Utrecht University

UMC Utrecht

MASTER THESIS

ARTIFICIAL INTELLIGENCE

Identification of adverse drug reactions in Dutch electronic health records

Author:
Gijs MOURITS

First Supervisor:
Dr. Marijn SCHRAAGEN

Daily Supervisor:
Sander TAN

Second Supervisor:
Dr. Meaghan FOWLIE

Research & Data Technology
IT Department, UMC Utrecht

January 26, 2022

"Our intelligence is what makes us human, and AI is an extension of that quality."

Yann LeCun

Abstract

Adverse drug reactions (ADRs) are a common cause of morbidity and mortality. Especially elderly people are more susceptible to ADRs due to multiple factors such as age and polypharmacy. In current clinical practice most data on ADRs is gathered through spontaneous reports, which leads to under-identification of ADRs. Gathering more post-marketing data on ADRs is valuable for research and can improve pharmacovigilance.

In this work we address the problem of under-identification of ADRs by applying natural language processing (NLP) techniques on Dutch electronic health records (EHRs). Specifically, we analyze the admission and discharge letters of 93 patients aged 70 or older, with at least 5 chronic prescribed drugs, admitted through the emergency department to the geriatric department. These letters have been annotated for the presence of adverse drug events (ADEs) and ADRs. Additionally, annotations were provided for when recognition of ADRs was explicitly mentioned in-text by clinicians. The dataset contains 3301 relevant concepts, 337 annotated ADEs and 129 annotated ADRs. We use this dataset as a golden standard to develop NLP models in this field.

Our approach consists of two steps. First we apply and tweak MedCAT, a concept extraction and linking tool. We use this tool to recognize drug and event entities, the components of ADEs, in texts. Next, we perform relation extraction while incorporating information about the entities. For this, We use two existing BERT-based approaches. Instead of using BERT, we use the state-of-the-art belabBERT model that is pre-trained on Dutch corpora. These models are trained to automatically recognize ADRs. We use two relation extraction approaches, one where we search long-range at document-level and one where we identify ADRs mentioned by clinicians at sentence-level. We also provide a baseline model that counts every ADE as ADR.

Our approach for ADE recognition achieves an f1 score of 71.4%. The best model for long-range relation extraction to identify ADRs yields an f1 score of 60.7%, only slightly surpassing the baseline (53.1%). The variant where we identify mentions of ADRs at sentence-level yields an f1 score of 76.9% (for the best model) and manages to outperform the baseline by 18.0%. We argue that with some additional effort such as expanding the annotated dataset and gaining more insight in the relation between ADR mentions and true ADRs, our pipeline could be further optimized to allow implementation into clinical practice for ADR recognition.

Acknowledgements

First of all, I would like to thank Marijn Schraagen, my main thesis supervisor. With his experience in NLP research he would identify obstacles far in advance. I really appreciate the clear, structured and elaborate feedback Marijn provided during my thesis. He gave me insights and tips that have led to increased quality of this thesis. Additionally, thanks goes to Meaghan Fowlie for being my second supervisor and providing feedback.

I would also like to thank Sander Tan, my supervisor at UMC Utrecht. He was always available to answer my questions and I learned a lot from his programming skills. Sander's positivity and NLP enthousiasm motivated me throughout my thesis. The same goes for the rest of the Analytics team, Mark Snackey and Vincent Menger, who made me feel part of the team. I have really enjoyed working together for the past year and appreciate the occasional breaks where we would get Broodje Ben, play games or talk about birds.

Next, I would like to thank Esther Uijtendaal, who has also been very enthustiatic about this project and sacrificed a lot of her time to assist. In addition, I want to thank Wilma Knol and Toine Egberts, who, together with Esther and Sander, were involved in discussions about the clinical aspect of this project. Special thanks also goes to Bastiaan Sallevelt, who carried out the research that this project is largely based on (Noorda et al., 2022). I contacted Bastiaan several times and he was always willing to answer my questions.

Contents

Chapter 1

Introduction

At the department of Information Technology ("Directie Informatie Technologie") in UMC Utrecht, the analytics team is working on text mining tools to improve healthcare. The geriatric department pointed out that there is a lack of structural documentation of ADRs in EHRs which leads to under-identification of ADRs and complicates research towards ADRs (Section 1.1.1). Therefore, the goal of this thesis is to detect adverse drug reactions (ADRs) in Dutch admission and discharge letters of geriatric patients. Such research can improve post-marketing discoveries of ADRs through retrospective studies and lead to early detection of ADRs, as described in Section 1.2.

In this chapter we will formulate the problem and go over key terminology in this field. In Section 1.1 we will discuss the process of pharmaceutical drug development, the difficulty of detecting accessory ADRs and the impact of ADRs on the elderly population. In Section 1.2 we will elaborate on the application of artificial intelligence (AI) in healthcare, specifically natural language processing (NLP), and how some NLP methods such as concept extraction and relation extraction can be used to detect ADRs. In the final section of this chapter, Section 1.3, we provide an overview of the research questions we aim to answer in this thesis.

