extracted from all other words in a Danish medical text?
- How can recognised disease entities in Danish be linked to a knowledge base to achieve a common realm of understanding?

For the sake of understanding, we wish to emphasise that the term "Corpora" is referring to a database containing text products. "Corpus" is used for talking about a single EHR. Finally, "Document" is regarding a single sentence from a EHR. These terms will be used consistently throughout this paper.

Chapter 2

Background

The objective of the background section is to explore and present best-practices, theoretical approaches and potential resources for creation of a knowledge base.

2.1 Electronic Medical Resources

In this study, we make use of the computational tool of NLP why there is a need for having medical resources in an electronic format. Two types of resources are used in this study; a database containing EHR in Danish, and a reference database containing ground-truth on diseases in general. The acquisition and composure of these resources are now presented.

2.1.1 Database with Danish electronic health records

The data set giving the basis for research in this study was obtained through the IT-University of Copenhagen. This database is hereafter on referred to as the "E4C-2010"-corpora. Before acquiring the data set, it was pre-processed to de-identify the records as only authorised personnel is allowed to view the identifiable version of the data. The de-identification process did not include any modification to the clinical data. However, changes were made to identifiers that can be used to derive what patient or doctor that is implicated in a given consultation (Pantazos, Lauesen, and Lippert, 2017). In the following, we present the impact of the pre-processing steps undertaken.

Personal data

A permutation table was created to map existing identifiers to new ones. These identifiers include first male names, first female names, last names, street names, zip codes, hospital and clinic names (Pantazos, Lauesen, and Lippert, 2017). For the Civil Registration Number, a table mapping existing numbers to distorted numbers was created. The distortion started with formatting CPR numbers (Civil Registration Number) written in the format $DDMMYY - CSSG$, where $DDMMYY$ is the birth date, C indicates the birth century, SS indicates serial number, and G indicates gender. The DD and MM were changed to a random, valid day and month, while C was not changed. SS was changed, while gender, G, was not changed (Pantazos, Lauesen, and Lippert, 2017). For other identifiers as emails, phone numbers, and URLs, a random variable was substituted with the existing value (Pantazos, Lauesen, and Lippert, 2017).

Whenever ambiguous words were encountered, the words were handled in two possible ways (Pantazos, Lauesen, and Lippert, 2017). If the ambiguous word occurs more than 200 times, it is conceived as safe and the word is kept in the database. Otherwise, the patient record is deleted, but the reference to other patients will not be repaired. Instead, it will direct to a deleted patient record (Pantazos, Lauesen, and Lippert, 2017).

Dependency between tables

Changes were also made to tables containing references to other tables in the database.

CPR The table containing CPR (Civil Registration Numbers) was modified by collecting all CPR and substitute them with a random number. If the number was already used as a substitution, it was re-randomised (Pantazos, Lauesen, and Lippert, 2017).

Last names Three sources were used to collect last names; the names from the data set, Danmarks Statistik and a study of names at the University of Copenhagen, resulting in 56,339 last names. A frequency count was done on the merged table, and last names in the data set were replaced by last names with according frequency in the merged last name table (Pantazos, Lauesen, and Lippert, 2017).

Male and female names The same approach for last names was adopted for male and female names. Gender-specific names were extracted from the data set by inspecting the CPR, giving away the gender of the patient. The final collection of names contains 11,415 male names and 13,044 female names (Pantazos, Lauesen, and Lippert, 2017).

Street names Street names were collected from the patients' addresses where the floor and entrance letters were left out. In support, addresses were collected from the civil register in Denmark, which resulted in a total amount of 25,429 distinct addresses.

Zip codes Zip codes were collected from the Danish Postal Service, amounting 1,396 zip codes.

Hospital names and clinic names Hospital and clinical names were collected from the "E4C-2010"-corpora, Region Hovedstaden, Region Sjælland, Region Syddanmark, Region Midtjylland, Region Nordjylland, Queen Ingrid's Hospital in Greenland, Faroe Islands website, Sygehusvalg and Brancheforeningen for Privathospitaler og Klinikker (the trade association). The result was a list of 219 clinic names and 93 hospital names.

Ambiguous words A table of ambiguous names was extracted using the Danish Dictionary in Microsoft Office Word 2010 yielding a list of 3,557 names. This list was reduced by a medical expert, leading to a list of 1,952 entries. Finally, this list was enriched by 3,246 additional eponymous names retrieved from the website "Who named it" containing medical eponyms (Whonamedit?, 2010; Pantazos, Lauesen, and Lippert, 2017).

The data set initially contained 437,164 EHRs. Then, 69,914 EHRs were deleted due to data corruption (old test data and records remaining after a failure of the system). Furthermore, 41,119 EHRs were deleted because it contained rare ambiguous words or the patient was older than 90 years old. Eventually, we acquired a data set that contains 323,122 EHRs in Danish (Pantazos, Lauesen, and Lippert, 2017).

2.1.2 Prominent databases with disease information

Causes of death have been registered ever since the middle of the 15th century (Moriyama et al., 2011). This was the beginning of a persisting ambition to classify diseases and causes of death in a systematic manner allowing the generation of statistics (WHO, 2004). In 1890, Jacques Bertillon created the first catalogue of causes of death called "The Bertillon Classification of Causes of Death", which was adopted by 26 countries. This has ever since been known as the ICD-1 (WHO, 2004).

Throughout the years, multiple revised and continued classifications have been published. In the sixth version of ICD, a number system was introduced to combine the disease description with a specific unique code to ease the classification of diseases (Moriyama et al., 2011). In 1997, The eighth version of ICD was integrated into the Danish National Patient Register (Nielsen, 2017). This register serves to classify diseases in Denmark. The Danish register is based on the ICD database augmented with disease descriptions. This database is referred to as the SKS (Patientregistrering, 2018). As mentioned, the ICD classification is updated on a continuous basis, where the latest edition is named ICD-11 (WHO, 2004). The current implementation in the Danish SKS is however still based on the ICD-10.

2.2 Locating sentences in a text

The practice of finding sentences in a text has many names, but in this paper, it is referred to as **S**entence **B**oundary **D**isambiguation (SBD) (Kiss and Strunk, 2006). This task concerns dividing a piece of text into individual sentences. Although the task has a simple objective, the execution can be more complicated. In addition, the success of this task affects later processing steps as errors introduced in this step propagates to subsequent processes (Kiss and Strunk, 2002; Reynar and Ratnaparkhi, 1997; Kiss and Strunk, 2006). A trivial approach to segment sentences would be to look for punctuation marks. However, punctuation marks are used in abbreviations, initials and proper nouns, ordinal numbers, and ellipses. In contrast to the punctuation mark, exclamation and question marks depict the end of a sentence (Kiss and Strunk, 2006). Finally, it is common in a free text that the writing is not consistent with the grammatical rules of the language. Therefore, the simple objective of finding sentences is perceived as rather complex.

Regular expressions have been proposed as a solution to overcome the challenges mentioned above. However, Grefenstette and Tapanainen, 1994 argues in their study that this approach produced inferior results and is therefore not found adequate. Nonetheless, the researchers extended their approach with the use of a lexicon to recognise non-abbreviation tokens, frequent abbreviations, and domain-specific abbreviations. This extension was shown to be of great use where the technique locates almost all sentence boundaries in a text (Grefenstette and Tapanainen, 1994).

However, Reynar and Ratnaparkhi, 1997 argued that this approach is still not sufficient for overcoming all challenges related to SBD. Therefore, they applied a supervised learning method to provide a better solution. Their solution is based on a **M**aximum **E**ntropy **M**odels (MEM), which learns a set of rules from annotated text (Reynar and Ratnaparkhi, 1997). By computation of the joint probability distribution, the researchers sought to map the nearest context of a token to determine whether it is a sentence boundary or not.

In relation, Kiss and Strunk, 2006 proposes an unsupervised learning method called the Punkt System. This is to provide a language-independent technique where the model is created on the domain-specific corpus. In doing so, the technique is regarding the local context of a word and generalised knowledge from the entire corpus to find abbreviations, ellipses, sentence boundaries, abbreviations followed by sentence boundary, and ellipses followed by a sentence boundary (Kiss and Strunk, 2006). The **N**atural **L**anguage **T**ool **K**it (NLTK) provides an implementation of the Punkt system (Bird, Klein, and Loper, 2009). This implementation is distributed with pre-trained language specific models, including a Danish model, that enable instant disambiguation of sentence boundaries.

In Table 2.1, we present performances of other tool-kits that have been applied with the purpose of finding sentence boundaries.

TABLE 2.1: SBD results on medical texts. Precision (Pr), Recall (Re), and F1-score

	GENIA			**i2b2**		
Toolkit	Pr	Re	F1	Pr	Re	F1
Stanford	98 %	98 %	98 %	58 %	34 %	43 %
Lingpipe	98 %	97 %	98 %	57 %	33 %	42 %
Splitta	99 %	98 %	99 %	59 %	35 %	43 %
SPECIALIST	89 %	94 %	92 %	58 %	53 %	56 %
cTAKES	62 %	76 %	68 %	93 %	97 %	95 %
Average	**89.2 %**	**92.6 %**	**91.0 %**	**65 %**	**50.4 %**	**55.8 %**

2.3 Dividing sentences into words

The practice of dividing a sentence into words is referred to as word tokenisation. The purpose of dividing sentences into words is to obtain a single unit representation. This allows the further processing of single words and their context. Word boundaries can be detected by finding the whitespace that separates one word from the other. However, there are situations where splitting merely on whitespaces is not sufficient (Horsmann and Zesch, 2016).

Many challenges can be experienced when performing word tokenisation. In some situations, words are encapsulated in parentheses, but the parenthesis is frequently not part of the word why it needs to be separated from the surrounding parentheses (Horsmann and Zesch, 2016). However, parentheses are often assumed to be part of the word in a medical setting (Tomanek, Wermter, and Hahn, 2007). Another obstacle is the use of punctuation, quotes, and special characters.

Several approaches seek to overcome the challenges mentioned above by the use of regular expressions that capture certain tokens individually (Grefenstette and Tapanainen, 1994). Also, hand-crafted word lists are used to help the capturing of words that are not recognised by the regular expressions (Horsmann and Zesch, 2016). Another approach is the use of tree-banks that helps to annotate the syntax of a language and as a result of this, enables the detection of word boundaries (MacIntyre, 1995; Horsmann and Zesch, 2016).

Finally, unsupervised and supervised models have been used to map the statistical properties of word token boundaries. The models are generally trained on corpora such as PennBioIE, JULIE, and GENIA (Wrenn, Stetson, and Johnson, 2007; Tomanek, Wermter, and Hahn, 2007).

2.4 Finding specific words in a text

In this section, we explore the sub-task of information extraction that has the purpose of finding and classifying specific words in unstructured text. Examples of such words are mentions of persons, organisations, medical doses, diseases, time and dates, quantities, and so forth (Peng and McCallum, 2006).

This task can be approached by using the techniques of **N**amed **E**ntity **R**ecognition (NER). In the following, we begin by exploring the most prominent concepts within NER. Hereafter, we present practical appliances and related theory.

2.4.1 Word tagging

The discipline of tagging words is strongly restricted by the domain in which words are to be tagged. That is, all languages are composed differently. Moreover, the use of a given language may vary within different sub-domains. The technique of word tagging is used to generate a simple representation of complex language structures to allow further processing tasks. Word tagging can be performed in numerous ways depending on the purpose of the language processing task. In

the following, we present the most prominent approaches for tagging words.

Part **O**f **S**peech (POS)-tags are used to partition a text into known word classes such as nouns, verbs, pronouns, prepositions, adverbs, conjunctions, participles, and articles (Jurafsky and Martin, 2014). This classification is useful in NLP as it helps to reveal information about a given word and its surrounding context. This can help to understand the pattern of a given sentence (Jurafsky and Martin, 2014).

This comes beneficial in extracting information as POS-tagging facilitates the labelling of named entities as persons or organisations (Jurafsky and Martin, 2014). Several corpora have been used for recognising POS classes in English text where the most remarkable are the Brown corpus, Wall-Street-Journal (WSJ) corpus, and the Switchboard corpus (Jurafsky and Martin, 2014). In Danish, the most prominent corpus is the Danish Dependency Treebank proposed by Mathias Kromann (Trautner Kromann, 2003).

The level of detail achieved by using the POS-format gives rise to a detailed output such as finding word-relations across texts. However, the detailed input demands higher time consumption and is dependent on a larger data set needed for exploring and validating POS patterns. To compensate for this, the approach of word chunking has been proposed.

Beginning- **I**nside- **O**utside (BIO)-tags is a common tagging scheme used within word chunking. This format separates a given text into word classes of B, I and O (Ramshaw and Marcus, 1995). "B" is used for highlighting the beginning a word of interest. "I" emphasises that the subsequent word of the current is connected in providing meaning. "O" is used for classifying the given word as outside, meaning that it is of no interest (Ramshaw and Marcus, 1995). An extension of the BIO format is the **B**eginning- **I**nside- **L**ast- **O**utside- **U**nit-length (BILOU) format, which has shown to be beneficial as well in tagging chunks of words (Ratinov and Roth, 2009a).

The BIO-scheme makes it possible to learn patterns of text, and thereby assist in extracting specific information. Compared to the POS-tags, this scheme approaches the partition of a text more simply due to a substantially lower number of word classes (Ramshaw and Marcus, 1995).

2.4.2 Approaches to Named Entity Recognition

To find the distribution of labels from a given tagging scheme in a text, the approach of NER can be applied. In this part, we show the popular approaches and techniques within NER. In the first part, we consider machine learning models and how these are encoded. In the following part, we regard optimisation of models, and moreover how the models are modified to adopt a generalising truth. Finally, techniques for decoding a trained model are presented.

Model selection

The area of NER within NLP is a well-explored research area where many applications have adopted distinct approaches. The choice of model depends on the domain use and thereby the performance of the systems. A common objective for all is that they strive to solve a statistical classification problem. A prominent assumption within the area of NLP is to use NER models that obey the Markov property stating that during a sequential mapping, a given state is independent of the past and the future (Dymarski, 2011).

Hidden **M**arkov **M**odel (HMM) is a popular approach that provides a finite trainable stochastic automate. That is, an inference model is constructed on the assumption of the Markov property. The model is a probabilistic graphical model that makes it possible to predict a sequence of unknown hidden variables from a set of variables in an observed sequence (Dymarski, 2011).

An extension of the HMM is the **M**aximum **E**ntropy **M**arkov **M**odels (MEMM). This model determines the probability distribution of prior knowledge that leads to the current state. The

highest entropy of previous information is assumed to determine the best combination of previous knowledge, leading to the current state in the model (De Martino and De Martino, 2018).

These two approaches are however subject to the label bias problem meaning that the output models make local decisions, not accommodating the global probability of a label sequence given an observed sequence (Phuong, Phan, and The Trung, 2013). To overcome the label bias problem, the technique of **C**onditional **R**andom **F**ields (CRF) was proposed by Lafferty, McCallum, and Pereira, 2001.

CRF are used to build probabilistic models that can segment and label data in sequence form (Lafferty, McCallum, and Pereira, 2001). The conditional model determines the probabilities of possible label sequences given an observed input sequence. CRF is conceived as a finite state model with unnormalized transition probabilities that accounts for the global maximum likelihood. This is done by computation of the joint probability distribution of the total label sequence Y when considering the observed input sequence X. This is beneficial because CRF will hold no strict independence assumptions allowing the inclusion of any context information (Lafferty, McCallum, and Pereira, 2001).

CRF is a widely used technique that has been applied within text processing (Peng and McCallum, 2006; Sha and Pereira, 2003), bio-informatics (Settles, 2005; Liu et al., 2006; Sato and Sakakibara, 2005), and computer vision (Lavergne, Cappé, and Yvon, 2010).

Finally, the common application of neural approaches in machine learning is also found beneficial in recognising entities. The **R**ecurrent **N**eural **N**etwork (RNN) is useful for making classifications that are not only based on the current observation sequence, but also previous decisions made in the network (Tarasov, 2015). An extension of this approach is the **L**ong **S**hort **T**erm **M**emory (LSTM) that models intermediary observed sequences with its long-distance dependency. This approach has shown to achieve greater accuracy than a traditional RNN approach (Sak, Senior, and Beaufays, 2014).

Using the properties of a word

The main objective within NER is to train a model such that it recognises the structures of a given language. The best foundation for approaching this objective is to obtain information from the language itself. This can be achieved by the inclusion of individual- and contextual word properties, also known as features.

Individual word properties regard any additional representation of a token. A widely used representation is the shape of the word. In general, the word shape considers properties such as the length of the word, the inclusion of capitalised letters, and digits, and collection of characters (Manning and Schütze, 1999).

Another prominent approach to capture individual word properties is by the use of N-grams. These are characterised as a contiguous sequence of n characters that are taken from a longer string. N-grams can focus on the sequential text by either inspecting the characters included in a word or the combination of words in a text (Cavnar and Trenkle, 1994). N-grams can be of different sizes where the most general are 1-word; uni-grams, 2-words; bi-grams, 3: tri-grams and 4-words; four-gram (Cavnar and Trenkle, 1994). N-grams are typically used to support the detection of resemblance between parts of text (Cavnar and Trenkle, 1994; Broder et al., 1997).

The lemma of a word is also seen as important because it can be used to find the original form of a word from its inflected form. This can help to find the actual meaning of the word e.g. *lying* might indicate the verb $lie - lay$ or $lie - lied$. The lemma is thus helping to disambiguate the word facilitating a generalisation on the representation of the word (Manning and Schütze, 1999).

Current research has been successful in using the distributional similarity between tokens as a word representation. Different clustering techniques can be used to gain knowledge about the distributional semantic similarity. These techniques seek to generalise by forming bins or equivalence classes of words (Manning and Schütze, 1999). This serves to allocate words to similar environments where the conception of one word most likely applies to all other words in that cluster (Manning and Schütze, 1999).

Brown clustering is a renowned branch of word clustering (Derczynski and Chester, 2016). It is an unsupervised technique used to group word types that have similar distributional information (Derczynski and Chester, 2016). The technique is a greedy, hierarchical, agglomerate hard clustering algorithm that divides a vocabulary of words into a set of clusters with minimal loss of mutual information (Brown et al., 1992; Van Rijsbergen, 1977). The algorithm intends to create a number of pre-defined clusters where the output clusters are organised as leaves of a binary tree. Paths to clusters are given as bit strings expressing branches from the root (Derczynski and Chester, 2016).

Another approach is Word2Vec. This procedure also includes the construction of a tree but is based on word vectors. This approach seeks to learn the distributed vector representations to capture syntactic and semantic word relationships (Mikolov et al., 2013). This is done to find word representations that function as a centre for predicting surrounding words in a sentence or document.

A final approach to consider is word clustering by the **G**lobal **V**ectors for Word Representation (GloVe) procedure proposed by Pennington, Socher, and Manning, 2014. GloVe is a log-bilinear model with a weighted least-squares objective. The overall idea is to observe the ratios of word-word co-occurrence probabilities as these are conceived to reveal information on the use of the words (Pennington, Socher, and Manning, 2014).

Contextual word properties help to learn the structures of sequences because words are assumed to be characterised by "the company it keeps" (Manning and Schütze, 1999). This can be achieved by inspecting highly frequent situations where one word representation is concurrently appearing with other specific word representations. The relationship of word representations can then be used to determine the likelihood of a given context occurring if observing a given word (Manning and Schütze, 1999). Thus, the context helps to obtain the properties of a word and in what language structures the word is used. In other words, the contextual appearance eases the analysis of the similarity between sequences (Manning and Schütze, 1999).

Model optimisation

Optimisation of machine learning models is in general conceived as a very difficult task. For that reason, it has become a tradition to design the objective function such that it results in a curved graph to ease the optimisation problem (Goodfellow, Bengio, and Courville, 2016). Thus, the simplified problem when optimising a model is to locate a global minimum or maximum, indicating an acceptable solution and level of training (Goodfellow, Bengio, and Courville, 2016). Different approaches to model optimisation include iterative scaling algorithms (Lafferty, McCallum, and Pereira, 2001; Darroch and Ratcliff, 1972; Goodman, 2002), conjugate gradient (Kazama and Tsujii, 2003) or Stochastic Gradient Descent (Goodfellow, Bengio, and Courville, 2016). However, the most efficient approach has shown to be the Quasi-Newton method (Sutton and McCallum, 2010; Phuong, Phan, and The Trung, 2013; Byrd et al., 1995).

Model generalisation

In machine learning, situations occur where the model is trained to an optimal point, but the model fails to process real data satisfyingly due to its inability to generalise (Goodfellow, Bengio, and Courville, 2016). This is defined as the gap between training error and generalisation error. This

challenge arises when the training set contains a large number of parameters, but no focus points are provided to the learning algorithm (Sutton and McCallum, 2010). This could lead to a situation where the model is either under- or over-trained (Goodfellow, Bengio, and Courville, 2016). For that reason, the technique of regularisation is widely used. There is no generally applicable method of regularisation, but is rather a task of finding the best solution for the specific task that a model is to solve (Goodfellow, Bengio, and Courville, 2016). The general idea is to achieve regularisation by adding a restriction to the objective function of a given model.

Common and simple approaches to regularisation are linear models of linear regression, and logistic regression (Goodfellow, Bengio, and Courville, 2016). Another approach is the L_1 regularisation norm that shrinks weights norms that are too high to zero. This leads to a sparse solution where the majority of the input features have a weight of 0. Thus, it assigns insignificant features with an insignificant weight of 0 and significant features with a non-zero weight (Sutton and McCallum, 2010). On the other hand, the L_2 regularisation norm shrinks weights but maintains all weights as non-zero values. This approach is beneficial when all features are affecting the output of the objective function. This technique leads to a non-sparse solution in the sense that many features are included with a non-zero weight (Goodfellow, Bengio, and Courville, 2016).

Model decoding

To decode a model, a sequence is considered as a path through a graph where each node is a possible state at a given time. To find the most probable label sequence, one is interested in finding the longest path by viewing the path as a product of the transition probabilities along the path and the probabilities observed at each state (Russell and Norvig, 2016).

The Viterbi algorithm is a renowned technique for finding the most likely sequence (Sutton and McCallum, 2010; Ratinov and Roth, 2009b). It is seen as similar to filtering because it runs forward along a given sequence, and determines the probability at each state, allowing it to filter out unnecessary steps (Russell and Norvig, 2016). Along the path, when the algorithm observes potential best sequences, it maintains pointers from the given states to eventually output the best sequences. Then the best sequence can easily be selected from the candidate paths outlining the best possible sequences (Russell and Norvig, 2016).

An alternative to the Viterbi algorithm is the A* search, which is known as the best-first search algorithm (Russell and Norvig, 2016). With an optimistic heuristic, this algorithm can be used to compute the most probable label sequence. It evaluates states along the path by the product of the cost to reach the given state, and the cost to get from that state to the goal. By the optimistic heuristic, the state with the highest probability will always be expanded. This will continue until the observed data sequence has ended, leaving back the best combination of states with the highest probability (Russell and Norvig, 2016). Ratinov and Roth, 2009b argue that the Beam-search variant of A* search is even faster and just as accurate as the Viterbi algorithm. However, the A*-search may appear unfit because it only considers the locally highest probability. This might lead to the expansion of subsequent states that have a low probability. In contrast, the Viterbi algorithm is learning during the phase of decoding, and takes previous choices into account, helping to produce the globally most probable label sequence (Russell and Norvig, 2016). An extension of the Viterbi algorithm is proposed in the NCRF++ framework where the use of neural approaches helps to make the sequence labelling task more efficient and effective (Yang and Zhang, 2018) .

2.4.3 Related work on Named Entity Recognition

Each year, The SIGNLL Conference on Computational Natural Language Learning publishes a shared task to be solved by the use of NLP (Computational Language Learning, 2019b). By giving the same foundation for creating different systems, the shared task can be used to benchmark performances within different categories of NLP practices (Computational Language Learning, 2019b).

Since its first task of Noun Phrase chunking in 1999, the tasks have for instance included text chunking (2000), NER (2002, 2003), Syntactic and Semantic Dependencies (2009), Grammatical Error correction (2013, 2014) and Morphological analysis (2018) (Computational Language Learning, 2019a). In particular, the progress made on NER is of high relevance for this study as it emphasises the benchmark performance of this sub-task within NLP.

In 2003, sixteen systems participated in solving the shared task of NER on an English-language data set using the POS-tagging format (Computational Language Learning, 2003). The corpus was obtained from the Reuters Corpus where the training set consists of 946 articles with 14,987 sentences accumulating to 203,621 word tokens (Sang and De Meulder, 2003a).

In Table 2.2, we present the implemented models and the related performance.

TABLE 2.2: CoNLL 2003 shared task: Named Entity Recognition results. Precision (Pr), Recall (Re), and F1-score

Authors	*Pr*	*Re*	*F1*	*Tagging scheme*	*Model*
Florian et al., 2003	88.99 %	88.54 %	88.76 %	POS	MEM and HMM
Chieu and Ng, 2003	88.12 %	88.51 %	88.31 %	POS	MEM
Klein et al., 2003	85.93 %	86.21 %	86.07 %	POS	MEM, HMM and CMM
Zhang and Johnson, 2003	86.13 %	84.88 %	85.50 %	POS	Risk minimisation
Carreras, Màrquez, and Padró, 2003	84.05 %	85.96 %	85.00 %	POS	Voted perceptrons
Curran and Clark, 2003	84.29 %	85.50 %	84.89 %	POS	MEM
Mayfield, McNamee, and Piatko, 2003	84.45 %	84.90 %	84.67 %	POS	SVM
Carreras, Màrquez, and Padró, 2003	85.81 %	82.84 %	84.30 %	POS	AdaBoost.MH
McCallum and Li, 2003	84.52 %	83.55 %	84.04 %	None	CRF
Bender, Och, and Ney, 2003	84.68 %	83.18 %	83.92 %	POS	MEM
Munro, Ler, and Patrick, 2003	80.87 %	84.21 %	82.50 %	POS	N/A
Wu, Ngai, and Carpuat, 2003	82.02 %	81.39 %	81.70 %	POS	AdaBoost.MH
Whitelaw and Patrick, 2003	81.60 %	78.05 %	79.78 %	None	HMM
Hendrickx and Van Den Bosch, 2003	76.33 %	80.17 %	78.20 %	POS	MB learning
De Meulder and Daelemans, 2003	75.84 %	78.13 %	76.97 %	POS	MB learning
Hammerton, 2003	69.09 %	53.26 %	60.15 %	POS	RNN
Average	**82.67 %**	**81.83 %**	**82.17 %**	N/A	N/A

The majority of the submitted implementations used a wide range of features to enrich the training of the respective machine learning models. These features include affix information such as n-grams, prefixes and suffixes, bag-of-words, chunk tags, gazetteers and lexical features (Computational Language Learning, 2003).

In strong relation with the CoNLL shared task (Computational Language Learning, 2019b), the i2b2 tranSMART Foundation (Foundation, 2019) seeks to create a related engagement by posting shared tasks on NLP within the medical domain. In 2010, the i2b2 tranSMART Foundation (Foundation, 2019) posted an assignment on extracting medical entities from a data set. The size of the corpus in the challenge was 394 training reports, 477 test reports, and 877 unannotated reports (Uzuner et al., 2011). The size of the training set accumulates to 30,673 sentences and 260,573 word tokens (Gurulingappa, Hofmann-Apitius, and Fluck, 2010).

In Table 2.3, we present the tagging scheme and models implemented as well as the performance of these systems. The table only includes information from publicly available papers.

TABLE 2.3: Named Entity Recognition results on medical texts. Precision (Pr), Recall (Re), and F1-score

Authors	*Pr*	*Re*	*F1*	*Tagging scheme*	*Model*
Bruijn et al., 2010	83.64%	86.88%	85.23%	Modified BIO	semi-Markov model
Kang et al., 2010	81.31%	81.10%	82.21%	POS	HMM and Linear-Chain CRF
Gurulingappa, Hofmann-Apitius, and Fluck, 2010	84.00%	80.00%	82.00%	BIO	CRF
Patrick et al., 2011	84.88%	78.92%	81.79%	POS	CRF
Jonnalagadda et al., 2012	83.20%	78.71%	80.89%	POS	CRF
Average	**83.40%**	**81.12%**	**82.42%**	N/A	N/A

2.5 Disambiguate the meaning of mentions

The process of linking words to a common understanding is referred to as **N**amed **E**ntity **D**isambiguation (NED) (Balog, 2018). In the following, we first present the process for creating a knowledge base, and hereafter approaches for establishing links between words and entities.

To make the linking possible, there is a need for establishing a solid knowledge base for the given domain. Essentially, this is the practice of creating an ontology that represents the global truth within a specific domain. The quality of the knowledge base depicts the system's ability to obtain meaning. That is, the environment of the application is limited by the level of knowledge contained in the ontology (Sowa, 1995).

In the creation of a knowledge base, it is important to consider the relationship between concepts in the ontology. In particular, hierarchical ontologies have shown to be useful in mapping associations between similar concepts (Khan and Safyan, 2014; Jiménez-Ruiz and Grau, 2011).

The knowledge base can also be referred to as a dictionary because the functionality constitutes a lookup with a query that returns a response (Balog, 2018). Seeing that the query may take different forms of writing, or different words are used in the search for getting the same response, the idea is to accommodate those different situations. Therefore, surface forms can be created on the dictionary itself to construct different versions of the content (Balog, 2018). Moreover, external resources can be used to augment the dictionary with synonyms or categories related to the content (Medelyan, Witten, and Milne, 2008; Cucerzan, 2007). Eventually, these techniques help to increase the possibility of finding a match in the dictionary.

Although the above strategies have been considered it might happen that the dictionary cannot find a related link. In such a situation, Cimiano, 2006 suggests that a fallback strategy is implemented.

When using the dictionary, the first task is to collect all candidates that match a given query (Balog, 2018). The querying of a word is often performed by string matching. The task can be approached by the usage of different relaxed matching techniques such as character dice score, skip bi-gram dice score, and Hamming distance (Dredze et al., 2010).

A look up in the dictionary may produce multiple results. Therefore, the next challenge is to create a heuristic such that the best matching entity is selected from all candidate links. The resulting collection of candidates may appear very large why it should be pruned to contain only the most relevant (Balog, 2018). An approach to this challenge is to rank the candidates based on the relevancy for the mention being queried. Balog, 2018 proposes three measures for this purpose.

First, one is to consider the commonness of an entity. This can be achieved by the use of some sort of statistics that can help to produce the popularity of entities (Medelyan, Witten, and Milne, 2008). From that, the relative probability of an entity being the right link for a specific mention can be acquired.

Then, a similarity measure is used to determine the contextual closeness between the description of a candidate entity and a given document in which the mention appears. There are several

approaches to calculate this where the most prominent are Cosine distance measure (Balog, 2018; C. Bunescu and Pasca, 2006), dot product, and Jaccard index (Kulkarni et al., 2009).

The measure of commonness and similarity is then combined in achieving a final confidence score, which is used to rank the collected candidate entities (Medelyan, Witten, and Milne, 2008; Balog, 2018). Having the reduced and ranked collection of candidate entities, the highest ranking entity can be chosen as the disambiguation for that mention (Balog, 2018).

However, if a document contains more than one mention, there is typically a relation between those mentions (Balog, 2018). Therefore, it might be advantageous to measure that relation before selecting the final links to the mentions. The strength of the relationship between candidates can be expressed by a coherence score (Cucerzan, 2007). The combination of candidates that achieve the highest coherence and confidence scores are chosen as the best disambiguation for the mentions in a document (Cucerzan, 2007).

2.6 Evaluation measures

The quality of systems designed for information retrieval is often assessed by the ability of the system to find correct sentence boundaries, entities or linkages. The most popular measures for making this assessment include precision (positive predictive value) seen in Equation 2.1, and recall (true predictive rate) seen in Equation 2.2 (Hand and Christen, 2018). The most common way for combining the recall, and precision measures is by the F-measure seen in Equation 2.3 (Christen, 2012; Christen and Goiser, 2007; Getoor and Machanavajjhala, 2012; Manning, Raghavan, and Schütze, 2010). The F-measure calculates the harmonic mean by use of the recall, and precision measures, which is useful as it regulates large values, and gives importance to smaller values (Sang and De Meulder, 2003b). To give a correct assessment of the performance, the F-measure provides the adjusted measure of the two, showing the relationship between the two in the F-measure.

$$Precision = \frac{\sum True\ positive}{\sum Predicted\ condition\ positive} \tag{2.1}$$

$$Recall = \frac{\sum True\ positive}{\sum Condition\ positive} \tag{2.2}$$

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.3}$$

Chapter 3

Methodology

In this section of the thesis, we first present the extraction and processing of the acquired data set. Hereafter, the approaches to the development of the NLP pipeline are presented.

3.1 Pre-processing of the data set

In this section, we present the processing of the data set after it was acquired. The ambition is to outline the composition of the data set and hereafter highlight modification and design choices taken in the process of preparing the data set for further processing.

3.1.1 GDPR and ethical considerations

Cf. Article 4(13), (14) and (15) and Article 9 and Recitals (51) to (56) of the **G**eneral **D**ata **P**rotection **R**egulation (GDPR), health-related data is perceived as sensitive personal data (Parliament and The European Union, 2016). According to Article 4(15), the protection of sensitive data related to natural persons must be preserved. The protection applies to both automatic and manual processing where processes must not lead to the disclosure of sensitive information. For that reason, the data set used in this study was at all times kept in a closed environment at a local server at the IT-University of Copenhagen. Moreover, cf. Article 9, it is prohibited to include processes revealing the unique person sensitive data. Therefore, the data set is used in a state where all entries have been de-identified, omitting the possibility of tracing specific medical personnel and patients. Moreover, examples of data points are in this paper presented with placeholders instead of actual personal values. First- and last names are represented by [Name], locations and addresses are represented by [LOC], and real values are represented by [X].

3.1.2 Description of the data model

The 323.122 patient EHRs are stored in a single MDF file that is kept on a local server at the IT University of Copenhagen. The MDF is a database file that contains information making up the 323.122 EHRs. The information is distributed among 23 tables in the "E4C-2010"-corpora. Table 3.1 shows the tables contained in the database file. Please see B.1 for a full mapping of relations in the database.

TABLE 3.1

DOCUREF	E4C_CLIENTDIAGNOSIS	E4C_CLIENTDIAGNOSISLOG
E4C_CLIENTDRUGSIDEEFFECT	E4C_CLIENTDRUGTABLE	E4C_CLIENTOBJECTIVE
E4C_CLIENTOPERATIONCODES	E4C_CLIENTTABLE	E4C_CLIENTVACCINATION
E4C_CLINICALDATA	E4C_CONSULTATIONSERVICES	E4C_DRUGDOSEHISTORY
E4C_EDIFACTMEDDISSEND	E4C_EDIFACTMESSAGERECEIVE	E4C_FAMILYRELATION
E4C_MEDICALRECORD	E4C_MEDICALRECORDLINE	E4C_MEDICALRECORDLINELOG
E4C_PRESCRIPTIONLINE	E4C_REFERRALRECEIVED	E4C_REMINDERTABLE
E4C_SERVICECLAIMS	EMPLTABLE	

3.1.3 Extracting the data set

The database file was attached to an active local Microsoft Windows Server 2017 (Windows, 2019). Each table mentioned in Table 3.1 were then extracted as a CSV flat file format. To accommodate the structure of the data, Table 3.2 highlights the settings that were applied to achieve a faultless extraction with no loss of data.

TABLE 3.2

Encoding (code page)	Latin_1 - ANSI 1252 - Latin1
Locale	Danish - Denmark
Format	Delimited
Text qualifier (quotechar)	'-'
Row delimiter	{CR}{LF}
Column delimiter (separator)	' | '
With headers	Yes

The "E4C_MEDICALRECORDLINE" is the only table used further on this project. This is because the table contains the actual medical texts in the field of "MEDICALRECORDTEXT". Table 3.3 presents the schema of a row in the "E4C_MEDICALRECORDLINE". The combination of all documents that are stored in this field constitutes the "E4C-2010"-corpora. The original table contained 7.548.240 rows. The contents of this table were modified in several steps. These are elaborated now.

TABLE 3.3: Properties of objects in the E4C_MEDICALRECORDLINE table

MEDICALRECORDID: Integer	**MEDICALRECORDTEXT:** String	**ICD10-ID:** String
ICPID: String	**MEDICALRECORDSUMMARY:** String	**MEDICALRECORDLINEID:** Integer
RECORDSUBCATEGORY: String	**TRANSDATE:** Date	**EXTERNALLETTERCATEGORY:** Integer
LINECATEGORY: Integer	**EMPLID:** String	**FORFOLLOWUP:** Boolean
HOKUSID: Integer	**DIAGNOSISCOMMENT:** String	**DIAGNOSISCATEGORY:** Integer
STARTTIME: Integer	**ENDTIME:** Integer	**MODIFIEDDATE:** Date
MODIFIEDTIME: Integer	**CREATEDDATE:** Date	**CREATEDTIME:** Integer
CREATEDBY: String	**DATAAERAID:** String	**RECID:** Integer

3.1.4 Pre-processing of the table E4C_MEDICALRECORDLINE

In the first processing step, the data set was split into two separate data sets based on the condition of whether the field "ICD10ID" was empty or not. This field indicates whether the record line has been labeled or not. This resulted in a division where one part contains 7.003.944 rows without a related ICD-10 code and 544.296 rows with an ICD-10 code. The ambition is to build a pipeline that is robust and can handle misspellings and noisy data. Therefore, no noisy data and misspelled words were handled.

However, some EHR contains less than 3 characters in the field "MEDICALRECORDTEXT". These entries were removed because it is unlikely that they contain any real information. Examples on this are "TK", "-", ".", " ".

3.2 Finding sentences in medical text

The first step in the pipeline concerns the initial processing of EHR in raw text. The idea is to segment every EHR corpus into a document representation taking the form of a bounded sentence. This segmentation is introduced to streamline and normalise the input for future processes. In this way, the later processing is performed at a uniformed level where all sentences from the *"E4C − 2010"*-corpora adopt the same general linguistic format. In the following, we present this process and the applied tools.