1.1 Pharmaceutical drugs and Adverse Drug Reactions

In Europe, strict rules apply to the introduction of new pharmaceutical drugs. Generally, pharmaceutical drugs are first extensively tested on animals to test for acute and long-term toxicity. The next stage is studying the effects on humans in clinical trials. Clinical trials consist of four phases which roughly contain the following steps:

- Phase I) Testing on a small population of (typically) healthy volunteers and monitoring how the pharmaceutical drugs are absorbed and whether there are acute side effects (definition explained below).
- Phase II) Determining the efficacy and dose of the pharmaceutical drugs on actual patients.
- Phase III) This is the most extensive phase and includes hundreds or thousands of participants, typically varying in duration from one to four years (USA Food and Drug Administration (FDA), 2018), this makes it possible to observe rarer side effects. Information is gathered on safety and efficacy and the pharmaceutical drug is compared to the default treatment or a placebo.
- Phase IV) Post-marketing surveillance.

There are a few key definitions, as defined by the World Health Organization (WHO), when talking about pharmaceutical drugs (World Health Organization, 2007). **Adverse drug events** (ADEs) are "any untoward medical occurrence that may be present during treatment with a pharmaceutical drug but does not necessarily have a causal relationship with this treatment". **Side effects** are known reactions to a pharmaceutical drug that are related to its pharmacological properties (causal relation). These are registered on the pharmaceutical drug's label or elsewhere. **Adverse drug reactions** (ADRs) are "any response to a drug which is noxious and unintended, and which occurs at doses normally used in man for prophylaxis, diagnosis, or therapy of disease, or for the modification of physiological function". This means that ADEs also include medical errors such as misdosing, administrative errors whereas we only speak of ADRs when causality is suspected. ADRs are therefore a subset of ADEs.

ADRs are the cause of increased morbidity and mortality (Patton and Borshoff, 2018). A meta-analysis that includes 49 studies shows that the mean prevalence of ADRs leading to death was 0.20% (95% CI:0.13-0.27%) (T. K. Patel and P. B. Patel, 2018). ADRs can directly affect patients but also indirectly, as it can lead to unnecessary diagnostics or procedures (Patton and Borshoff, 2018). Among outpatients (non-hospitalized) 52% (95% CI:42-62%) of ADRs were preventable and among inpatients this number was 45% (95% CI:33-58%) (Hakkarainen et al., 2012). These admissions bring along huge costs. In the Netherlands a single ADR-related hospital admission is estimated to cost somewhere between 2132-4915 euros per admission, the annual costs are estimated to be somewhere between 186-430 million euros per year (Beijer and De Blaey, 2002). In a more recent systematic review, covering studies from the US and European countries, costs for ADEs ranged from 702-7318 euros per patient (Marques et al., 2016). Although ADRs form a subset of ADEs and only partly the cause of these costs, it still indicates the significant costs associated with ADRs. Part of these costs are due to prolongation of hospital stay, extra laboratory tests, procedures and treatments (Gautier et al., 2003). Better recognition and early detection of ADRs would not only decrease morbidity and mortality, but also costs.

1.1.1 Over- and under-identification of ADRs

The WHO states that the pre-marketing safety evaluations (animal studies and clinical trials) lead to both over- and under-identification of ADRs. Over-identification takes place because all ADRs that are found during clinical trials are listed as side effects in the pharmaceutical drug's label, even though causality has not been proven (World Health Organization, 2007). Pharmaceutical companies do this to legally protect themselves. Under-identification is caused by those ADRs that are very rare and have not been observed during the clinical trials (World Health Organization, 2007).

One way to gather post-marketing data on ADRs is through spontaneous reports, when patients or health practitioners report the occurrence of ADRs in practice. This is the most valuable method for early detection of ADRs and is also useful for obtaining further information on known ADRs (Härmark, Hunsel, and Grundmark, 2015). This post-marketing surveillance method has a high rate of under-reporting. In a systemic review by Hazell and Shakir (2006), they found that the median underreporting rate of serious ADRs across 37 studies was 94%.

Another post-marketing surveillance strategy is monitoring through clinical studies. Randomized controlled trials (RCTs) and observational studies are done for this purpose. RCTs are expensive and rarely lead to the discovery of unknown ADRs, therefore pharmaceutical companies often lean on observational studies to assess efficacy and safety after pharmaceutical drugs have been released on the market and new issues arise (World Health Organization, 2007). Often these observational studies have too few participants to prove any statistically significant results on causality between pharmaceutical drugs and ADRs.

There are many examples of drugs that were released on the market and had to be recalled due to ADRs (Saluja et al., 2016). It took 20 years after flucloxacillin entered the market to recognize its hepatotoxic effect, especially in elderly. This is one of many examples that illustrates how current post-marketing surveillance approaches under-perform in the identification of ADRs (Routledge, O'Mahony, and Woodhouse, 2004).