3.2.1 The composition of the data

At this point in the pipeline, this is the first instance where we experience the complexity of the "E4C-2010"-corpora. This is evident when examining the writing style and the use of different languages that are not complying with the traditional structure of the Danish language.

A critical issue is that the EHRs contain complex, irregular usage of symbols, ordinal numbers, ellipses and in general, inconsistent writing. In Table 3.4, we present examples of some apparent writing styles conceived as noisy data.

TABLE 3.4: Examples of noise encountered in the "E4C-2010"-corpora

Concept	**In-text example**	**Challenge**
Measurements	"BS 5.6", "TP 36.4 småt øreterm"	The punctuation mark cannot function as a sentence boundary, but might be perceived as being so
Drugs	"Selo-Zok.", "Rp T.Sparkal"	Hyphen or punctuation included, but these characters do not constitute a sentence boundary
Enumerations	"Medicin ved udskrivelsen: Kinin 200mg x 1. Magnyl 75mg x 1. Furix 40mg x 1."	This is an enumeration indicating the drug prescriptions and is in fact one sentence, but the multiple usage of punctuation marks presents candidates for sentence boundaries
Dates and time	"05.10.2008", "14.00"	Includes punctuation that indicates sentence boundary, but is actually not
Ordinal numbers	"Obs uspecifik vaginitis Metronidazol 2 g 1. og 3. dag"	The punctuation mark is used for marking ordinals of days and is not a sentence boundary
Ellipses	"Tumor, der ligger subkonjunktivalt sandsynligvis præ..... i nedre øjenlåg"	Ellipses indicating omission of words or thought, but includes numerous punctuation marks indicating a sentence boundary
Document structuring using characters	"*** Lægevagtsnotat / epikrise ***"	Title markup where sentence is bounded by three times "***"
Structure of discharge summary	"_____"	Multiple, consecutive hyphens used for highlighting structure of document and has no relations to previous and preceding sentences
Consecutive exclamation points and question marks	"!!!", "??"	These points and marks normally indicates sentence boundary, but there is not a new sentence for each consecutive mark or point

Another critical issue discovered is the abbreviations found in the text. It is clear that clinical personnel create sentences with a wide range of abbreviations that are not only derived from the Danish language. Other uses are Latin and English. For that reason, this study cannot approach the issue of abbreviations by simply relying on a list of known Danish abbreviations. Consequently, sentence boundaries are not easily recognised. That is, no particular character or word consistently provides certainty that a sentence ends or begins. Instead, there is a need for creating a solution accommodating the complex and less obvious composition of the sentences. In Table 3.5, we highlight the most common abbreviation-related obstacles experienced during this examination.

TABLE 3.5: Examples of abbreviations encountered in "E4C-2010"-corpora

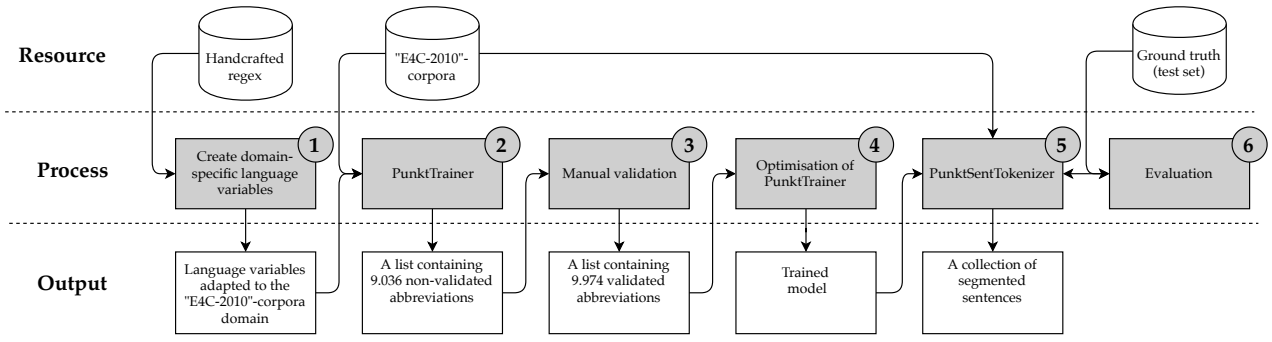
In-text example	**Challenge**
"Patient er sat i beh."	Abbreviations are used in the end of sentences. Such period is a part of the abbreviation while also constituting a sentence boundary.
"Tab. Norvasc 5mg", "Tabl. Cozaar"	Certain abbreviations are frequently followed by a capitalised letter, but the final period rarely constitutes a sentence boundary.
"Der gives endvidere brusetabl. til natten", "Desuden forlydender om opmærksomhedsprbl. hos [NAME] bror"	Abbreviations used in the medical domain does not follow the general domain of the language and some medical personal make up their own abbreviations. Also, abbreviations might be long.
"Smerter i hofteregion f.eks hvis pt. ligger på ve. side"	Some common Danish abbreviations are used, but not necessarily in correct spelling. Here the token "f.eks" ("for eksempel") is missing the final period.
"Obj. auris dxt./sin. i.a"	Foreign abbreviations are common in medical texts and are used among other Danish abbreviations. Here the Latin tokens "sin." (sinister, left) and 'dxt.' (dexter, right).
"Kendt dolores extr. siden [YEAR]"	Many medical definitions are written in a short-hand abbreviation. In this example the token "extremitatis" is abbreviated to "extr.".

3.2.2 Developing a tool for finding sentences in Danish EHR

Based on the observations made in the examination of the data, it is already now possible to discard the possibility of solely relying on regular expressions to detect sentence boundaries. This is because no global rule-set exists for abbreviation usage in the sub-domain of EHR in Danish. Moreover, since many of the sentences begin with a lower-case token there is a need for additional knowledge to identify the sentence boundaries. In the following, we present the approach for establishing a global rule-set that accommodates the specific structure of the "E4C-2010"-corpora.

In this study, we follow the Punkt System for unsupervised multilingual SBD proposed by Kiss and Strunk, 2006. In doing so, we use the implementation PunktSentenceTokenizer published by the NLTK (Bird, Klein, and Loper, 2009). We used three components from this package; PunktLanguageVars, PunktTrainer and PunktSentenceTokenizer. In Figure 3.1, we illustrate our process to disambiguate sentence boundaries.

FIGURE 3.1: Visualisation of implemented SBD process



As visualised in Figure 3.1, steps 1 to 4 include preparation and training of the model to be later used for splitting sentences in steps 5 to 6.

The first process (Step 1) regards the use of the component PunktLanguageVars (Bird, Klein, and Loper, 2009). The task is to create the foundation for finding candidate abbreviations and sentence boundaries. These are found using a set of regular expressions. Initially, the model considers each candidate sentence boundary and abbreviation. In later processes, ellipses and hyphens are considered. For that purpose, three different language and domain-specific expressions are formulated:

- Candidates for sentence boundaries ('.', '?', '!', '***') are captured using `(!|\.|\?|\*{3,})`. This expression simply searches for any of the defined boundary characters
- Multi-character punctuation such as ellipses and hyphens are captured by `(\-{2,}|\.{2,}|(\.\s){2,}\.)` to discard such as sentence boundaries.
- Internal punctuation being the special characters within a sentence (':', ';', ',') is captured by the expression `(,|;|:)`. This expression finds potential abbreviations by considering period-final tokens that precedes these characters as such

The next process (Step 2) makes use of the PunktTrainer component (Bird, Klein, and Loper, 2009). This step takes its offset in the defined language variables created in the previous process (Step 1). The PunktTrainer runs in two stages. In the first stage, the PunktTrainer uses the language variables to propose an annotation of abbreviations and sentence boundaries in the training set. This is done by computing statistics on token collocation bonds, token lengths, internal periods, and occurrences without a final period. These statistics are referred to as global evidence (Kiss and Strunk, 2006). Then, in the second stage, the output from the first stage is corrected and further annotated. This involves the detection of abbreviations and ellipses in the ends of sentences, initials, and ordinal numbers. However, these are not easily captured using global evidence why the local context of each token is considered. Based on these tuned features, the unsupervised model is used to elicit all candidate abbreviations from $1,000,000$ random EHRs from the "E4C-2010"-corpora. The preliminary product of this process was a list of 9.036 candidate abbreviations.

The following process (Step 3) includes intervention by the researchers to manually validate the output from the previous step. This step involves the inclusion and removal of abbreviations that the model did not encounter or wrongly collected as candidates. Meaning that the model is forced to acknowledge a given token as an abbreviation because it was left out of the candidates. This often applies when - as discovered in the examination - capital letters follow a punctuation mark, but the sentence is, in fact, one and the punctuation mark has been wrongly used for enumeration. The output of this process is a modified list of $9,974$ approved Danish abbreviations used in medical texts.

The next process (Step 4) is an intermediary process where the approved list is fed back to the model. Moreover, arriving at this point in the process, we decided to delete the statistics of sentence start boundaries obtained by the model. This is because the model did not discover any trending patterns in the data, which diminished the possibility of generalising.

Arriving at this point in the process (Step 5), the PunktSentenceTokenizer component is used (Bird, Klein, and Loper, 2009). This component handles the final segmentation of sentences in the corpora based on the pre-trained model from the PunktTrainer. The output of the model is then the segmented sentences.

3.2.3 Evaluation of the model

To evaluate the performance of the SBD model, it is popular to look at the correct and incorrect labels of sentence boundaries. To categorise whether a boundary is correct or not, there is a need for creating a ground-truth.

In this study, we constructed a ground-truth by manually annotating a test set. The process was to first select and remove one random EHR from the *"E4C − 2010-corpora*. Then, the document is manually segmented into sentences. These sentences are then added to a reference document where all sentences are separated by a new line. The process continues until all sentences used for evaluation have been created. This reference document constitutes the ground-truth used for evaluation of the SBD.

To prepare the test set, a copy is made of the original test set and modified by removing all line breaks. Essentially, the copy then becomes a one-line document without any sentence breaks. This copy is segmented using the developed SBD-model and the output is compared to the reference document containing the ground-truth on correct sentence boundaries.
The SBD is evaluated by adopting the proposed technique from Newman-Griffis et al., 2016 including the counts of **T**rue **P**ositive (TP), **F**alse **P**ositive (FP), and **F**alse **N**egative (FN). These measures are presented in a confusion matrix, allowing the computation of precision-score, recall-score and F1-measure (Newman-Griffis et al., 2016). **T**rue **N**egative (TN) are usually included in the matrix as well but trivially does not provide any information for this evaluation. That is, a TN represents the number of situations where the SBD-model correctly disambiguate a point as not being a sentence boundary. However, the inclusion of TN distorts the F1-measure in the sense that the majority of characters are not representing a sentence boundary. This would reward the SBD-model with a higher F1-measure because it will capture a high number of non-boundaries. Therefore the count of TN is left out. TP is counted whenever a sentence boundary from the reference document is also found by the SBD-model. FP are counted when the model produces a sentence boundary that was not present in the reference document. Finally, FN represents situations where a sentence boundary in the reference document is not produced by the SBD-model. The composition of the test set is found in Table 3.6.

TABLE 3.6: Composition of test set for Sentence Boundary Disambiguation

Sentences	100
Characters	4,764
Boundary characters	200
Non-boundary characters	4,564

Also, we wish to compare the performance of the trained model from this study with the default distributed Danish model from the NLTK (Bird, Klein, and Loper, 2009).

3.3 Finding words in a sentence

To locate words in sentences, we use a simple technique for the tokenisation of words. The off-the-shelf solution, *word_tokenize*, from the NLTK was used. The tokenisation is performed on basis of the Penn Treebank (Bird, Klein, and Loper, 2009). The implementation is configured to produce a consistent output in each sentence. That is, certain characters are separated from the beginning and end of tokens. This involves punctuation characters being distanced from real words while single quotes and commas are only split when being followed by a white space character. Also, every last period in a sentence (the end boundary character) is separated from the final token in that sentence. In contrast, hyphens inside tokens (not wrapped by white-space on any side) are kept to maintain the token-token dependency in e.g. drug names. In this way, a given input sentence is tokenised and outputs a sentence as an ordered list of word tokens.

3.4 Finding disease-related words

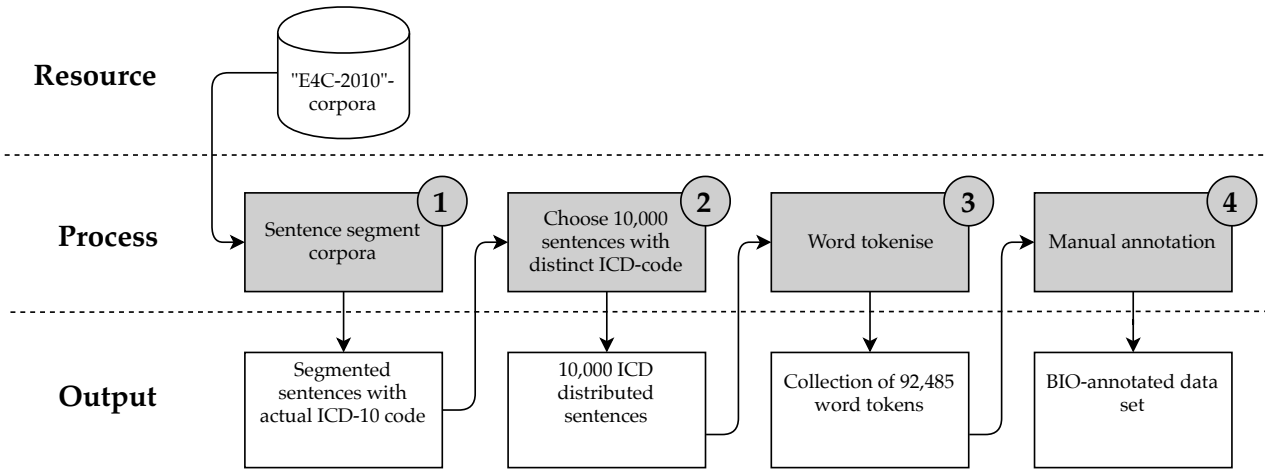
In this study, we wish to elicit specific disease mentions from texts why there is a need for applying a tool to assist the labelling of specific tokens in a sequence. For this purpose, we use the Stanford NER Classifier to create a Linear-Chain CRF-model (Stanford-Natural-Language-Processing-Group, 2019). To facilitate the use in sub-domain of Danish EHR, we have undertaken a wide range of steps towards creating the most optimal configuration of the CRF. In the following, we present our approaches to encoding, optimisation, and decoding of the CRF-model.

3.4.1 Encoding of the Conditional Random Fields

Process for creating knowledge base

To accommodate the encoding process of the CRF-model, there is a need for creating a data set for training the model. This process is illustrated in Figure 3.2.

FIGURE 3.2: Visualisation of process for creating data set for training



The first process (Step 1) is to merely segment the corpora into sentences. This is done by using the developed SBD-component. Each sentence is accompanied by the ICD-10 label categorising the parent EHR from where the sentence originates.

The next process (Step 2) is to elicit a portion from the collection of segmented sentences for the knowledge base. The idea is to derive a collection of sentences that provides a representation of the full "E4C-2010"-corpora. This is done under the assumption that the language and construction of sequences in the EHR are dependent on the topic of the document. For that reason, all ICD-codes present in the corpora are to be included in the data set for training the model. However, this is not providing a correct statistical distribution as ICD-codes with a higher frequency in the corpora are not included the equivalent times in the knowledge base. Nonetheless, we found this as useful because it means that the data set will represent a wide extract of different language structures from the corpora. In this process, we wish to extract 10,000 sentences for the training set. However, in the "E4C-2010"-corpora there are 8,698 distinct ICD-codes. Therefore, we extracted 8,698 sentences with distinct ICD-codes, and 1,302 randomly selected sentences.

The subsequent process (Step 3) is to tokenise each sentence into word tokens. The component of Word Tokenisation is used for this purpose where each sentence is returned as a list of word tokens.

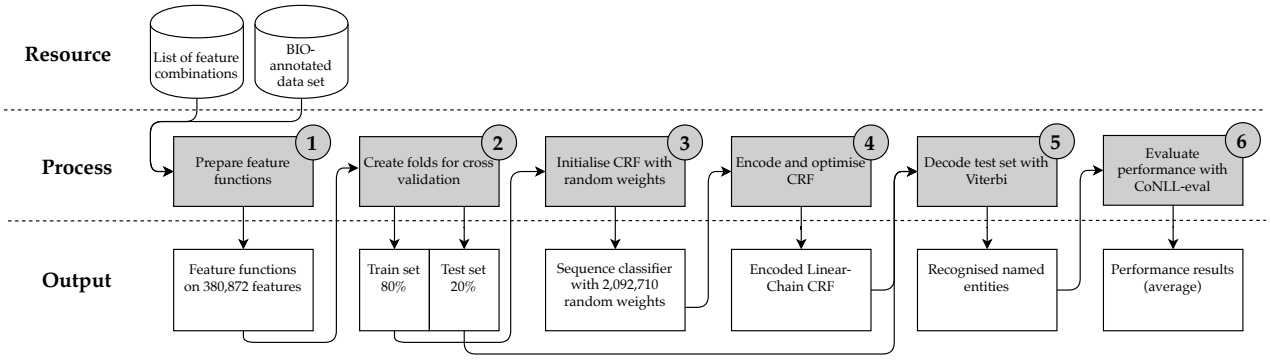
The list of word tokens is appended to a TSV file with an empty line separating all tokens in one sentence from the next. The TSV file furthermore contains a subsequent column for labelling in the next step. The output from this process is a TSV file containing 92,485 rows in a word - answer format and 10,000 blank lines representing the sentence boundary.

In the final process (Step 4), we manually annotate the derived tokens using the BIO-scheme (Ramshaw and Marcus, 1995). To support this activity, we explored several medical encyclopedias such as www.Sundhed.dk (2019), www.Wikipedia.dk (2019) and www.Medinfo.dk (2019) for categorisation of diseases, drugs and so forth. The final output of this process is a BIO-annotated data set containing 92,485 manually annotated word tokens.

Process for creating model

Having prepared the data for training, the creation of the CRF-model can begin. This process includes various sub-tasks to acquire the most optimal construction of the model and is illustrated in Figure 3.3. In the following, we present each task accordingly.

FIGURE 3.3: Visualisation of process for creating CRF-model



The first process (Step 1) is to make a selection of features that encapsulates the structure of the medical texts in the Danish EHRs. This selection serves to provide the training sequence with a focus on sub-sequences that are to be learned from. By selecting features, we are specifying the relevant areas for the model to extract knowledge from. The input to this process is a list of token features focusing on different properties of the word tokens. The combination of token properties is numerous why we applied a trial/error approach where a higher number of combinations for tuning the feature parameters were tested. This was done to find the best combination of feature parameters leading to the final list of properties used as input in Step 1. In Table 4.5, we present the results of these experiments. The final set of features includes n-grams, word shapes, distributional similarity measures, and dictionary look-ups.

The list is composed of features that represent two types of properties. State features are used to include individual characteristics of the word while transition features emphasise the relationship between tokens in the structure of a sentence.