1.1.2 Elderly population

Elderly people are more susceptible to ADRs due to multiple factors such as age and polypharmacy (use of multiple pharmaceutical drugs). Polypharmacy can lead to changes in pharmacokinetics (how the body processes the pharmaceutical drug) and interactions between different pharmaceutical drugs (Patton and Borshoff, 2018).

The rate of elderly patients being hospitalized because of an ADR-related event is 16.6% as compared to 4.1% in younger patients. 88% of these ADR-related hospital admissions are potentially preventable in the elderly population (Beijer and De Blaey, 2002). Another study based on Dutch hospitalizations found that the population $\geq$75 years had a more than 4 time higher risk of being hospitalized by ADRs in comparison to the population of 55-64 years (Ruiter et al., 2012).

Hospitals keep track of all kinds of data: patient data, hospital-wide statistics (admissions, length of stay, etc.), financial data. A viable data source for post-marketing surveillance are electronic health records (EHRs). EHRs are patient-centered records that can contain information on medical history, diagnoses, medications, treatment plans, radiology images, laboratory results and more (Office of the National Coordinator for Health Information Technology, 2019). This data is useful for identifying ADRs and is largely captured in the form of natural language, free text. For example, in treatment plans doctors may explicitly mention the presence of ADRs, or the medical history may mention previous drug-induced events. Artificial intelligence can be used to automatically process this data.

1.2 Artificial Intelligence

Although the development of Artificial intelligence (AI) started in the 1950s, it was only in the 1970s that it found its first medical applications: CASNET and MYCIN were among the first AI systems that gave physicians advice on patient management and diagnosis (Kaul, Enslin, and Gross, 2020). The period between 1970 and 2000 is often referred to as the "AI winter", because there were fewer developments and there was less interest in AI development. One development in this period was DXplain, a decision support system released in 1986 (Amisha, Pathania, and Rathaur, 2019). DXplain generated a differential diagnosis based on a set of symptoms. Between 1985 and 2000 interest increased significantly, illustrated by the increased amount of contributions to the international conference on Artificial Intelligence in Medicine (AIME, Peek et al., 2015). Interest renewed because of the opportunities offered by new methods such as machine learning, deep learning and computer vision.

In 1991, Baxt (1991) used an artificial neural network to automatically identify myocardial infarction in patients presenting to an emergency department. In 2007, IBM had developed Watson, a question-answering system. Watson managed to impress the public by winning first place in a television show competing against humans. It used a technology called DeepQA, using natural language processing and statistical machine learning methods, to find the most probable answer in a large database of free text.

Natural Language Processing (NLP) is a sub-domain of AI that focuses on the interpretation of natural language data (both spoken and textual data). It is used for tasks like translation, chatbots, voice assistants, personalized advertising, spam email filtering, and has many more use cases. One of the biggest NLP challenges is that of word sense disambiguation: discover the meaning of a word in its context. NLP also deals with challenges such as spelling mistakes, context, temporal aspects and lexical variations. Biomedical NLP (BioNLP) is the application of NLP on biomedical data. Biomedical language is a language on its own, complex with many acronyms, abbreviations, and domain specific concepts. Therefore, we need to adapt NLP models specifically to this domain, by training and testing them on medical texts.

Taking into consideration the challenges named in the previous sections, a problem that severely affects the elderly population is the under-identification of ADRs. Therefore, we have to come up with new post-marketing surveillance strategies to identify ADRs among the elderly. As mentioned before, hospitals keep track of all kinds of data. In this research, we have access to a dataset of admission and discharge letters from the geriatric department that have been annotated for the presence of ADEs and ADRs (Section 3.2). We are in need of methods to structure this data. NLP offers a possible solution to make this data manageable.

In the past, electronic health record data had to be analyzed manually. Now, there are several NLP techniques that we can use to extract useful information automatically. Methods such as clustering can be used to obtain an overview of the data and can be applied directly to unstructured corpora to, for example, extract the main topics. Another technique is to transform unstructured data into structured data. Structured data can be analyzed more easily. Analysis of this data can lead to post-marketing discoveries of ADRs through retrospective studies and also to early detection of ADRs. These new techniques can be used to improve pharmacovigilance and bypass the disadvantages of traditional methods that lead to under-reporting and bias, as explained in Section 1.1.1 (Alomar et al., 2020).

In this thesis, the main research goal is: *To detect adverse drug reactions (ADRs) in Dutch admission and discharge letters of geriatric patients.* A first step is to extract clinical concepts from text. Clinical concept extraction is already widely used within the medical domain in similar fields like disease and drug-related studies (Fu et al., 2020). This is a strategy that matches clinical phrases with concepts from clinical ontologies. It builds on an NLP technique called *named entity recognition* to detect entities in texts such as diseases, symptoms and drug names. Subsequently it uses entity linking to link medical concepts to these entities (Section 2.1). This way, we end up with clinical texts that are annotated with the corresponding concepts, easing data analysis. This leads to our first sub-question: *How well can our concept extraction model identify concepts from Dutch admission and discharge letters of geriatric patients?* Concept extraction is a multifaceted methodology of which the performance depends on many features and design choices. Some examples: 1) Including certain concept databases to improve concept recognition, or instead excluding databases to prevent false positives. 2) Pre-processing choices concerning cleaning, normalization or spelling correction. 3) Methodology used to disambiguate entities. This leads to the following sub-question: *In what ways can we improve concept extraction to identify concepts from Dutch admission and discharge letters of geriatric patients?*

As a second step, *relation extraction* can be used to automatically extract relations from texts, for example: patient-drug usage, disease-symptoms, drug-adverse effects (ADRs). Therefore, our next sub-question is: *How well can our relation extraction model identify ADRs from Dutch admission and discharge letters of geriatric patients?* There are several approaches to developing a relation extraction model such as rule-based models with regular expressions, non-deep machine learning methods or deep learning methods. Some specific relation extraction methods are described in sections 2.3 and 2.4. In the past few years there has been a shift to deep learning methods because these significantly outperform non-deep machine learning methods (Hahn and Oleynik, 2020). BioBERT (Section 2.4.3) is a well-known example of a deep pre-trained English biomedical language representation model that achieves state-of-the-art results on biomedical texts (Lee et al., 2020). BelabBERT is a similar, although not specific for the biomedical domain, Dutch pre-trained language representation model (Wouts et al., 2021). Because these deep learning methods require a lot of data while annotated data is only scarcely available in the medical domain, the use of traditional models also remains a viable option. This gives rise to the following sub-question: *Can we use a belabBERT-based model for relation extraction or do we need to use less data intensive methods?*. Because we also have annotated entities in our relation extraction task we also want to incorporate this entity information, this leads to another sub-question: *What is the best way of incorporating entity information for relation extraction?*

In this chapter we talked about the process of introducing new pharmaceutical drugs and key terminology related to pharmacovigilance. We mention how under-reporting leads to under-identification of ADRs and how this affects health care for the elderly population. As a solution, we propose using NLP techniques in electronic health records. In Section 2.1 we discuss available resources for NLP in the biomedical domain. As mentioned in this section, there are several NLP techniques to turn free text into useful data, we chose to use clinical concept extraction followed by relation extraction. Examples of methods for clinical concept extraction are described in Section 2.2 and for relation extraction in Section 2.3. In Section 2.4 we go over the specific approaches chosen for this thesis to perform clinical concept and relation extraction. In Chapter 3 the chosen methodology is discussed, this includes a description of the datasets, pre-processing steps and modelling choices. In Chapter 4, the results for both concept and relation extraction are evaluated for different models. Lastly, in Chapter 5 we discuss the results and provide opportunities for future work. Moreover, we describe the potential of this research for clinical practice.

1.3 Research Goals

The main research goal and its corresponding sub-questions as introduced in the previous section:

1. Main goal: To detect adverse drug reactions (ADRs) in Dutch admission and discharge letters of geriatric patients.

   (a) How well can our concept extraction model identify concepts from Dutch admission and discharge letters of geriatric patients?

i. In what ways can we improve concept extraction?

(b) Can we use these identified concepts to recognize ADEs?

(c) Can we use a belabBERT-based model for relation extraction or do we need to use less data intensive methods?

(d) What is the best way of incorporating entity information for relation extraction?

(e) What insights are gained about NLP in Dutch clinical practice?

1.4 Code and Data

Code and data available on request:

- Form: https://www.umcutrecht.nl/en/data-request-form-umc-utrecht
- Contact Sander Tan: S.C.Tan-3@umcutrecht.nl

Chapter 2

Related Work

Natural language processing (NLP) is described as an area of research that explores computational techniques to analyze and represent natural human language (both spoken and written) for a range of tasks or applications (Liddy, 2001; Chowdhury, 2003). The challenge is to enable automatic analysis of unstructured data, natural human language. This enables large scale analysis of the data with the ability to generate meaningful insights. It can be used for many purposes, some examples are: translation, spam email filtering, voice assistants, personalized advertising and text categorization.

2.1 Biomedical NLP

First, we will go over the broad concept of biomedical NLP, as the text data we will be working with is biomedical data. Biomedical NLP (BioNLP) is the application of NLP on biomedical data, generally dealing with two types of texts: scientific articles and clinical documents (Huang and Lu, 2016; Cohen, 2013). Within the biomedical domain the language is complex (acronyms, abbreviations, etc.) and there is a large variety of textual data that is also highly heterogeneous, BioNLP is therefore a challenging problem. However, there are a variety of available resources to make BioNLP easier. In this section we will go over available resources and explain how some of these are relevant to our research and how some are not.

2.1.1 Language Corpora for Pre-Training

Language corpora are an essential building block for many NLP tools. As this thesis is set in Dutch context, we aim for biomedical language corpora in Dutch. However, most available corpora are not useful for this thesis as they are in English. In this section we describe some of these English corpora, in Section 3.4 we give an overview of the Dutch language corpora used for our unsupervised training step.