In the CRF model, these features are represented as feature functions (Chang and Sung, 2005). The conception used in this study includes the use of multiple parameters in the general form :

$$f(X, i, y_{i-1}, y_i) \tag{3.1}$$

where y_i correspond to the label of token i in the document X. Some features rely solely on the label while others are computed using all parameters.

In the following, we present our implementation of features in the CRF.

Character-based N-gram features The character n-grams such as uni-gram, bi-gram, and tri-grams of the tokens are used to represent a correlation of prefixes or suffixes in tokens (Cavnar and Trenkle, 1994). To capture this relation, the character n-grams are utilised for feature extraction:

$$n(w) = n - grams\ in\ w \tag{3.2}$$

where $n(w)$ represents the set of possible n-grams in a token, w. For the creation of n-grams, a minimum length of 1, and the maximum length of 5 were configured. This feature produces prefix n-grams, middle n-grams, and suffix n-grams (Cavnar and Trenkle, 1994). Moreover, each word token, w, is included in its full form. This is to provide information on word tokens that exceed the maximum n-gram threshold. One example is the token, w, for the word *pest*:

$$n(w) = p, e, s, t, pe, es, st, pes, est, pest$$
$$w = pest$$

This example makes it clear that the word feature, w, is already present in the n-gram feature, $n(w)$, since the token, *pest*, is of shorter length than the maximum length threshold. This is not the case with a word such as *byldepest*. The potential of n-grams in the perspective of feature functions is also evident in this example when considering other tokens such as *byldepest* or *hudpest*. In that case, the n-grams produced overlap and thereby allows for consideration of relationships between those tokens.

Word-shape features

Another feature included in the model is the token word-shape. This concept provides a normalisation of the orthographic features in each token (Manning and Schütze, 1999). The class of the word shape is then used as a feature:

$$s = shape \tag{3.3}$$

The combination of potential classes for the word shape is a process of testing. Therefore, we tested a great number of combinations for classifying word shapes and continued with the best fitting combination in our feature function. The feature function is created using 4 different classifications of shapes for each token. These word-shapes classes are:

- "ALL-DIGITS" for tokens that consist entirely of digits
- "ALL-UPPER" for tokens that are all upper cased characters
- "ALL-LOWER" for tokens that are all lower cased characters
- "MIXED-CASE" for tokens that are capitalised (every character but the first is lower cased)
- "OTHER" for tokens that satisfy none of the above class constrains, e.g. tokens that are not capitalised but contain upper case characters

An example of this feature applied to the sentence "Han har svær Parkinsons" is the word shape class $"MIXED - case"$ for the token *Parkinsons*.

Distributional similarity class features The distributional similarity is used as a feature following the assumption that semantically similar items also have similar meanings (Manning and Schütze, 1999). The *distSim* feature expresses what word cluster a given word belongs to. In this study, we approach the generation of clusters in two ways; Brown clustering (Derczynski and Chester, 2016) and GloVe (Pennington, Socher, and Manning, 2014). The reason for testing the impact of two different approaches is that the marginal performance may vary depending on the domain. Therefore, we wanted to explore which of the approaches that are best fitting the Danish medical texts. In the following, we present our approaches for creating clusters.

Brown word clustering is a renowned technique, vastly used within the field of NLP (Derczynski and Chester, 2016). The idea is to learn the representations of words from bi-gram mutual information (Van Rijsbergen, 1977) and from that construct a binary hierarchy over the input words (Derczynski and Chester, 2016). The assumption behind this technique is that similar words appear in similar contexts and have a similar distribution of words to the left and right. Brown Clustering is a greedy, hierarchical, agglomerative hard clustering algorithm used to divide a vocabulary of words into a set of clusters with minimal loss of mutual information (Brown et al., 1992). The algorithm operates with the objective of creating a given number of predefined clusters where the output clusters are organised as leaves of the binary tree.

Traditionally, the number of clusters is predefined as the window size. The size of the window depicts the number of words to consider in the first iteration. Here, the top frequent words within the size of the window are put into distinct clusters. Then, the next most frequent word is added as a leaf-node before the algorithm tries to merge two clusters based on the lowest decrease in **A**verage **M**utual **I**nformation (AMI). This process continues until every word in the vocabulary has been assigned to a cluster (Derczynski and Chester, 2016). To formally annotate Brown clustering:

Definition 3.4.1 *Let S denote an input sequence and let V_s denote the unique words in S (i.e., the vocabulary), sorted by descending frequency. Then, the k'the symbol in V_s is denoted by $V_s[k]$, the. A cluster, C_i, is a subset of V_s and all clusters are disjoint (i.e. $C_i \cap C_j \neq \emptyset \Rightarrow i = j$). A complete clustering of the vocabulary is a set of clusters $C = C = \{C_0, ..., C_{|c|-1}\}$ that is complete $\cup C_i = V_s$. Adjacent symbols (i.e. bigrams) in S are denoted $\langle l, r\rangle$ and the relative frequency of $\langle l, r\rangle$ in S is denoted by $p(\langle l, r\rangle)$. Further, let $p(\langle l, *\rangle) = \sum_{r \in V(s)} p(\langle l, r\rangle)$ and $p(\langle *, r\rangle) = \sum_{l \in V(s)} p(\langle l, r\rangle)$. Analogously, adjacent symbols from C_i and C_j are denoted by $\langle Ci, C_j\rangle = \sum_{l \in C_i, r \in C_j} \langle l, r\rangle$ and the relative frequency in S of $\langle C_i, C_j\rangle$ as $p(\langle C_i, C_j\rangle)$. Finally, let $p(\langle C_i, *\rangle) = \sum_{l \in C_i p(\langle l, *\rangle)}$ and $p(\langle *, C_j\rangle) = \sum_{r \in C_j} p(\langle *, r\rangle)$.*

Then, the mutual information of two classes, $C_i, C_j \in C$, denoted $MI(C_i, C_j)$ is denoted as:

$$MI(C_i, C_j) = p(\langle C_i, C_j\rangle) log_2 \frac{p(\langle C_i, C_j\rangle)}{p(\langle C_i, *\rangle) p(\langle *, C_j\rangle)} \tag{3.4}$$

As mentioned, the merge of clusters is performed on the pair of clusters that have the lowest decrease in AMI. The AMI of C is the sum of mutual information of all pairs of clusters in C:

$$AMI(C) = \sum_{C_i, C_j \in C} MI(C_i, C_j) \tag{3.5}$$

To increase the performance of the merging operation, we adopted the approach of Generalised Brown proposed by Derczynski and Chester, 2016. This technique facilitates the separation of the set size from the number of output clusters. This approach allows the generation of multiple cluster size combinations from the same tree as long as $|C| < a$. In the produced clusters, each cluster class is represented by an identifier expressed as a bit string. In the generation of Brown Clusters, active

set sizes of 2,500 and 5,000 are used.

Another approach is GloVe proposed by Pennington, Socher, and Manning, 2014. This method seeks to create a semantic vector by representing each word with a real-valued vector. GloVe is a log-bi-linear model with a weighted least-squares objective that makes efficient use of statistics. The overall idea is to observe the ratios of word-word co-occurrence probabilities as these are conceived to contain some form of meaning (Pennington, Socher, and Manning, 2014). Therefore, the first step is to create a co-occurrence matrix from all the words in the corpus. The co-occurrence between two words is measured by the frequency of occurrence of one word in the context of the other (Pennington, Socher, and Manning, 2014). This matrix is then modified to include a single vector for each word. The vector contains the relative probability to all other words in the corpora. This probability is calculated by the objective function:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - logX_{ij})^2 \tag{3.6}$$

where $w \in \mathbb{R}^d$ are word vectors, $\widetilde{w} \in \mathbb{R}^d$ are separate context word vectors, b_i is a bias for w_i, and $\widetilde{b}_j$ is a bias for $\widetilde{w}_j$.

By maximisation of the objective function, the most likely words are captured to determine similar semantic meaning (Pennington, Socher, and Manning, 2014). It essentially comes down to answering the question of "word a is to word b as word c is to X" by finding the word d representation that is closest to $word\,b - word\,a + word\,c$ by the use of cosine similarity (Pennington, Socher, and Manning, 2014).

From the word vectors created we used the clustering algorithm of K-means to induce the distributional similarity clusters (Han, Pei, and Kamber, 2011).

Common for both approaches is that the configuration of hyper-parameters has an impact on the contribution of each type of clusters. Therefore, we applied the approach of grid-search to tune the relevant hyper-parameters including the number of output clusters, the minimum occurrence of words, and finally, the active set size (Bergstra and Bengio, 2012). The output number of clusters is important because it has an impact on the purity of clusters. Purity expresses the number of different classes in one cluster. In other words, how ambiguous the clusters appear to the model when searching for similar words to extract meaning (Han, Pei, and Kamber, 2011). The minimum occurrence is used to sort out words that are not conceived to contain substantial information due to its low frequency in the corpora. Minimum occurrences of 0, 3, and 10 have been used during the test phase. The best configuration was found to be a Brown clustering approach with an active set size of 5,000 and with a minimum occurrence of 3. Our best results were achieved when the number of merged output clusters is set to 900. In Table 4.9, we present the results of the tuning of hyper-parameters. All clusters are constructed on the full "E4C-2010"-corpora.

An example from this study is the word token *synlig* where the distributed similarity class in a Brown cluster is:

$$distSim = 00000011001110$$

Within this specific cluster, examples of other words are [’påviselig’, ’tegnpå’, ’tegntil’, ’synl’, ’synbar’, påvisbar, akutkrævende].

Gazetteer match features To compensate for words that are not apparent in the "E4C-2010"-corpora, an additional dictionary was constructed on the SKS-database. This allows the CRF-model to perform additional look-ups in the search for the correct tag from the BIO-format.

The dictionary was constructed by considering each disease description from the SKS-database as a sentence. This sentence is then lower-cased and saved in the dictionary with the tag *"I"* for inside. Then, the sentence is tokenised into words. The first word token in each sentence is then considered as the beginning and is added to the dictionary with the tag *"B"*. Then, if the description is multi-token, the remaining words are tagged by conceiving the first word as a beginning, *"B"*, and the remaining words as insides, *"I"*. This token-tag pair is then saved to the dictionary. This approach gives rise to the issue of a single token having more than one possible tag. This is clear when considering the following two descriptions of different entities "(A20) - Pest" and "(A207) - Septisk pest". In the first example, only one token is tagged, $(Pest = B)$, while the second allows for two tags, $(Septisk = B), (pest = I)$. This functionality allows for the token *pest* to be considered in feature functions relaying on both *I*- and *B* tags. In this way, a feature, $g(w)$, is added:

$$g(w) = gazette\ entries\ matching\ w \tag{3.7}$$

where $g(w)$ is a list of partial matches in the gazetteer. A partial matching technique is used when performing a look-up in the dictionary. That is, any description from the dictionary that contains the given word token results in a match. In this way, the sequence of tokens inside the disease description is also considered when tagging beginning and insides of new diseases.

Proper name features To further enrich the construction of the CRF-model, the feature for considered a word token as a proper name is included. To facilitate the use of this features, there is a need for a knowledge base containing proper names. This knowledge base is created using the publicly available lists of approved names in Denmark. These lists are published and maintained by Ankestyrelsen (Økonomi- og Indenrigsministeriet, 2019). This study has utilised three lists from this source containing names on male, female, and last names. Moreover, the list of male, and female names are enriched by a list of 1054 names that applies as both male, and female names. The list of male names contains 18249 names, the female names list contains 22305 names, and the list on last names if of size 197 (Økonomi- og Indenrigsministeriet, 2019).

Much like the distributed similarity represented a word cluster class, the feature for names is used as a class feature:

$$name(w) = FEMALE\_NAME||MALE\_NAME||LAST\_NAME \tag{3.8}$$

where the token w, is either a last name, male or female name, or not represented in any of the name lists.

Combination of features These feature functions are then utilised on local states considering just the token. In addition, each feature is also utilised to capture sequential transitions. That is, every feature for a token is considered in the context of its sentence to represent the relation between tokens. An example of a feature function for this concept is the word shape (3.4.1) class feature. Such a function for example considers the previous token, e.g. $ALL - DIGITS - TPS$, where the feature is based on the previous token word shape class (in this case 'ALL-DIGITS'). Furthermore, a disjunctive sequential feature function is also utilised. This function considers the absence of

features in one direction. An example of such feature is $Contusio - DISJP$, where the word feature (3.4.1) considers the lack of the token *Contusio* in the previous direction in the sequence of tokens.

The feature functions results in an average of 380,872 features being produced.

Having determined the features to be used and created the related feature functions, the next process (Step 2) is to prepare for cross-validation of the system. This process is also marking the start of an iterative process from Step 2 to Step 6 where the cross-validation accommodates the CoNLL-Eval (Step 6). The cross-validation is introduced to provide a less optimistic estimate on the performance of the NER system. This process splits the data set in two parts where 80% and 20% is allocated for the training set and test set respectively.

The following process (Step 3) is to instantiate the Linear-Chain CRF model (Sutton and McCallum, 2010; Lafferty, McCallum, and Pereira, 2001) defined as:

Definition 3.4.2 *Let Y, X be random vectors, $\theta\{\theta_k\} \in R^K$ be a parameter vector, and $\{f_k(y, y', x_t)\}_{k=1}^K$ be a set of real-valued feature functions. Then a linear-chain conditional random field is a distribution $p(x|y)$ that takes the form:*

$$p(x|y) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right\} \tag{3.9}$$

where $Z(x)$ is an instance-specific normalisation function, and T is the length of the input sequence.

$$Z(x) = \sum_{y} \prod_{t=1}^{T} exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t) \right\} \tag{3.10}$$

The model is constructed on a fold from the data set acquired in the previous step including 8,000 documents (4 folds of 2,000 sentences) with an average size of 74,138 word tokens with one of three possible labels; B, I, O. This leads to an average of 380,872 features and 2,092,710 random weights in the untrained sequence classifier. The amount of word tokens and thus features and weights varies according to the composition of folds in the given iteration of training.

The untrained CRF from the previous process (Step 3) is in this step (4) subject for optimisation. The ambition is to train the model to its optimal point for future use of predicting correct label sequence Y when presented data sequence X. To train the Linear-Chian CRF, the goal is to determine the parameters $\theta = (\lambda_1, \lambda_2, ...; \mu_1, \mu_2, ...)$ from training data $D = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{N}$, where each $x^i = \left\{x_1^{(i)}, x_2^{(i)}, ...x_T^{(i)}\right\}$ is a sequence of inputs, and each $y^i = \left\{y_1^{(i)}, y_2^{(i)}, ...y_T^{(i)}\right\}$ is a sequence of desired label sequences with the empirical distribution:

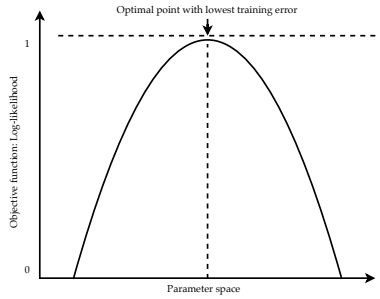
$$p^{\sim}(x) = \frac{1}{m} = \sum_{i=1}^{m} \delta(\mathbf{x} - \mathbf{x}^{(i)}). \tag{3.11}$$

To determine the parameters, the CRF is trained by maximising the log-likelihood function, meaning that the parameters are selected such that the data from the training set has the maximum probability under the model (Sutton and McCallum, 2010; Lafferty, McCallum, and Pereira, 2001). The maximum log-likelihood is used in an attempt to match the distribution of the model with the

empirical distribution such that the dissimilarity between the two is minimised (Goodfellow, Bengio, and Courville, 2016). The objective function is defined as the log-likelihood function where the ambition is to maximise (Sutton and McCallum, 2010; Lafferty, McCallum, and Pereira, 2001):

$$\ell(\theta) = \sum_{i=1}^{N} log p_\theta(y^{(i)}|x^{(i)}) \tag{3.12}$$

FIGURE 3.4: Toy example of the relationship between log-likelihood and parameter optimisation



Then, we initiate the learning phase that maximises the log-likelihood function by optimisation of parameters. The optimal point for training is illustrated in Figure 3.4. In this study, we apply the Quasi-Newton method of **L**imited Memory- **B**royden- **F**letcher-**G**oldfarb- **S**hanno (L-BFGS) (Byrd et al., 1995) as the learning algorithm. The goal of this process is to accommodate situations where the CRF may experience several training examples with the same input x, but with different values of label sequence y (Sutton and McCallum, 2010). Therefore, the model needs to compute and fit the distribution $p(y|x)$ to all different values y that are compatible with x. The L-BFGS uses the Quasi-Newton technique of making a quadratic approximation to the log-likelihood function where it seeks to find a global optimum. In doing so, the L-BFGS makes use of the first- and second order derivatives of the objective function to perform the approximation.

To obtain the first-order partial derivatives, we compute the gradient of each edge in the CRF by using the *Forward-Backward* algorithm (Sutton and McCallum, 2010; Goodfellow, Bengio, and Courville, 2016). The objective of the algorithm is to compute the marginal probability of an edge. The idea is to first define a set of *forward variables* α_t where each is a vector size of M (where M is the number of states) (Sutton and McCallum, 2012):

$$\alpha_t(j) =^{def} p(x_{\langle 1\ldots t\rangle}, y_t = j) \tag{3.13}$$

$$= \sum_{y_{\langle 1\ldots t-1\rangle}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \tag{3.14}$$

Then, the alpha values can be computed by recursion:

$$\alpha_t(j) = \sum_{i \in S} \Psi_t(j, i, x_t)\alpha_{t-1}(i) \tag{3.15}$$

Now, the backward probability by recursion is very similar and is defined as:

$$\beta_t(i) =^{def} p(x_{\langle t+1 \dots T \rangle} | y_t = i) \tag{3.16}$$

$$= \sum_{y_{\langle t+1 \dots T \rangle}} \prod_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}) \tag{3.17}$$

with the recursion of:

$$\beta_t(i) = \sum_{j \in S} \Psi_{t+1}(j, i, x_{t+1}) \beta_{t+1}(j) \tag{3.18}$$

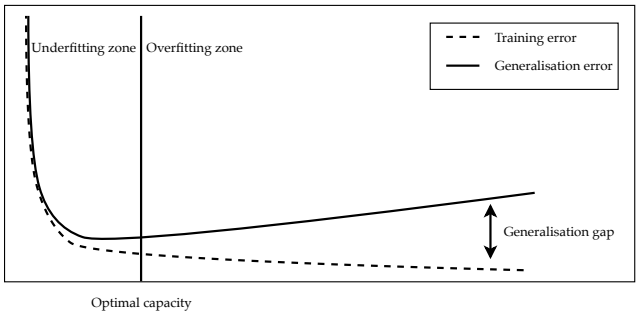
From the first-order partial derivatives, we are now able to compute the second-order partial derivatives needed for the Quasi-Newton optimisation (Sutton and McCallum, 2010). With the L-BFGS, we seek to make an approximation to the inverse Hessian matrix, H^{-1}. We do not wish to compute the complete inverse Hessian matrix because it imposes a high demand for memory (Goodfellow, Bengio, and Courville, 2016). Instead, the L-BFGS constructs an approximation, M_t, to inverse Hessian matrix, H^{-1} (Goodfellow, Bengio, and Courville, 2016). In each iteration, the algorithm seeks to better its approximation of H^{-1} by making low-rank updates to M_t (Goodfellow, Bengio, and Courville, 2016). In this way, the algorithm includes some knowledge about the inverse Hessian matrix by storing a given amount of the vectors that were used to update M_t at each time step (Stanford-Natural-Language-Processing-Group, 2019; Byrd et al., 1995). The size of past guesses is set to 20 in this project as it was found adequate when regarding the running time and memory usage of the algorithm.