Language corpora can be used for supervised learning, such as summarization tasks: an example is where a model needs to learn how to generate abstracts from articles (with articles as input and their abstracts as target). More common, is using language corpora for unsupervised learning, such as the pre-training of models like BERT (Devlin et al., 2018), where BERT learns contextual word embeddings from English texts. Bigger corpora do not necessarily lead to better biomedical domain word embeddings (Chiu et al., 2016). What matters more, is how well a corpus represents the domain that the NLP tool is applied to (Y. Wang et al., 2018). Here are some examples:

- The open research corpus contains 81.1M text drawn from English academic papers across many fields of study of which 12.8M belong to the discipline of medicine (Lo et al., 2019). The corpus contains around 25B tokens in full article texts, and 15B tokens in abstracts. This corpus also contains a collection of articles drawn from MEDLINE.

- MEDLINE is a database that contains more than 28 million references to biomedical academic papers (National Library of Medicine, 2021a). These references link to more than 5200 journals in around 40 different languages, of which more than 80% is in English.
- MIMIC-III is an English dataset that contains around 50k hospital admissions to critical care units (Johnson et al., 2016). What makes MIMIC unique is the variety of classes of data it contains on each patient: demographic, laboratory, medication, notes, billing and more.

2.1.2 Annotated Corpora

When training machine learning models to recognize advanced features in texts it can be useful to annotate the data, creating opportunities for specific supervised learning tasks. We will be evaluating the performance of our model at concept recognition of drugs and events in Dutch clinical texts, so we seek a dataset with Dutch texts annotated for drugs and events. In Section 3.2 we describe the annotated dataset that we ended up using in our research. Below we provide some examples of other datasets annotated for concepts. We cannot use most of these as they contain English annotated texts. In the few Dutch concept annotated datasets, they either focus on an entirely different concept category (not drugs and events, but genetic for e.g.) or the annotated drugs and events make up only a very small part of the dataset. Examples are:

- The Mantra GSC is a multilingual annotated corpus. The terms in the texts are annotated with their corresponding UMLS concept unique identifiers from three vocabularies: MeSH (Lowe and Barnett, 1994), SNOMED-CT (SNOMED, 2021) and MedDRA (Mozzicato, 2009). In total it contains 5530 annotations across five languages: English, French, German, Spanish and Dutch.
- The GENIA corpus is a collection of 2500 English abstracts (not all available to the public) from the MEDLINE database, relating to the transcription factors in human blood cells (Kim et al., 2003). GENIA contains annotations on six categories: part-of-speech, constituency syntax, terms, events, relations and co-referential expressions.
- The NCBI disease corpus (Doğan, Leaman, and Lu, 2014) contains 793 English PubMed abstracts that are annotated for disease mentions and linked to corresponding concepts using the concept databases MeSH and OMIM. It contains annotations for 6892 disease mentions, linked to 790 unique disease concepts.

2.1.3 Concept Databases

A challenge in natural language is that unique concepts are expressed in various ways. Take the unified medical language system (UMLS) concept "feeling sick" for example, this can be expressed as "malaise", "feeling bad", "ill feeling", among many other terms (Bodenreider, 2004). In a concept database, these terms are linked to a unique concept. These concepts can therefore serve as a coding terminology for information in texts. The UMLS Metathesaurus (Bodenreider, 2004) brings together many biomedical lexical resources such as MedDRA (Mozzicato, 2009), SNOMED (SNOMED, 2021), ICD-10 (SNOMED, 1993), LOINC (McDonald et al., 2003) and more. Since UMLS also contains many Dutch ontologies, we will use this for our project. UMLS will be further described in Section 2.4.1.

2.1.4 NLP Tools

There are many tools available to process texts for the general NLP domain: sentence boundary detectors, tokenizers, normalizers, named entity recognition systems (NERs). Many of these show poor performance when applied to a different/specific domain (Neumann et al., 2019). Unfortunately, alternatives that are tailored to the biomedical domain have mostly been developed specifically for English texts, so we will not be using those. Below are some examples:

- SciSpaCy is designed especially for processing biomedical texts and can do tokenization, part of speech tagging, dependency parsing and named entity recognition (Neumann et al., 2019). SpaCy, the underlying architecture of SciSpaCy, also has pipelines trained specifically for Dutch language (spaCy, 2021a).
- BioLemmatizer is a lemmatization tool specifically for the English biomedical domain (H. Liu et al., 2012). It outperforms several existing lemmatizers.