The process of the L-BFGS is to perform iterations over a collection of line searches with the given direction determined by the gradient descent $p_t = M_t g_t$. Then, the inverse Hessian approximation M_t is updated and the new gradient is determined for depicting the direction of the line-search in the subsequent iteration. The L-BFGS will then perform a line-search in this direction to depict the size of the step ϵ^* (Byrd et al., 1995). Each iteration leads to an update of the parameters to change the gradient for the subsequent line search:

$$\theta_{t+1} = \theta_t + \epsilon^* p_t \tag{3.19}$$

In training the model, the stated purpose is to minimise the training error achieved by maximising the log-likelihood function. When doing so, the model improves its ability to capture the truth of the training set. However, if the model only encapsulates the truth from the training set, the model may not be able to correctly predict data points that have not been presented to the model before. Therefore, the model will at this point only be able to predict on the partial truth obtained from the training set (Goodfellow, Bengio, and Courville, 2016). Therefore, we modify the model to focus on the most important features and diminish the focus on less significant features to improve the ability to predict unseen data points. In the context of training the model, this corresponds to finding the optimal point where the model is neither under- or over-fitting data points (Goodfellow, Bengio, and Courville, 2016). We visualise this relationship in Figure 3.5

FIGURE 3.5: Visualisation of relationship between training- and generalisation error (toy example)



To encounter the issue mentioned above, we apply the L^2 regularisation term (Goodfellow, Bengio, and Courville, 2016). This regularisation technique enables the model to capture the most significant features from a space of many interrelated features (Goodfellow, Bengio, and Courville, 2016). The regularisation is achieved by adding a restriction to the objective function. The restriction is represented by a norm penalty $\Omega(\theta)$ where the objective function is denoted as $\widetilde{J}$ (Goodfellow, Bengio, and Courville, 2016). The L^2 regularisation forces weights closer to the origin by targeting parameters with low co-variance to the output of the objective function (Goodfellow, Bengio, and Courville, 2016). The L^2 regularisation term is defined as $\Omega(\theta) = \frac{1}{2} \parallel w \parallel \frac{2}{2}$. The objective function with L^2 regularisation term is then defined as:

$$\widetilde{J}(\theta; X, y) = \sum_{i=1}^{N} log p_\theta(y^{(i)}|x^{(i)}) + \alpha\Omega(\theta) \tag{3.20}$$

where $\alpha \in (0, \infty)$ is a hyper-parameter scaling the relative contribution of the norm penalty term, Ω, to the objective function. The value of α depicts the factor of regularisation (Goodfellow, Bengio, and Courville, 2016).

Decoding of Conditional Random Fields

In the final part of the pipeline (Step 6), the trained model from the previous component (Step 5) is used to find the most probable label sequence Y for a given input data sequence X. The data sequence X is a sentence from the test fold with corresponding feature vectors from Step 2. The probability is defined by a composite of the stated transition and emission probabilities. This task is conceived as a search problem. To infer the most likely label sequence, the dynamic programming algorithm, Viterbi, was implemented. The Viterbi algorithm becomes advantageous in this setting because it does not need to compute the probability for every possible label sequence Y, but merely infers the most likely sequence rather than computing all (Sutton and McCallum, 2010).

The process of decoding the CRF model and hereby elicit the most probable assignment $y* = argmax_y p(y|x)$ (Sutton and McCallum, 2010) is defined as:

$$\delta_t(j) = \max_{i \in S} \Psi_t(j, i, x_t)\delta_{t-1}(i) \tag{3.21}$$

where $\Psi_t(j, i, x_t)$ are the transition weights, and $\delta_{t-1}(i)$ is the marginal probability on each state.

The Viterbi algorithm is then used to recursively compute the most probable assignment by computing the probability of state transitions forward. When arriving at the end of the sequence, the

algorithm backtracks the states it has visited throughout the computation and eventually proposes the set of states, label sequence Y, that are most probable to classify data sequence X (Sutton and McCallum, 2010).

Evaluation of entity recognition

In the final process (Step 6), we evaluate the ability of the CRF-model to recognise entities in the medical texts. We approached this matter in two ways.

First, we wish to explore the relationship between the size of the training set and performance. This evaluation is done by training the CRF-model on 20 %, 40 %, 60 %, 80 %, and 100 % of the available training tuples.

Secondly, we wish to evaluate the overall performance of the CRF-model. To provide the least optimistic view of the ability to recognise entities, we applied the practice of K-fold cross-validation. We applied the manual technique of grid-search to determine the value of the hyper-parameter, K (Bergstra and Bengio, 2012). We evaluate the performance with $K = 5$, and $K = 10$, where the hyper-parameter of 5 was shown to provide the least optimistic view on the model performance. For that reason, we chose this value for performing the cross-validation of the system. In Table 3.7, we present average size measures of the folds used in the cross-validation.

TABLE 3.7: Average contents of the 5-fold training sets

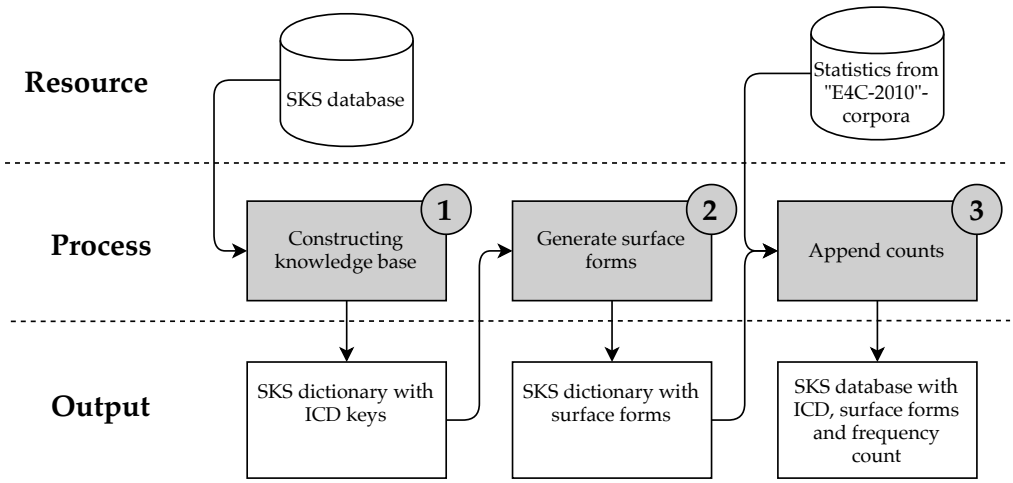
Number of documents	8,002
Number of tokens	73,982
Number of features	366,697
Number of weights	716,067

3.5 Disambiguation of medical words

We have now obtained the collection of documents with recognised disease mentions from the NER process. Next, we wish to disambiguate these mentions by linking them to an ontology (Sowa, 1995). In the following, we outline our approach to creating a dictionary for this system and our strategy for using this to disambiguate the recognised disease mentions.

3.5.1 Construction of knowledge base

FIGURE 3.6: Visualisation of dictionary construction



The first process (Step 1) in constructing the knowledge base, we are combining the English ICD-10 database with the Danish SKS-database. This merging operation is performed because the EHRs in the "E4C-2010"-corpora are labelled by the ICD-10 code, but contains medical text in Danish. Therefore, to enable the linking between the "E4C-2010"-corpora and the ICD-10 database, there is a need for Danish ICD-10 description. Thus, the idea is to create a knowledge base that contains the Danish disease description with the relating international ICD-10 code.

The input to this process is the SKS-database. This collection of data contains definitions of administrative matters, treatments, diseases, typical accidents, functional capacities and states of health, operations, medicinal products, anaesthesia, medical examinations, Clinical Physiology and Nuclear Medicine, and some additional procedures (Sunhedsdata-Styrelsen, 2019; Nielsen, 2017). All categories in the database are defined by a prefix. Disease categories are prefixed with a "D" for diagnose. The following part of the code corresponds to the ICD-10 equivalent identifier. Therefore, the relation between the international ICD-code classification and the equivalent SKS disease identifier can be expressed as:

$$ID_{SKS} = \text{'d'} + ID_{ICD} \tag{3.22}$$

Then, we modified the SKS-database to only contain the Danish disease descriptions and the related ICD-10 code. Since the SKS-database adopts the hierarchical ordering of codes from the ICD categorisation, our dictionary is also ordered in a hierarchy. In this way, the dictionary reflects the relationship between disease concepts (Khan and Safyan, 2014; Jiménez-Ruiz and Grau, 2011).
In the next process (Step 2), we generate simple surface forms on the dictionary. This is done to accommodate situations where a text span does not match any disease description in the dictionary. To encounter this problem, the surface forms help to provide textual variations of the dictionary to improve the chance of retrieving a result from the dictionary (Zhang et al., 2011).

The most obvious cases are when a disease mention is presented in upper-case, but the corresponding text span in the dictionary is presented in lower-case. Therefore, the dictionary is in this process extended with descriptions in both lower- and uppercase versions to heighten the possibility of a match between a mention and an entity in the dictionary.

In the final process (Step 3), we use the modified dictionary from the two previous steps. This step performs a frequency count on each individual ICD-10 code that is encountered in the "E4C-2010"-corpora. These frequency counts are appended to the corresponding ICD-10 code in the modified dictionary. This means that the modified dictionary may contain diseases that are not encountered in the corpora, leading to a frequency count of 0 in the dictionary. The construction of the knowledge base leads to the composition of the dictionary presented in Table 3.8.

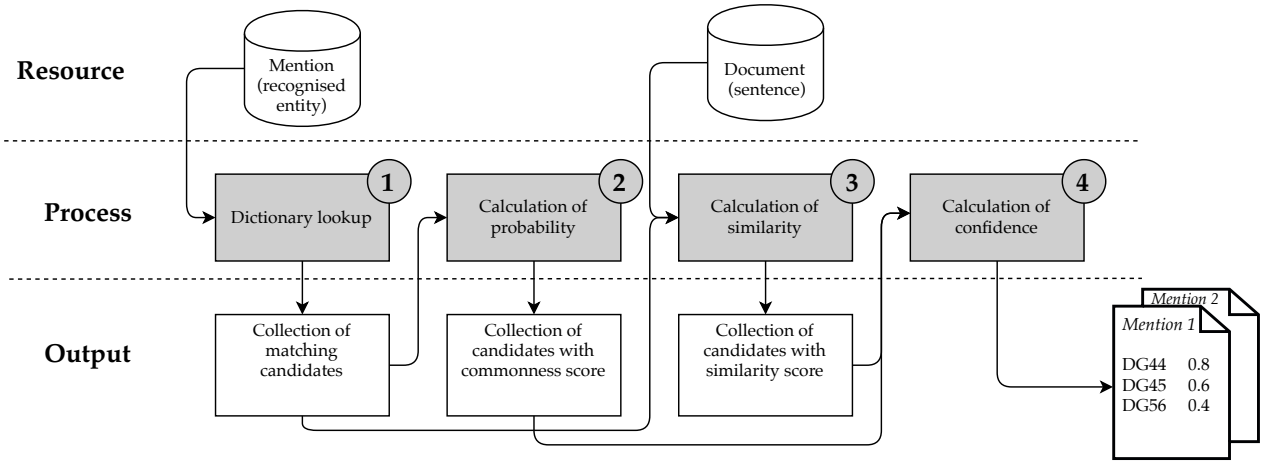
TABLE 3.8: Composition of the constructed knowledge base

Key	**Value**
ICD-10 code	• SKS-description • List of surface forms • Frequency count

3.5.2 Candidate selection and ranking

Having created the knowledge base, we now consider the selection of disease entities from the dictionary. This process is illustrated in Figure 3.7.

FIGURE 3.7: Visualisation of candidate selection and ranking



The first process (Step 1) is to perform a lookup with a query in the dictionary. The input query is the text-span of the disease mention. This lookup is performed for all recognised diseases by string matching. In doing so, we compare the string resemblance between the disease and descriptions and surface forms in the dictionary. If a match is found, the corresponding ICD-10 code is added to an intermediary collection of candidates. The final output is the list of all matching disease links in the dictionary for each disease mention.

In the next process (Step 2), we handle the list of candidates for each disease mention from the previous process. The idea is to calculate the relative probability of the candidate disease links in question. This is done by evaluating the proportion of appearances for the given candidate against the total number of appearances amongst all candidate links. This relationship is expressed below where the maximum-likelihood probability for an entity (e) is the right link of a mention (m). (Medelyan, Witten, and Milne, 2008).

$$Commonness_{e,m} = \frac{n(m,e)}{\sum_{e' \in \varepsilon} n(m,e')}, \tag{3.23}$$

Now, the list of all candidate disease links are appended with their respective *Commonness-score*.

In the third process (Step 3), the purpose is to calculate the contextual similarity, $sim_F(m,e)$, between the entity description term vector, d_e, and the document term vector representation of the mention, d_m (C. Bunescu and Pasca, 2006). This relationship is elicited by applying the similarity function, F, of Cosine similarity:

$$Sim_F(e,m) = \frac{d_m \cdot d_e}{\|d_m\| \cdot \|d_e\|} \tag{3.24}$$

$$cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3.25}$$

where d_m is the mention's term vector representation and d_e is the entity's term vector representation. The list of candidate links is then appended with the respective similarity score and brought forward to the next process.

The following process (Step 4) is to combine the *commonness* - and *similarity* score in a *confidence* measure, which expresses the product of the two previous scores (Medelyan, Witten, and Milne, 2008). When having the *confidence* score, it is possible to profoundly rank the list of candidate links according to this score:

$$Confidence(e, m) = Sim_{e,m} \cdot Commonness_{e,m} \tag{3.26}$$

The output of this process is thus the sorted set of candidates for each mention ordered by their individual confidence score.

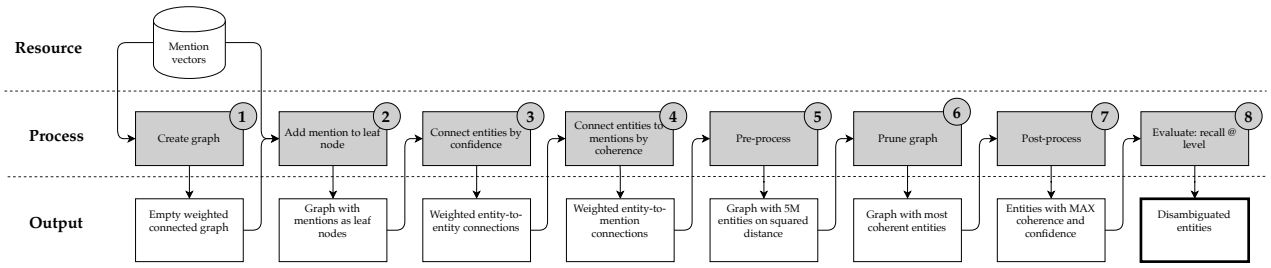
3.5.3 Entity inter-relatedness

During the process of disambiguation, we realised that not all documents take the same form. Generally, a document will only contain one disease mention. In this case, the disease link with the highest confidence is chosen as the right entity link.

However, in some situations, a document contains several mentions, e.g. *"Der tages endvidere titre mhp. atypisk lungebetændelse og det viser sig, at der er en positiv titer overfor mycoplasma pneumoniae"*. This sentence contains two entities *'atypisk lungebetændelse'* and *'mycoplasma pneumoniae'*. Therefore, we consider the disambiguation of disease mentions as two-fold. The problem for selecting the best entities for a single mention in a sentence (steps 2 to 4 in Candidate selection and ranking) constitutes a ranking problem within individual disambiguation (Balog, 2018). On the other side, collective disambiguation of multiple mentions within one sentence constitutes an inference problem (Balog, 2018). In the following, we present our approach to disambiguate multiple disease mentions within a single sentence.

Based on our observations from the data set, there presumably exists an inter-dependency between each mention in a document. That is, the majority of sentences focus on one medical phenomenon. This relatedness is conceived as advantageous when trying to disambiguate multiple disease mentions from the same document. This because every disambiguation of a disease mention is assumed to act as a clue for the disambiguation of other disease mentions in the same document (Balog, 2018). Based on this assumption, the idea is to maximise the confidence between each disease mention and each disease link while maximising the inter-relatedness between candidate disease links. We approached the collective disambiguation with the theory of graphs. In the following, we outline the process for resolving the issue of multiple disease mentions in one sentence. The processes are illustrated in Figure 3.8.

FIGURE 3.8: Pipeline for collective disambiguation



The first process (Step 1) is to instantiate an empty weighted undirected graph. In the next process (Step 2), all disease mentions, m, are added to the graph as leaf-nodes. Then (Step 3), all entity links for a given disease mention are added to the graph by creating an edge between the entity link to the relevant disease mention and all other entity links in the graph. Then, the weights on the edges are modified in two ways. Edges between a disease mention and an entity link are modified to represent the local compatibility, which is the confidence score obtained from the ranking process in the candidate selection. The following process (Step 4) is connecting all entities in the graph with all other entity links. In doing so, this process also adds the weights on edges connecting entity links to entities expressing the distance between nodes. The weight is expressed by the coherence score, highlighting the relatedness of the two topics. Our technique for finding the coherence score is presented below. It is inspired by Strube and Ponzetto, 2006.

```
coherence(icd1, icd2):
   score = 0
   for i in range(0, min(len(icd1), len(icd2))):
      If icd[i] == icd2[i]:
         score += (i + 1)
   return score / sum([i for i in range(1, max(len(icd1), len(icd2))+1)])
```

In the remaining steps, the approach proposed by Han, Sun, and Zhao, 2011 is adapted for processing of the constructed graph. In Step 5, the graph is pre-processed to lower the number of entities in the graph to highlight the most prominent and important entities, which supports the notion of inter-relatedness. Hence, the objective is to keep entities with the highest weights. Then, for all entities, the shortest path from the entity to every mention through intermediary entities is summed up. In this process, we use the inverse weights of the edges in the graph. By this, the graph is reduced to contain $kx|M_d|$ closest entities, where $|M_d|$ is the number of mentions in the graph, and k is set to 5 in this project.

The following process (Step 6) is then using the reduced graph for further pruning. The idea is to find the most optimal combination of entity links for all disease mentions in the document by pruning the graph using the minimum weighted degree, $mwd(G)$. This is obtained by calculating the weighted degree, $wd(i)$, which is the sum of the weights on all incident edges for all entities that are not the last for a mention in the graph:

$$wd(i) = \sum_{j \in \prod(i)} w(i,j) \tag{3.27}$$

where $j \in \prod(i)$ is the neighbourhood (all connected nodes) to node i and $w(i,j)$ is weight of the edge between node i and j.

Then, the minimum weighted degree (also referred to as density of the graph), $mwd(G)$, is defined as (Hoffart et al., 2011):

$$mwd(G) = \frac{min_{e \in \varepsilon_c} wd(e)}{|\varepsilon_c|} \tag{3.28}$$

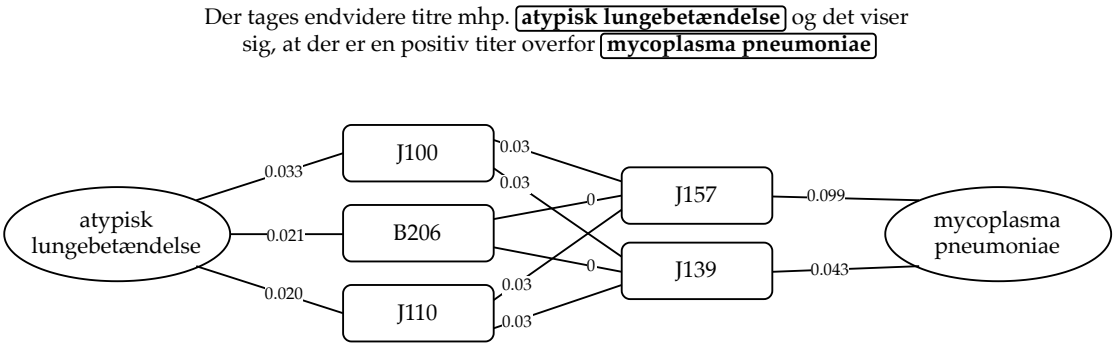
where ε_c is the set of entity nodes in the graph, $wd(n)$ is the weighted degree of a node.

The ambition is then to choose the entity that is not the last entity for a mention and has the lowest weighted degree, $wd(i)$. This entity link and all incident edges are removed from the graph. Then,

the minimum weighted degree of the graph is re-calculated. This process will continue until the minimum weighted degree decreases, expressing that the optimal composition of mention-entity pairs can be found in the graph (Balog, 2018).

In the final processing component (Step 7), we enumerate all possible paths from disease mention to disease mention in the graph. In this process, we find the best connection and thereby optimise the coherence between disease links and confidence between disease mentions. This allows ranking where the lowest sum of weights highlights the best combination of entities to create a path from disease mention to disease mention. That is, the set of entity links with the highest inter-relatedness is proposed as the best disambiguation for the disease mentions in the document. In this process, the order of removal is stored for later evaluation at the recall level. In this way, a queue maintains the order of which entities were removed from the graph and thereby allows ranking of the next best candidates in the next step. An example of a pruned graph is shown in Figure 3.9.

FIGURE 3.9: Example of constructed multiple entity graph



Evaluation of Entity Disambiguation

The last process (Step 8) is to evaluate the proposed ranking of entity combinations. This step imposes a practical implication seeing that medical personnel is only labelling an EHR with one ICD-code while a sentence sometimes contains more than one disease mention relating to distinct ICD-codes. That is a problem because we are not able to, with certainty, find the best ICD-code based on the selection of proposed disambiguated ICD-codes. It is assumed that there exists a bias towards certain codes why these are chosen as the main labelling of the EHR before other codes. To encounter this problem, we implemented the method of recall at different levels to tell whether the system actually captures the right labels, but does not suggest them as the best matching label. Another observation was that the designed system prefers specific labels whereas the actual labelling showed that more generic codes are used in practice. Therefore, we also evaluate the NED-component at different hierarchical levels of categories for the codes. We do this to compare whether the system actually captures, and to what extent, the medical area of a given disease mention.

Chapter 4

Results

In this section, we present the individual performance of the main sub-components Sentence Boundary Disambiguation, Named Entity Recognition, and Named Entity Disambiguation.

4.1 Sentence Boundary Disambiguation

To evaluate the performance of the SBD-component, the output of sentences produced is used as the main evaluation criterion. These results can be found in Table 4.3 and 4.4. Nonetheless, the final performance of the SBD is dependent on the list of abbreviations. The ability to detect abbreviations is shown in Table 4.2. Examples of correct and wrong abbreviations found in the corpora can be seen in Table 4.1.

TABLE 4.1: Examples of 10 correct and 10 wrong abbreviations found by the model

Correct abbreviations	**Wrong abbreviations**
dagpsyk.afd	*bagbenet*
aftenbehdl	*kloroform*
mercilonbeh	*gastroskopi*
inejc	*probat*
børnehosp	*mørkerædsel*
tyndtarmspåvirkn	*pskyisk*
ballonudv	*hyperæniske*
i.a.pt	*polypbiopsi*
vejtrækn.probl	*stomierne*
hudblødn	*hostemikstur*

As evident by Table 4.2, the unsupervised model used for abbreviation detection succeeds to correctly find abbreviations in almost all cases. Only 51 abbreviations were manually removed from the list during the manual validation process. This mounts up to 99.43% correct abbreviations and 0.56% wrong abbreviations.

TABLE 4.2: Results on detection of abbreviations

Abbreviations found by the model	9,036
Correct abbreviations	8,985
Wrong abbreviation	51
Added abbreviation	989
Total no. of final abbreviations	9,974

TABLE 4.3: Confusion matrix summarising the performance of SBD process

		Actual	
		Boundary	Non-boundary
Predicted	Boundary	166	14
	Non-boundary	34	4,550

TABLE 4.4: Precision (Pr), Recall (Re), and F1 score of the default distributed Danish model from NLTK and the domain-specific model

	"E4C-2010"-Corpora		
Model	Pr	Re	F1
Domain-specific model	92.22 %	83.00 %	87.37 %
PunktSentenceTokenizer (Danish)	89.22 %	82.10 %	85.51 %

From Table 4.4, we observe that the domain-specific model outperforms the default PunktSentence-Tokenizer on all measures.

4.2 Named Entity Recognition

The evaluation of the NER-component is here presented in two sections. In the first part, we show the impact on performance provided by the integrated features extractions. These results can be found in Tables 4.5, 4.8, 4.6, 4.7, and 4.10. In the second part, we present the impact of the training set size. This can be found in Table 4.2.2. Hereafter, we show the final average performance of the NER-component. These results can be found in Figure 4.2.3, and Table 4.12.

4.2.1 Impact of integrating feature extractions

To extract the relative impact of different features, the CRF-model has been trained using different combinations of feature extractions. The baseline measure considers no additional features, but relies solely on tokens and the respective BIO-labels from the training set. The result of additional features for the CRF is presented in Table 4.5.

TABLE 4.5: Impact of utilising additional features in the CRF

Features	Pr	Re	F1
Baseline	72.14 %	35.40 %	47.49 %
+ Similarity class (DistSim)	74.46 %	52.80 %	61.79 %
+ Word Shape (SHAPE)	74.03 %	40.30 %	52.19 %
+ N-grams (NG)	72.57 %	55.02 %	62.59 %
+ Gazetteer match (GAZ)	71.48 %	43.34 %	53.96 %
+ Namelist match (NAME)	72.96 %	36.57 %	48.72 %
+ + DistSim, SHAPE	71.64 %	51.64 %	60.01 %
+ + + DistSim, SHAPE, NG	73.48 %	60.86 %	66.58 %
+ + + + DistSim, SHAPE, NG, GAZ, NAME	75.74 %	62.73 %	68.63 %

In the following, we present the highest weighted features in the CRF-model trained with all available features extractions. In Table 4.6, we present weight features for the label "B", Table 4.7 for the label "I", and Table 4.8 for the label "O".

TABLE 4.6: Top ten weights for B-label features

Description	Feature	Type	Weight
Ingen-DISJP	W (full N-gram)	Current class	0.952
000000000001100-PSEQpDS	DistSim	Current and previous class	0.918
let-DISJN	W (full N-gram)	Current class	0.714
000000000001100-DISTSIM	DistSim	Current class	0.703
mistanke-DISJP	W (full n-gram)	Current class	0.691
#ød#	Mid N-gram	Current class	0.675
B-GAZ	GAZ match	Current class	0.653
00010110000010-PSEQpDS	DistSim	Current and previous class	0.634
#a>#	Suffix N-gram	Current class	0.594
#tu#	Mid N-gram	Current class	0.560

From Table 4.6 it is evident that the word cluster with the ID "000000000001100" has great importance for B-labels. Examples of words in this cluster are "dement", "hæs", "deprimeret", "valgusløst", "dyspnisk", and "forkølet". Moreover, it is interesting to note that Suffix-, Mid, and word N-grams seem important in detecting B-labels.

TABLE 4.7: Top ten weights for I-label features

Description	Feature	Type	Weight
000100010110-PSEQpDS	DistSim	Current class	0.678
000100010110-DISTSIM	DistSim	Current class	0.556
000101010110-PSEQpDS	DistSim	Current class and previous class	0.499
Contusio-DISJP	W (full n-gram)	Current class	0.485
<l	Prefix N-gram	Current class	0.483
Lipiderne-DISJP	W (full n-gram)	Current class	0.482
ALL-DIGITS-TNS1	Word Shape	Current class and previous class	0.480
I-GAZ2	GAZ match	Current class	0.449
slået-DISJP	W (full n-gram)	Current class	0.440
er-let-PSEQW2	W (full n-gram)	Current class and previous class	0.438

From Table 4.7 it is observed that the word cluster with the ID "000100010110" has great importance for I-labels. Examples of words in this cluster are "leddet", "hjertet", "hænderne", "lysken", "lungerne", "pungen", and "øjnene". Compared to the B-label, the I-label is mainly making use of word N-grams.

TABLE 4.8: Top ten weights for O-label features

Description	**Feature**	**Type**	**Weight**
ALL-DIGITS-PSEQpS	Word Shape	Current and previous class	1.194
<he	Prefix N-gram	Current class	1.14
null-NDISTSIM	DistSim	Current class	1.093
OBJEKTIVT-DISJP	W (full n-gram)	Current class	0.719
OTHER-TNS1	Word Shape	Current and previous class	0.716
I-GAZ	GAZ match	Current class	0.680
#<til#	Prefix N-gram	Current class	0.625
FEMALE_NAME	Namelist match	Current class	0.610
derhjemme-NW	W (full n-gram)	Current class	0.602
#podn#	Mid N-gram	Current class	0.587

From Table 4.8, we observed that the highest weights are not dominated by word clusters. Moreover, it is seen that the weights related to O-labels are higher than those included in tagging B- and I-labels. This could indicate that the model is rather certain about the relationship between some features and the correct label. An example of this is when the model experiences that the sequence only contains digits where this word shape is highly related with the O-label. This is also to be seen in comparison with a lower weight for the same feature used to recognise I-labels.

In the following, we highlight the extensive grid-search applied for hyper-parameter optimisation used to find the best composition of word clusters (Bergstra and Bengio, 2012). Table 4.9 outlines the configuration of the clusters, and the pre-determined active set size, and factor of minimum occurrence of words.

TABLE 4.9: Precision (Pr), Recall (Re), and F1 score of CRF in comparison of word clusters and cluster sizes. The highest scores for measures within each cluster type are bold-faced.

|C|	**Brown** C = 5,000, M = 3			**Brown** C = 5,000, M = 10			**Brown** C = 2,500, M = 3			**GloVe** Centroid clustering, M = 0		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
200	73.84 %	59.70 %	66.02 %	73.84 %	59.70 %	66.02 %	74.11 %	60.86 %	66.84 %	73.23 %	56.89 %	64.04 %
300	75.77 %	60.63 %	67.36 %	**75.77** %	60.63 %	67.36 %	73.99 %	60.16 %	66.37 %	74.36 %	57.94 %	65.13 %
400	75.11 %	59.58 %	66.45 %	75.11 %	59.58 %	66.45 %	73.57 %	60.16 %	66.20 %	73.23 %	56.89 %	64.04 %
500	74.33 %	61.21 %	67.14 %	74.53 %	60.16 %	66.58 %	73.97 %	60.75 %	66.71 %	73.49 %	57.01 %	64.21 %
600	74.75 %	60.86 %	67.10 %	74.89 %	60.98 %	67.22 %	74.57 %	60.63 %	66.88 %	74.96 %	58.06 %	**65.44** %
700	75.04 %	61.80 %	67.78 %	75.54 %	61.33 %	67.70 %	74.15 %	**60.98** %	66.92 %	74.89 %	57.83 %	65.26 %
800	75.39 %	62.27 %	68.20 %	75.14 %	61.10 %	67.40 %	74.28 %	60.05 %	66.40 %	73.80 %	57.24 %	64.47 %
900	**76.14** %	**62.62** %	**68.72** %	75.57 %	**61.45** %	**67.78** %	74.46 %	59.93 %	66.41 %	74.77 %	58.18 %	65.44 %
1,000	74.72 %	61.45 %	67.44 %	73.81 %	59.93 %	66.15 %	73.71 %	59.93 %	66.11 %	**75.23** %	57.48 %	65.17 %
1,100	74.08 %	61.10 %	66.97 %	74.14 %	60.28 %	66.49 %	74.06 %	60.05 %	66.32 %	73.60 %	57.01 %	64.25 %
1,200	73.78 %	60.16 %	66.28 %	75.00 %	60.28 %	66.84 %	74.86 %	60.51 %	66.93 %	74.77 %	58.18 %	65.44 %
1,300	74.00 %	60.51 %	66.58 %	74.53 %	59.81 %	66.36 %	74.35 %	60.28 %	66.58 %	73.80 %	57.24 %	64.47 %
1,400	74.36 %	60.98 %	67.01 %	74.49 %	59.70 %	66.28 %	74.71 %	60.05 %	66.58 %	74.25 %	57.94 %	65.09 %
1,500	74.00 %	60.51 %	66.58 %	74.38 %	59.70 %	66.23 %	74.82 %	60.40 %	66.84 %	72.55 %	57.13 %	63.92 %
1,600	74.82 %	59.35 %	66.19 %	74.82 %	59.35 %	66.19 %	74.38 %	59.70 %	66.23 %	74.13 %	57.24 %	64.60 %
1,700	74.45 %	59.58 %	66.19 %	74.45 %	59.58 %	66.19 %	74.27 %	59.35 %	65.97 %	73.91 %	57.59 %	64.74 %
1,800	74.60 %	59.35 %	66.10 %	74.60 %	59.35 %	66.10 %	74.82 %	60.40 %	66.84 %	73.67 %	**58.18** %	65.01 %
1,900	73.76 %	59.11 %	65.63 %	73.76 %	59.11 %	65.63 %	74.71 %	60.05 %	66.58 %	72.85 %	57.36 %	64.18 %
2,000	73.55 %	59.46 %	65.76 %	73.55 %	59.46 %	65.76 %	74.93 %	59.70 %	66.45 %	72.60 %	56.66 %	63.65 %
2,500	74.23 %	59.23 %	65.89 %	74.05 %	59.00 %	65.67 %	75.63 %	59.81 %	66.80 %	73.19 %	57.71 %	64.53 %
5,000	75.59 %	59.70 %	66.71 %	75.37 %	59.35 %	66.41 %	**76.26** %	60.05 %	**67.19** %	74.62 %	57.71 %	65.09 %

As evident from Table 4.9, the best tuning was shown to be with the Brown clustering approach with an active set size of 5,000, a minimum occurrence of 3, and 900 output clusters. This leads to an F1-measure of 68.72 % with a precision score of 76.14 %, and recall score of 62.62 %. We also

observe that the level of F-measure across all output measures are strongly affected by the relatively lower recall score compared to the precision score. Moreover, the best recall scores are experienced with lower cluster sizes. However, it is interesting to observe that the generation of Brown clusters with an active set size of 2,500 and minimum occurrence of 3 achieved the highest precision score with a cluster size of 5,000 whereas the best recall score was achieved with a cluster size of 700. This observation is found to be converse in the case of GloVe clusters where the precision score is greater when the cluster size is smaller, and the recall score is higher when the cluster size is large. A final note is that Brown clusters with an active set size of 5,000 are observed to outperform the Brown clusters with an active set size of 2,500. This is interesting because a higher set size also indicates a more noisy offset for the creation of the clusters. In Table 4.10, we put the relative contribution of the word clusters into perspective by showing the performance of the CRF-model without any word cluster features.

TABLE 4.10: Precision (Pr), Recall (Re), and F1-score of CRF without utilising any distributed similarity class

Pr	Re	F1
74.59	57.94	65.22

4.2.2 Impact of different number of training tuples

To assess the importance of training data, we have evaluated the performance of the CRF-model by using different sizes of the training set. This was done by dividing the total training set into partitions of 20%, 40%, 60%, 80%, and finally, 100%. The development of the precision-, recall, and F1-score is shown in Figure 4.2.2, and the training set sizes are shown in Table 4.11.

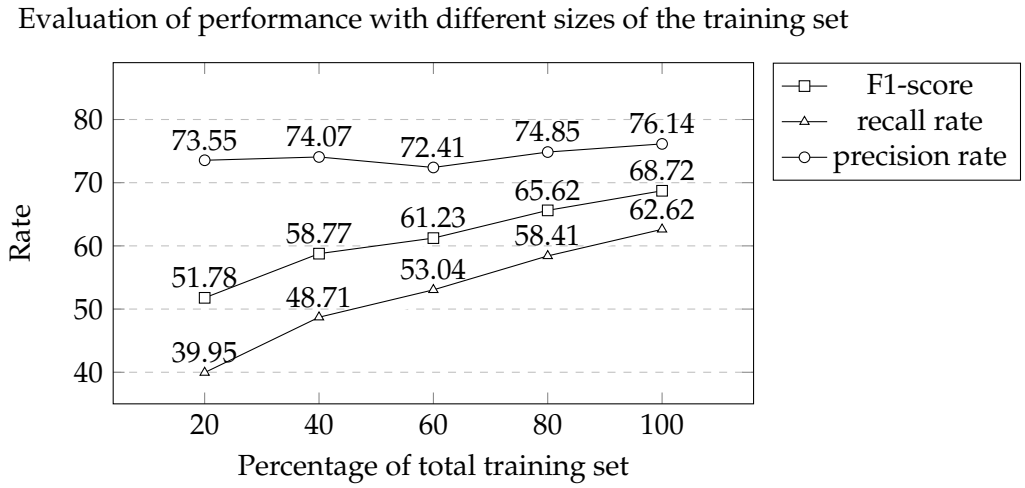


TABLE 4.11: Statistics on different sizes of the training set

Size	**Sentences**	**Word tokens**
20 %	1,668	14,763
40 %	3,310	29,544
60 %	4,920	44,367
80 %	6,487	59,214
100 %	8,002	74,138

From Figure 4.2.2, we observe that the performance in terms of the F1-score is increasing accordingly with a higher number of training tuples. This can be explained by looking at the recall-score that is significantly improved compared to the precision-score. Moreover, we note that the growth of the graph does not stabilise, which could indicate that the trend would continue if we obtained a larger training set.

4.2.3 Less optimistic view on CRF-model performance

To provide the least optimistic viewpoint on the performance of the CRF-model, the component was evaluated using K-fold cross validation, where $K = 5$. The different performance output achieved during the five iterations is shown in Figure 4.2.3, and the final average scores in Table 4.12.