2.2 Clinical Concept Extraction

Many BioNLP systems have been developed for general clinical concept extraction. Typically, clinical concept extraction tools consist of a named entity recognition (NER) and a named entity linking (NEL) component, this combination can be abbreviated as NER+L. According to Mohit (2014), named entity recognition is the problem of locating and categorizing important nouns and proper nouns in a text. An example: "Jim was diagnosed with a post-traumatic brain syndrome", where entities "Jim" and "post-traumatic brain syndrome" can respectively be categorized as a person and a disease. The next step is NEL, the task at hand is to link the found entities to the corresponding values in a knowledge base. Assume we are linking the disease to the UMLS knowledge base (Section 2.4.1) in the above example. Then we link the entity "post-traumatic brain syndrome" to the corresponding concept "post-concussion syndrome" in our concept database.

These extraction methods are adaptations from the general NLP domain and can be roughly divided into four categories: rule-based, non-deep machine learning, deep learning and hybrid approaches (Fu et al., 2020). Although the majority of research in clinical concept extraction is rule-based, the amount of deep learning based approaches found the largest relative increase in the past five years (Fu et al., 2020).

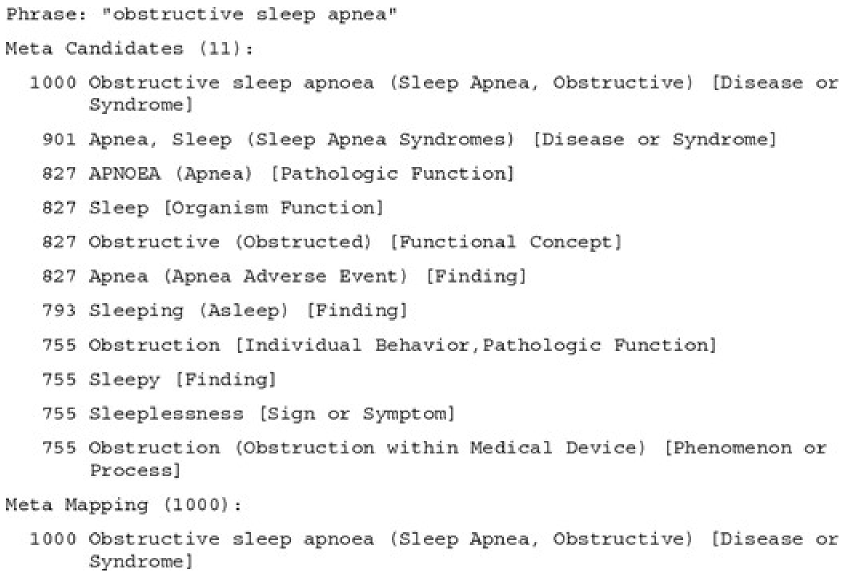
We will use MedCAT for clinical concept extraction, which we discuss in-depth in Section 2.4.2 (Kraljevic et al., 2021). Examples of tools similar to MedCAT are discussed in this section, these are MetaMap (rule-based) (Aronson, 2006), cTAKES (traditional machine learning) (Savova et al., 2010), Bio-YODIE (Gorrell, Song, and Roberts, 2018) and SemEHR (H. Wu et al., 2018). The reason that MetaMap and cTAKES are not appropriate for this thesis, is that they: 1) Can only be applied to English texts. 2) They are much slower than other tools (like MedCAT and BioYODIE). Bio-YODIE and SemEHR are faster and, similar to MedCAT, they can use any subset of UMLS which means they can also handle Dutch texts. However, we do not use Bio-YODIE or SemEHR as MedCAT significantly outperforms these, which can be seen in Table 2.3. In the next sections, we will briefly discuss the methodology of each of these alternative tools.

2.2.1 MetaMap

MetaMap was initially designed to improve the biomedical text search engines such as MEDLINE/PubMed. It links the input phrase to the corresponding UMLS concept (Section 2.4.1) and identifies candidate phrases that could provide similar results. It is also applicable on the general use case of mapping biomedical text to concepts in the UMLS Metathesaurus. Inherent to the methodology, MetaMap can only be applied to English texts and is also very slow, and therefore not useful for this study. The methodology is as follows (Aronson, 2006):

First, the text (input query) undergoes tokenization, sentence boundary detection and acronym/abbreviation identification. This is followed by part-of-speech tagging and the biomedical terms are detected using the SPECIALIST lexicon (National Library of Medicine, 2021b). Then, this is parsed into noun phrases using the SPECIALIST parser (a non-machine learning parser). For each noun phrase:

- I) Variant generation: generate all variants such as synonyms, acronyms, abbreviations, spelling variants, inflectional and derivational variants, and combinations of these.
- II) Compile set of candidates containing all UMLS strings that contains one of the generated variants:



```
Phrase: "obstructive sleep apnea"
Meta Candidates (11):
  1000 Obstructive sleep apnoea (Sleep Apnea, Obstructive) [Disease or
       Syndrome]
   901 Apnea, Sleep (Sleep Apnea Syndromes) [Disease or Syndrome]
   827 APNOEA (Apnea) [Pathologic Function]
   827 Sleep [Organism Function]
   827 Obstructive (Obstructed) [Functional Concept]
   827 Apnea (Apnea Adverse Event) [Finding]
   793 Sleeping (Asleep) [Finding]
   755 Obstruction [Individual Behavior,Pathologic Function]
   755 Sleepy [Finding]
   755 Sleeplessness [Sign or Symptom]
   755 Obstruction (Obstruction within Medical Device) [Phenomenon or
       Process]
Meta Mapping (1000):
  1000 Obstructive sleep apnoea (Sleep Apnea, Obstructive) [Disease or
       Syndrome]
```

FIGURE 2.1: An example of candidate generation (Aronson and Lang, 2010)

- III) Evaluation function that scores the found candidates on quality of the match. This evaluation function consists of four components: centrality, variation, coverage and cohesiveness. Coverage and cohesiveness are assigned twice the weight in the evaluation function.
  - Centrality: a boolean value, whether (part of) the string corresponds with the head of the phrase.