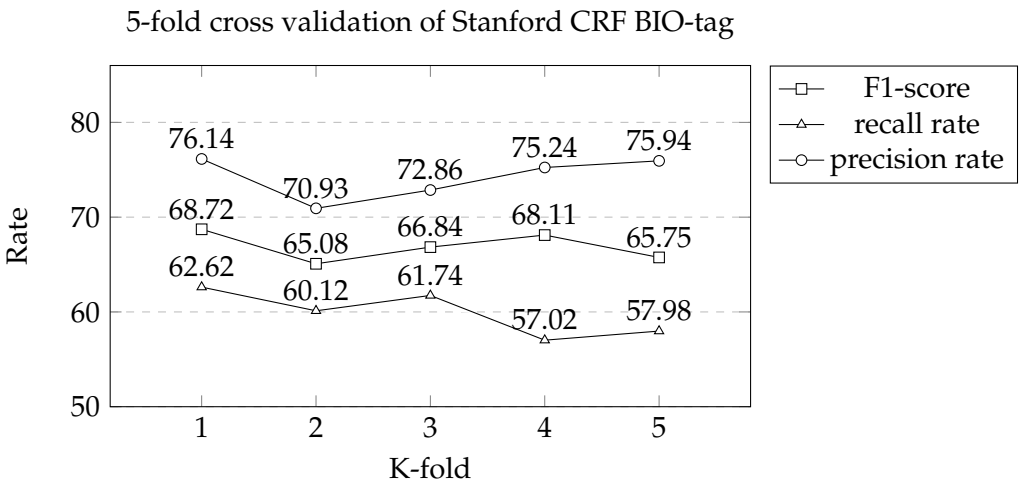


TABLE 4.12: Average scores from K-fold cross validation

Pr	Re	F_1
72.20	60.15	68.55

4.3 Named Entity Disambiguation

The evaluation of the NED-component is presented in two parts. In the first part, we present the performance of disambiguating individual entities from sentences with one entity. These results are found in Table 4.15. In the subsequent part, we present the performance of collective disambiguation. The results are outlined in Table 4.14. Both tables show the evaluation of the performance by the technique of recall@level. This is used to show the ability of the NED-component to find next-best candidates if the highest ranking entity was not the actual ICD-10 label.

4.3.1 Performance of disambiguating a single entity

The test set used for evaluating the ability to disambiguate sentences with one disease mention only contains sentences with single mentions. The measure of performances at different recall levels are presented in Table 4.15. Hereafter, we present a practical example of the systems ability to retrieve the right entities, but issue in ranking the correct as the first. We observe that the performance of the system is uplifted by 44.67 % when including all entity links until the recall level of 5.

TABLE 4.13: Evaluation of NED on 1,000 single mention sentences

Level	Re	F1
Recall@1	14.51 %	25.34 %
Recall@2	16.92 %	28.94 %
Recall@3	19.97 %	33.29 %
Recall@4	21.35 %	35.19 %
Recall@5	22.45 %	36.66 %

4.3.2 Performance in disambiguation of multiple diseases

The test used for evaluating the ability to disambiguate sentences with multiple mentions only contains sentences with multiple mentions. The measure of performances at different recall levels are presented in Table 4.14. We observe that the performance of the system is uplifted by 70.30 % when including all entity links to the recall level of 5.

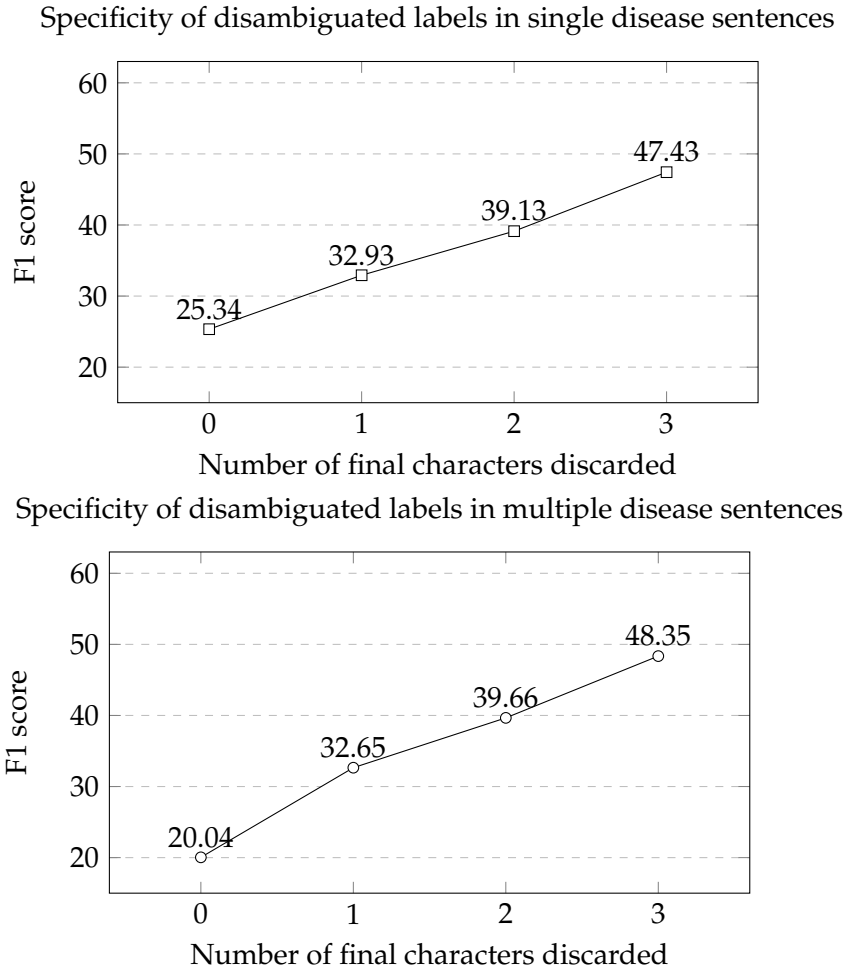
TABLE 4.14: Evaluation of NED on 1,000 multiple disease mention sentences

Level	Re	F1
Recall@1	11.14 %	20.04 %
Recall@2	16.32 %	28.06 %
Recall@3	17.85 %	30.29 %
Recall@4	19.33 %	32.39 %
Recall@5	20.58 %	34.13 %

A practical example of a Recall@Level match is the sentence *"Nuværende: I eller måske 1 år har pt. haft megen hovedpine."*. The NER-process elicited the disease mention *"hovedpine"* from this sentence. In the NED-process, the best matching entity link was found to be "G444" while the actual label is "G442". In consideration of the rank, we then found that the actual label does exist in the top five ranks (G444, G440, G443, G442, R51). From that, it is evident that the "G442" match results in a Recall@4.

4.3.3 Specificity of the ICD label

To evaluate the level of specificity in the disambiguated diseases, we manipulate the proposed codes to obtain a lower level of specificity. This process includes the removal of the final characters of each disease code. In this way, the disambiguated code is slightly generalised to represent the parent entity exploiting the hierarchy of ICD-codes. The evaluation is performed on multiple levels of suffix omissions. This involves evaluating the exact code (full length), 1 final character discarded, 2 final characters discarded, and 3 final characters discarded. In Table 4.3.3, we present the evaluation of sentences with single disease mentions using those parameters while we in Table 4.3.3 present the evaluation of sentences containing multiple disease mentions.



In Table 4.15, we present practical examples of obstacles encountered when concerning the notion of specificity.

TABLE 4.15: Examples of specificity encountered in disambiguation of disease mentions

Retrieved ICD	**Actual ICD**	**Original sentence**	**Disease mention**	**Comment**
M161	M16	"Svær hofteledsarthrose."	"hofteled-sarthrose"	The actual EHR was labelled with the main ICD-catagory ("Slidgigt" / "osteoarthritis"), while the disambiguation resulted in a more specific diagnose (Hofteledsartrose UNS)
G4741, None	G47	"Spec. har jeg ikke mistanke om narkolepsi eller narkolepsilign."	"narkolepsi", "narkolep-silign"	The first disease disambiguation relates to a more specific sub category than the correct label (G4741, G4744, G4742 in rank). Therefore, the main category and first-level subcategory does match. The second disease mention returned no results in disambiguation.
F329	F32	"05-04-01 : Indenfor den seneste tid tiltagende depressiv."	"depressiv"	(Depressiv enkeltepisode UNS) vs (Depressiv enkeltepisode), but the right entity label was in the next best entity ranks (F322, F32, F320, F321, F3211)

Chapter 5

Discussion and future research

This study is part of a Master project that was to be completed within the time-span of four months. This project is ambitious because, to the best of our knowledge, no existing research has applied NLP approaches on EHR in Danish. The time constraint and the knowledge gap impacted the choice of approaches that were applied. For that reason, we wish to critically evaluate the applied approaches and furthermore share our considerations for future research. In the following, we will discuss our approach in chronological sequence.

In our approach to SBD, we achieved a final F1-score of 87.37 % that is made up by a precision score of 92.22 %, and a recall score of 83.00 %. In comparison with the techniques applied to the GENIA corpus, we see that the average scores amount to an F1-score of 91.0 % with a precision score of 89.2 % and a recall score of 92.6 % (Newman-Griffis et al., 2016). From this, it is clear that our performance levels with the average performance on that corpus.

Moreover, when we regard the average performance of the approaches used on the i2b2 corpora, we observe that they achieved an average F1-score of 55.8 % with a precision score of 65.0 % and a recall score of 50.4 % (Newman-Griffis et al., 2016). Compared to these results, it is evident that our approach outperforms this average performance on this corpora.

Although our SBD approach is seen to level with state-of-the-art results, we still believe that the performance of our approach can be further improved. When we inspect the errors made by the SBD component, we find that most of these relate to sentences that do not include actual medical content. Instead, the content is seen as noisy data, which is not correctly handled by the model. This issue was not further considered because it has little impact on later components in our pipeline. Nonetheless, this issue affects the performance of the SBD component. Therefore, we suggest extending the use of regular expressions or handling the noisy data in a pre-processing phase.

In our approach to NER, we achieved a final F1-score of 68.55 % made up by a precision score of 72.20 % and a recall score of 60.15 %. When comparing these results with previous studies, it is evident that our approach does not level with the performance of previous projects on NER in other languages. The projects in the CoNLL-2003 shared task achieved an average F1-score of 82.17 % with a precision score of 82.67 % and a recall score of 81.83 % (Computational Language Learning, 2003). Moreover, in a medical setting, the average performance of the I2B2-challenge achieved an F1-score of 82.42 % with a precision score of 83.40 % and a recall score of 81.12 % (Foundation, 2019).

The variance in performance gives rise to the question of differences in approaching the task of NER in unstructured text. When inspecting the approaches in the CoNLL-2003 shared task, it is clear that many applied the modeling technique of HMM and MEMM. Moreover, almost all studies made use of the tagging scheme of POS (Computational Language Learning, 2003). The i2b2-challenge showed similar approaches to this study where CRF is the main modeling technique, and the BIO-format is used almost half of the times (Foundation, 2019). The use of the BIO-format could indicate that the combination of CRF and BIO-format is a good strategy for processing unstructured medical text. The explanation of the difference in performance is presumably found

elsewhere. This assumption leads to a comparison of obvious differences between current research and our approach. Here we see that the size of the training sets differs from ours.

The CoNLL-2003 corpus contains a training-set with 203,621 word tokens (Sang and De Meulder, 2003a), where the I2B2-2010 corpus contains 260,573 word tokens in the training-set (Gurulingappa, Hofmann-Apitius, and Fluck, 2010).

Our training set contains 92,485 word tokens, which is clearly substantially smaller. The differences are -54,57 % compared to CoNLL-2003 and -64,50 % compared to i2b2-2010. We perceive this as a possible explanation for the differences in performance. This notion is also supported by our experiment where we proved that our performance increases with a gradually larger training set.

Another interesting subject is to apply a neural tool in recognising entities from medical text. However, we see two main challenges in adopting this approach.

First, neural methods demand a substantial amount of data compared to non-neural approaches. Considering our challenges in creating enough training data for the encoding of our CRF approach, this demand seems unachievable in the scope of this project.

Second, we conceive neural approaches as a "black-box" decision model that does not allow disclosure of the underlying rationale for decisions made. However, in this study, we strived to achieve a transparent system where we can explain the basis for making decisions. Moreover, considering that the medical texts contain sensitive personal data, we want to allow for exposure of potential biases in the model.

In our approach to NED we succeed in creating a baseline functionality for the disambiguation of diseases. However, we found this to be a complex, domain-specific challenge requiring a thorough ontology and general medical knowledge. Therefore, we experienced two main challenges in our approach to NED.

First, in our approach, we realised that a great number of lookups resulted in no match or an incorrect match from the dictionary. We believe that this result can be partially explained by considering our construction of the ontology for this solution. We propose that future work should focus on extending the knowledge-base to achieve a more advanced foundation. This improvement could be achieved by extension of the surface forms with the incorporation of knowledge from medical databases on synonyms and words used in the same contexts.

Moreover, some of the resources that were created throughout this study could also be used. The list of abbreviations acquired in the SBD could be used to extend the dictionary with domain-specific synonyms, and abbreviations. Another approach could be to generate statistics on the "E4C-2010"-corpora where the frequency of words revolving around about specific diseases could be added to each entity in the dictionary. All in all, these approaches could help to further the rate of matching due to the greater amount of items related to each entity in the dictionary.

Second, we experienced practical implications in labelling EHR based on the disambiguated diseases. One practical implication is that we found that a sentence may contain distinct and even diverging disease mentions. This experience challenged our assumption that an EHR only concerns one disease.

Another implication is that the NED succeeds in finding a less specific category of a disease, but fails to allocate it to the specific category. Our system is designed to find exact matches, but the reality may be that medical personnel is more interested in finding the correct main category and gives less importance to the detailed categories.

Finally, it is evident that our system finds relevant matches when regarding different levels of recall. That is, the correct label is included in the list of candidate links, but not always suggested as the best match. The combination of all practical implications points to a gap between the system design and the use of EHR in the medical sector. This gap is not conceived as a matter of design flaws, but rather a question of acquiring knowledge about the professional practices in the sector. For that reason, we recommend that future projects engage in collaboration with medical personnel to obtain the required knowledge for designing an applicable practical system.

Chapter 6

Conclusion

Our ambition for this study was to propose a baseline approach for extracting and disambiguating diseases from EHRs in Danish. In doing so, we proposed a range of techniques to create a fully coherent NLP pipeline to process unstructured medical text in Danish.

Initially, we achieved the unlocking of a data set containing Danish EHR and made it accessible for this study and future research.

We have revealed the challenges of segmenting sentences and words in Danish medical texts. By the use of an unsupervised model, we succeeded in creating a tool that facilitates the detection of sentence boundaries in this sub-domain. Moreover, our performance levels with state-of-the-art practices with an F1-score of 87.37%. Our contributions in terms of SBD in Danish medical text is a functioning tool and a collection of 9,974 domain-specific abbreviations.

We applied a CRF-model where the joint probability of a given label sequence co-occurring with a data sequence was mapped to learn the structure of Danish medical language. We thereby prove the possible appliance of CRF in this sub-domain. To facilitate the training of the model, we manually annotated 92,485 word tokens using the BIO-scheme. Moreover, we performed extensive testing on the Brown and the GloVe clustering techniques. Also, we test and highlight the most important features for word representation within this sub-domain. Our result in recognising diseases in Danish medical text amounts to an F1-score of 68.55%.

Finally, we proposed a framework for NED in Danish medical texts. In the development of this framework, we experienced practical implications for using NED in the Danish medical sector. In relation to these, we map our considerations for succeeding with such implementation in the future.

In this study, we therefore succeed in creating a baseline approach for applying NLP tools on medical texts in Danish.

Appendix A

Applying the mappings

Table from Pantazos, Lauesen, and Lippert, 2017.

TABLE A.1

Identifying fields	
Civil registration number (CPR)	Replace it with the new CPR in the CPR mapping table
First name	Select the first male or first female mapping table according to the gender code in CPR. Replace first name with the new name in the mapping table
Last name	Replace it with a new name according to the mapping table
Address	An address contains a street name, a house number and sometimes a floor number and entrance position (e.g. Byvej 21, 2tv). Replace the street name according to the street mapping table. Replace numbers randomly with a number that has the same number of digits
Phone numbers	Alter each phone number to a random number with the same number of digits
E-mail	Alter the address with random characters before the letter @ and change the domain name to email.dk
Quasi-identifiers	
Zip code	Replace it according to the zip mapping table
City	Replace it with the city name in the zip mapping table
Country	Change it to Denmark
Date of birth	Set it from the new CPR
Date of death	Randomly change the day and month
Hospital name	Replace it with a new name according to the mapping table
Clinic name	Replace it with a new name according to the mapping table
Clinician first name	Replace it with a new name according to the mapping table for first names
Clinician last name	Replace it with a new name according to the mapping table for last names
Clinician alias	Replace it with the new first name of the clinician
Age	Remove all patients older than 90 years due to high anonymity risks

Appendix B

E4C-2010 database mapping

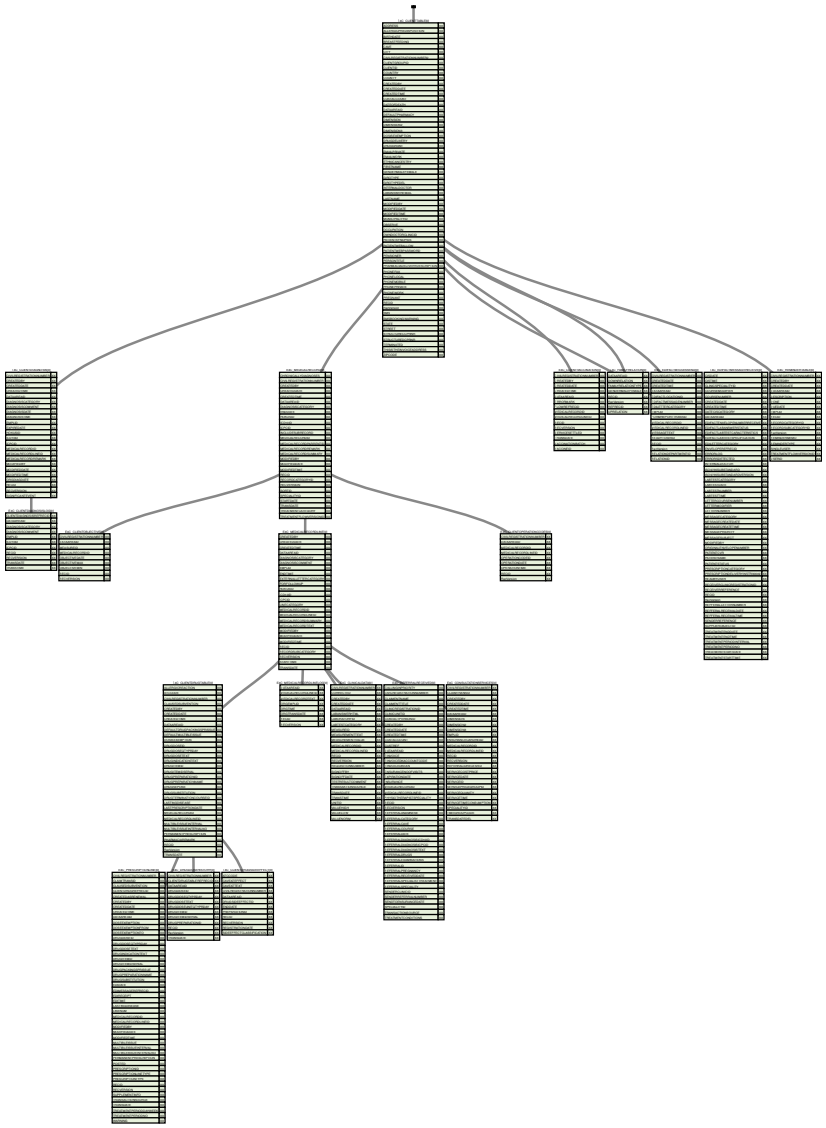


FIGURE B.1: We recommend digital reading for viewing this mapping

Bibliography

Afzal, Naveed et al. (2018). "Natural language processing of clinical notes for identification of critical limb ischemia". In: *International journal of medical informatics* 111, pp. 83–89.

Amazon (2018). Accessed 4 December 2018. URL: https://aws.amazon.com/blogs/machine-learning/introducing-medical-language-processing-with-amazon-comprehend-medical/.

Balog, Krisztian (2018). *Entity-Oriented Search*. Springer International PU. ISBN: 978-3-319-93933-9. DOI: 10.1007/978-3-319-93935-3.

Bender, Oliver, Franz Josef Och, and Hermann Ney (2003). "Maximum entropy models for named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 148–151.

Bergstra, James and Yoshua Bengio (2012). "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.Feb, pp. 281–305.

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc. ISBN: 0596516495, 9780596516499.

Broder, Andrei Z et al. (1997). "Syntactic clustering of the web". In: *Computer Networks and ISDN Systems* 29.8-13, pp. 1157–1166.

Brown, Peter F et al. (1992). "Class-based n-gram models of natural language". In: *Computational linguistics* 18.4, pp. 467–479.

Bruijn, Berry de et al. (2010). "NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features". In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*.

Byrd, Richard H et al. (1995). "A limited memory algorithm for bound constrained optimization". In: *SIAM Journal on Scientific Computing* 16.5, pp. 1190–1208.

C. Bunescu, Razvan and Marius Pasca (2006). "Using Encyclopedic Knowledge for Named entity Disambiguation." In:

Carreras, Xavier, Lluís Màrquez, and Lluís Padró (2003). "A simple named entity extractor using AdaBoost". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*.

Carreras, Xavier, Lluis Màrquez, and Lluis Padró (2003). "Learning a perceptron-based named entity chunker via online recognition feedback". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Cavnar, William B, John M Trenkle, et al. (1994). "N-gram-based text categorization". In: *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*. Vol. 161175. Citeseer.

Chang, Yu-shan and Yun-Hsuan Sung (2005). "Applying name entity recognition to informal text". In: *Recall* 1, p. 1.

Chieu, Hai Leong and Hwee Tou Ng (2003). "Named entity recognition with a maximum entropy approach". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 160–163.

Christen, Peter (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Christen, Peter and Karl Goiser (2007). "Quality and complexity measures for data linkage and deduplication". In: *Quality measures in data mining*. Springer, pp. 127–151.

Cimiano, Philipp (2006). *Ontology learning and population from text: algorithms, evaluation and applications*. Vol. 27. Springer Science & Business Media.

Computational Language Learning, Conference on (2003). *CoNLL 2003 - Language-Independent Named Entity Recognition (II)*. Accessed 15 April 2019. URL: https://www.clips.uantwerpen.be/conll2003/ner/.

— (2019a). *Previous Tasks*. Accessed 30 April 2019. URL: http://www.conll.org/previous-tasks.

— (2019b). *Shared task*. Accessed 30 April 2019. URL: http://mrp.nlpl.eu.

Cucerzan, Silviu (2007). "Large-Scale Named Entity Disambiguation Based on Wikipedia Data". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, pp. 708–716. URL: https://www.aclweb.org/anthology/D07-1074.

Cui, Liwen, Xiaolei Xie, and Zuojun Shen (2018). "Prediction task guided representation learning of medical codes in EHR". In: *Journal of biomedical informatics*.

Curran, James and Stephen Clark (2003). "Language independent NER using a maximum entropy tagger". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*.

Darroch, John N and Douglas Ratcliff (1972). "Generalized iterative scaling for log-linear models". In: *The annals of mathematical statistics*, pp. 1470–1480.

De Martino, Andrea and Daniele De Martino (2018). "An introduction to the maximum entropy approach and its application to inference problems in biology". In: *Heliyon* 4.4, e00596.

De Meulder, Fien and Walter Daelemans (2003). "Memory-based named entity recognition using unannotated data". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 208–211.

Derczynski, Leon and Sean Chester (2016). "Generalised Brown clustering and roll-up feature generation". In: *Thirtieth AAAI Conference on Artificial Intelligence*.

Dredze, Mark et al. (2010). "Entity Disambiguation for Knowledge Base Population". In: *Proceedings of the 23rd International Conference on Computational Linguistics*. COLING '10. Beijing, China: Association for Computational Linguistics, pp. 277–285. URL: http://dl.acm.org/citation.cfm?id=1873781.1873813.

Dymarski, Przemyslaw (2011). *Hidden Markov Models: Theory and Applications*. BoD–Books on Demand.

Fan, Jung-wei et al. (2011). "Part-of-speech tagging for clinical text: wall or bridge between institutions?" In: *AMIA Annual Symposium Proceedings*. Vol. 2011. American Medical Informatics Association, p. 382.

Fan, Yadan and Rui Zhang (2018). "Using natural language processing methods to classify use status of dietary supplements in clinical notes". In: *BMC medical informatics and decision making* 18.2, p. 51.

Florian, Radu et al. (2003). "Named entity recognition through classifier combination". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 168–171.

Foundation, i2b2tranSMART (2019). *Informatics for integrating Biology the Bedside*. Accessed 20 February 2019. URL: https://www.i2b2.org.

Getoor, Lise and Ashwin Machanavajjhala (2012). "Entity resolution: theory, practice & open challenges". In: *Proceedings of the VLDB Endowment* 5.12, pp. 2018–2019.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodman, Joshua (2002). "Sequential conditional generalized iterative scaling". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 9–16.

Grefenstette, Gregory and Pasi Tapanainen (1994). "What is a word, what is a sentence?: problems of Tokenisation". In:

Gurulingappa, H, M Hofmann-Apitius, and J Fluck (2010). "Concept identification and assertion classification in patient health records". In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. i2b2, Boston, MA, USA.

Hamid, H et al. (2013). "Validating a natural language processing tool to exclude psychogenic nonepileptic seizures in electronic medical record-based epilepsy research". In: *Epilepsy & Behavior* 29.3, pp. 578–580.

Hammerton, James (2003). "Named entity recognition with long short-term memory". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 172–175.

Han, Jiawei, Jian Pei, and Micheline Kamber (2011). *Data mining: concepts and techniques*. Elsevier.

Han, Xianpei, Le Sun, and Jun Zhao (2011). "Collective Entity Linking in Web Text: A Graph-based Method". In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '11. Beijing, China: ACM, pp. 765–774. ISBN: 978-1-4503-0757-4. DOI: 10.1145/2009916.2010019. URL: http://doi.acm.org/10.1145/2009916.2010019.

Hand, David and Peter Christen (2018). "A note on using the F-measure for evaluating record linkage algorithms". In: *Statistics and Computing* 28.3, pp. 539–547.

Hendrickx, Iris and Antal Van Den Bosch (2003). "Memory-based one-step named-entity recognition: Effects of seed list features, classifier stacking, and unannotated data". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 176–179.

Hoffart, Johannes et al. (2011). "Robust Disambiguation of Named Entities in Text". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 782–792. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145521.

Horsmann, Tobias and Torsten Zesch (2016). "LTL-UDE @ EmpiriST 2015: Tokenization and PoS Tagging of Social Media Text". In: *Proceedings of the 10th Web as Corpus Workshop*, pp. 120–126.

Jiménez-Ruiz, Ernesto and Bernardo Cuenca Grau (2011). "Logmap: Logic-based and scalable ontology matching". In: *International Semantic Web Conference*. Springer, pp. 273–288.

Jonnalagadda, Siddhartha et al. (2012). "Enhancing clinical concept extraction with distributional semantics". In: *Journal of biomedical informatics* 45.1, pp. 129–140.

Jurafsky, Dan and James H Martin (2014). *Speech and language processing*. Vol. 3. Pearson London.

Kang, Ning et al. (2010). "Erasmus MC approaches to the i2b2 Challenge". In: *Proceedings of the 2010 i2b2/VA workshop on challenges in natural language processing for clinical data*. i2b2, Boston, MA, USA.

Kazama, Jun'ichi and Jun'ichi Tsujii (2003). "Evaluation and extension of maximum entropy models with inequality constraints". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pp. 137–144.

Khan, Sharifullah and Muhammad Safyan (2014). "Semantic matching in hierarchical ontologies". In: *Journal of King Saud University - Computer and Information Sciences* 26.3, pp. 247 –257. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2014.03.010. URL: http://www.sciencedirect.com/science/article/pii/S1319157814000111.

Kiss, Tibor and Jan Strunk (2002). "Viewing sentence boundary detection as collocation identification". In: *In Proceedings of KONVENS 2002*, pp. 75–82.

— (2006). "Unsupervised Multilingual Sentence Boundary Detection". In: *Computational Linguistics* 32.4, pp. 485–525. DOI: 10.1162/coli.2006.32.4.485. eprint: https://doi.org/10.1162/coli.2006.32.4.485. URL: https://doi.org/10.1162/coli.2006.32.4.485.

Klein, Dan et al. (2003). "Named entity recognition with character-level models". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 180–183.

Kulkarni, Sayali et al. (2009). "Collective Annotation of Wikipedia Entities in Web Text". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

KDD '09. Paris, France: ACM, pp. 457–466. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557073. URL: http://doi.acm.org/10.1145/1557019.1557073.

Lafferty, John, Andrew McCallum, and Fernando CN Pereira (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In:

Lavergne, Thomas, Olivier Cappé, and François Yvon (2010). "Practical very large scale CRFs". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 504–513.

Liu, Yan et al. (2006). "Protein fold recognition using segmentation conditional random fields (SCRFs)". In: *Journal of Computational Biology* 13.2, pp. 394–406.

MacIntyre, Robert (1995). *Sed script to produce Penn Treebank tokenization*. URL: http://www.cis.upenn.edu/~treebank/tokenizer.sed.

Mandag-Morgen and Tryg-Fonden (2019). *Kodeks for god sundhedsformidling*. Accessed 21 May 2019. URL: https://www.mm.dk/misc/Kodeks_for_god_sundhedsformidling_TÃęnketankenMandagMorgen_TrygFonden.pdf.

Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze (2010). "Introduction to information retrieval". In: *Natural Language Engineering* 16.1, pp. 100–103.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press. ISBN: 0-262-13360-1.

Mayfield, James, Paul McNamee, and Christine Piatko (2003). "Named entity recognition using hundreds of thousands of features". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 184–187.

McCallum, Andrew and Wei Li (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*. Association for Computational Linguistics, pp. 188–191.

Medelyan, Olena, Ian H. Witten, and David N. Milne (2008). "Topic indexing with Wikipedia". In: AAAI Technical Report WS-08-15. Conference Contribution, pp. 19–24. URL: https://hdl.handle.net/10289/1776.

Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Moriyama, Iwao Milton et al. (2011). "History of the statistical classification of diseases and causes of death". In:

Munro, Robert, Daren Ler, and Jon Patrick (2003). "Meta-learning orthographic and contextual models for language independent named entity recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Newman-Griffis, Denis et al. (2016). "A Quantitative and Qualitative Evaluation of Sentence Boundary Detection for the Clinical Domain". In: *AMIA Joint Summits on Translational Science proceedings. AMIA Summit on Translational Science* 2016, pp. 88–97.

Nielsen, Kristian (2017). *Landspatientregisteret (LPR)*. Accessed 28 November 2018. URL: https://sundhedsdatastyrelsen.dk/da/registre-og-services/om-de-nationale-sundhedsregistre/sygedomme-laegemidler-og-behandlinger/landspatientregisteret.

Pantazos, Kostas, Soren Lauesen, and Soren Lippert (2017). "Preserving medical correctness, readability and consistency in de-identified health records". In: *Health informatics journal* 23.4, pp. 291–303.

Parliament, The Eurpoean and The Council of The European Union (2016). *General Data Protection Regulation*. Accessed 21 May 2019. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ\%3AL\%3A2016\%3A119\%3ATOC.

Patientregistrering (2018). *Sundhedsvæsenets Klassifikations System (SKS)*. Accessed 28 November 2018. URL: https://sundhedsdatastyrelsen.dk/da/rammer-og-retningslinjer/om-klassifikationer/sks-klassifikationer.

Patrick, Jon D et al. (2011). "A knowledge discovery and reuse pipeline for information extraction in clinical notes". In: *Journal of the American Medical Informatics Association* 18.5, pp. 574–579.

Peng, Fuchun and Andrew McCallum (2006). "Information extraction from research papers using conditional random fields". In: *Information processing & management* 42.4, pp. 963–979.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Phuong, Le-Hong, Xuan-Hieu Phan, and Tran The Trung (2013). "On the Effect of the Label Bias Problem in Part-of-Speech Tagging". In:

Ramshaw, Lance A. and Mitchell P. Marcus (1995). "Text Chunking using Transformation-Based Learning". In: *CoRR* cmp-lg/9505040. URL: http://arxiv.org/abs/cmp-lg/9505040.

Ratinov, Lev and Dan Roth (2009a). "Design challenges and misconceptions in named entity recognition". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL 09*. DOI: 10.3115/1596374.1596399. URL: http://www.aclweb.org/anthology/W09-1119.

— (2009b). "Design challenges and misconceptions in named entity recognition". In: *Proceedings of the thirteenth conference on computational natural language learning*. Association for Computational Linguistics, pp. 147–155.

Reiermann, Jens and Torben K.-Andersen (2019). *Guldgrube af sunhedsdata samler stoev*. Accessed 19 May 2019. URL: https://www.mm.dk/artikel/guldgrube-af-sundhedsdata-samler-stoev.

Reynar, Jeffrey C. and Adwait Ratnaparkhi (1997). "A maximum entropy approach to identifying sentence boundaries". In: *Proceedings of the fifth conference on Applied natural language processing -*. DOI: 10.3115/974557.974561.

Russell, Stuart J and Peter Norvig (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,

Sak, Haşim, Andrew Senior, and Françoise Beaufays (2014). "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: *Fifteenth annual conference of the international speech communication association*.

Sang, Erik F and Fien De Meulder (2003a). "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *arXiv preprint cs/0306050*.

— (2003b). "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition". In: *arXiv preprint cs/0306050*.

Sato, Kengo and Yasubumi Sakakibara (2005). "RNA secondary structural alignment with conditional random fields". In: *Bioinformatics* 21.suppl_2, pp. ii237–ii242.

Settles, Burr (2005). "ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text". In: *Bioinformatics* 21.14, pp. 3191–3192.

Sha, Fei and Fernando Pereira (2003). "Shallow parsing with conditional random fields". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 134–141.

Sowa, John F (1995). "Top-level ontological categories". In: *International journal of human-computer studies* 43.5-6, pp. 669–685.

Stanford-Natural-Language-Processing-Group (2019). *Standford Named Entity Recognizer*. Accessed 3 February 2019. URL: https://nlp.stanford.edu/software/CRF-NER.html.

Strube, Michael and Simone Paolo Ponzetto (2006). "WikiRelate! Computing Semantic Relatedness Using Wikipedia". In: *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*. AAAI'06. Boston, Massachusetts: AAAI Press, pp. 1419–1424. ISBN: 978-1-57735-281-5. URL: http://dl.acm.org/citation.cfm?id=1597348.1597414.

Sunhedsdata-Styrelsen (2019). *SKS-browseren*. Accessed 4 February 2019. URL: http://www.medinfo.dk/sks/brows.php.

Sutton, Charles, Andrew McCallum, et al. (2010). "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4, pp. 1–70.

— (2012). "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4, pp. 267–373.

Tarasov, DS (2015). "Natural language generation, paraphrasing and summarization of user reviews with recurrent neural networks". In: *Materials of international conference" Dialog*.

Tomanek, Katrin, Joachim Wermter, and Udo Hahn (2007). "Sentence and token splitting based on conditional random fields". In: *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*. Vol. 49, p. 57.

Trautner Kromann, Matthias (2003). "The Danish Dependency Treebank and the DTAG Treebank Tool". In: *IEEE Transactions on Learning Technologies - TLT*.

Tvardik, Nastassia et al. (2018). "Accuracy of Using Natural Language Processing Methods for Identifying Healthcare-associated Infections". In: *International Journal of Medical Informatics*.

Uzuner, Özlem et al. (2011). "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text". In: *Journal of the American Medical Informatics Association* 18.5, pp. 552–556.

Van Rijsbergen, Cornelis Joost (1977). "A theoretical basis for the use of co-occurrence data in information retrieval". In: *Journal of documentation* 33.2, pp. 106–119.

Whitelaw, Casey and Jon Patrick (2003). "Named entity recognition using a character-based probabilistic approach". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 196–199.

WHO (2004). "The History of ICD". In: Accessed 28 November 2018, 1–10. URL: `http://www.who.int/classifications/icd/en/HistoryOfICD.pdf`.

Whonamedit? (2010). *A dictionary of medical eponyms*. URL: `http://www.whonamedit.com/azeponyms.cfm`.

Windows, Microsoft (2019). *Windows Server Free Trial*. Accessed 6 May 2019. URL: `https://www.microsoft.com/en-us/cloud-platform/windows-server-trial`.

Wrenn, Jesse O., Peter D. Stetson, and Stephen B. Johnson (2007). "An unsupervised machine learning approach to segmentation of clinician-entered free text". In: *AMIA Annu Symp Proc* 2007. 18693949[pmid], pp. 811–815. ISSN: 1942-597X. URL: `https://www.ncbi.nlm.nih.gov/pubmed/18693949`.

Wu, Dekai, Grace Ngai, and Marine Carpuat (2003). "A stacked, voted, stacked model for named entity recognition". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 200–203.

Yang, Jie and Yue Zhang (2018). "Ncrf++: An open-source neural sequence labeling toolkit". In: *arXiv preprint arXiv:1806.05626*.

Zhang, Tong and David Johnson (2003). "A robust risk minimization based named entity recognition system". In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pp. 204–207.

Zhang, Wei et al. (2011). "Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling". In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*. IJCAI'11. Barcelona, Catalonia, Spain: AAAI Press, pp. 1909–1914. ISBN: 978-1-57735-515-1. DOI: 10.5591/978-1-57735-516-8/IJCAI11-319. URL: `http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-319`.

Økonomi- og Indenrigsministeriet (2019). *Navnelister*. `https://ast.dk/born-familie/hvad-handler-din-klage-om/navne/navnelister`.