  - Variation: a custom function to estimate how similar concepts from the UMLS are to the words in the phrase. Difference in spelling, inflectional and derivational forms, and a word being a synonym or acronym/abbreviation, are all taking into account in the function.
  - Coverage: a value to indicate the amount of overlap between an UMLS concept and the phrase. The amount of words overlapping and the length of these are used to compute this value.
  - Cohesiveness: this is about the connected components of the string, the maximal sequence of adjacent words that are part of the match.

- IV) Final evaluation: compares the top candidates of the several noun phrases and picks the overall best ones.

A few studies have been carried out to evaluate the performance of MetaMap. When applied to medical school lecture documents containing 4281 annotated concepts, MetaMap achieved 78% recall and 85% precision (Denny et al., 2003). In a more recent evaluation, MetaMap was compared to cTAKES (Section 2.2.2) and applied to the i2b2 Obesity dataset consisting of 1237 discharge summaries of overweight and diabetic patients (Reátegui and Ratté, 2018; Uzuner, 2009). MetaMap reached a 88% recall, 89% precision and 88% F-score.

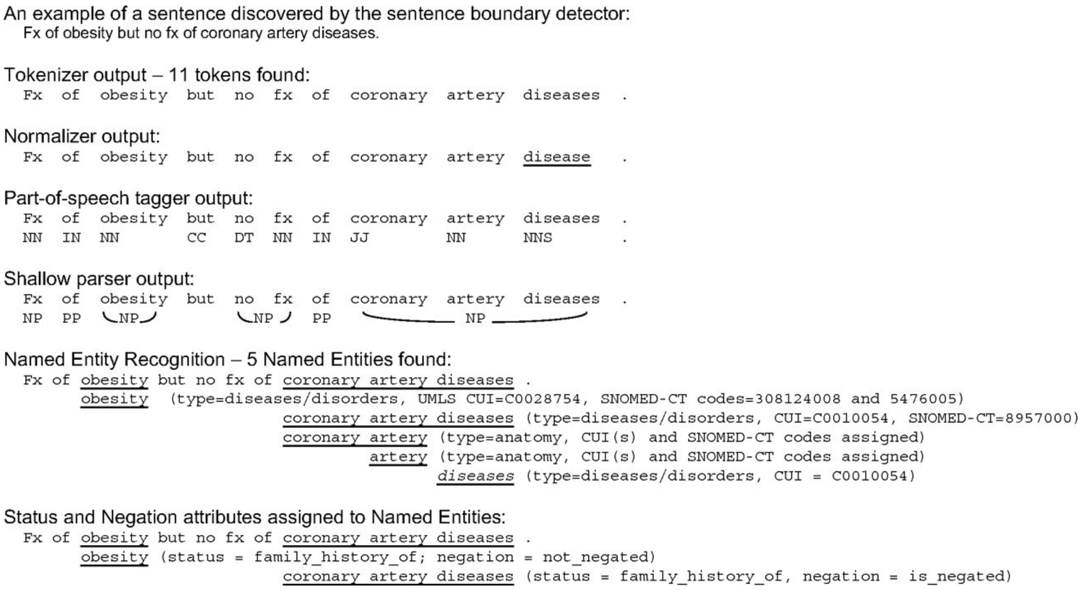
As MetaMap builds upon English-based syntactic algorithms and on the English SPECIALIST lexicon, it can only be applied to English text. It is also not applicable for real-time use, as processing phrases that have many potential mappings can still take several hours (Aronson and Lang, 2010). A more recent adapted version of MetaMap, is MetaMap Lite: it achieves faster processing and better performance than MetaMap and cTAKES on several datasets (Demner-Fushman, Rogers, and Aronson, 2017). However, the lite version comes with fewer features (hence "lite" in the name), can still only be applied to English text, and does not yield any form of word-sense disambiguation. Without word-sense disambiguation, a term will be linked to all corresponding concepts in UMLS.

2.2.2 cTAKES

Clinical Text Analysis and Knowledge Extraction System (cTAKES) is an open-source NLP tool designed for IE from free text in EHRs (Savova et al., 2010). We will not be using cTAKES in this thesis as it can only be used for English texts and is too slow for real-time implementation. It uses both rule-based and machine learning techniques as follows:

- I) OpenNLP (Apache Software Foundation, 2014) maximum entropy (ME) classifier sentence boundary detector.
- II) Rule-based tokenizer is used, followed by token normalization based on the SPECIALIST Lexical Tools (National Library of Medicine, 2021b).
- III) OpenNLP ME part-of-speech (POS) tagger to assign POS tags to the tokens.
- IV) OpenNLP ME shallow parser: shallow parsing combines POS tags into higher order units such as noun phrases.
- V) Named entity recognition (NER): this component matches the identified noun phrases to a subset of the UMLS dictionary (terms from SNOMED CT and RxNORM, further explained in Section 2.4.1). This results in the clinical

text being annotated with the corresponding UMLS concept unique identifiers (CUI). Negation and status annotations are also added.



```
An example of a sentence discovered by the sentence boundary detector:
   Fx of obesity but no fx of coronary artery diseases.

Tokenizer output – 11 tokens found:
   Fx  of  obesity  but  no  fx  of  coronary  artery  diseases  .

Normalizer output:
   Fx  of  obesity  but  no  fx  of  coronary  artery  disease   .

Part-of-speech tagger output:
   Fx  of  obesity  but  no  fx  of  coronary  artery  diseases  .
   NN  IN  NN       CC   DT  NN  IN  JJ        NN      NNS       .

Shallow parser output:
   Fx  of  obesity  but  no  fx  of  coronary  artery  diseases  .
   NP  PP    NP          NP      PP             NP

Named Entity Recognition – 5 Named Entities found:
   Fx of obesity but no fx of coronary artery diseases .
         obesity  (type=diseases/disorders, UMLS CUI=C0028754, SNOMED-CT codes=308124008 and 5476005)
                              coronary artery diseases (type=diseases/disorders, CUI=C0010054, SNOMED-CT=8957000)
                              coronary artery (type=anatomy, CUI(s) and SNOMED-CT codes assigned)
                                       artery (type=anatomy, CUI(s) and SNOMED-CT codes assigned)
                                              diseases (type=diseases/disorders, CUI = C0010054)

Status and Negation attributes assigned to Named Entities:
   Fx of obesity but no fx of coronary artery diseases .
         obesity (status = family_history_of; negation = not_negated)
                              coronary artery diseases (status = family_history_of, negation = is_negated)
```

FIGURE 2.2: An example query as processed by cTAKES (Savova et al., 2010)

cTAKES was evaluated on an annotated dataset consisting of 160 clinical notes from the Mayo Clinic EMR, containing 1566 annotated concepts (Ogren, Savova, Chute, et al., 2008). For exact matches cTAKES achieved a recall of 64.5%, precision of 80.1% and a F-score of 71.5% (Savova et al., 2010). In the same evaluation paper as mentioned in Section 2.2.1, cTakes slightly outperformed MetaMap on the i2b2 Obesity dataset (Reátegui and Ratté, 2018; Uzuner, 2009) with recall 91%, precision 89% and F-score 89%. Although cTAKES is faster than MetaMap (respectively 35m13s and 52m15s on the i2b2 dataset in Demner-Fushman, Rogers, and Aronson (2017)), it is still very slow.

2.2.3 Bio-YODIE

Bio-YODIE takes a similar approach to biomedical NER+L as MetaMap Lite, however with more focus on disambiguation (Gorrell, Song, and Roberts, 2018). Although Bio-YODIE is fast, can be applied to Dutch texts, and provides disambiguation features, it is outperformed by MedCAT, so we will not be using Bio-YODIE. Bio-YODIE consists of two steps. First is the resource preparation step, where (a subset of) UMLS is pre-processed into a form that can be used as efficient as possible, to speed up the whole process. Then there is the annotation step. Both are described below:

- I) Resource preparation consists of a set of steps that can turn (a subset of) UMLS into a gazetteer (a list with all terms of interest) and a database with concepts linked to terms:
  - Synonym acquisition by linking terms to alternative terms for the same concept in UMLS. Words very similar (based on Levenshtein distance) were labeled as synonyms, leftover words as well. For example, "gout rheumatic" and "rheumatoid arthritis" would first match "rheumatic" and

"arthritis" as synonyms and subsequently match the leftover "gout" to "rheumatoid".
    - A threshold value was set to reject synonyms with too many alternative terms available for its UMLS concept.
    - Manual editing to remove problematic linked synonyms.

- II) Annotation pipeline:
    - Locates potential entities by running the gazetteer.
    - Remove stop words.
    - Generate a list of candidates: these are the concepts from the database that correspond to the term. If there is only one applicable concept, pick that one, otherwise pick the most likely candidate based on:
        - Prior likelihood of a concept.
        - Co-occurrence graph of how related concepts are to each other.
        - Calculating vector representation for the term and comparing to word2vec embeddings of the concepts that are calculated over PubMed.

In an evaluation against MetaMap and MetaMap Lite, Bio-YODIE scores better than MetaMap at recall, precision and f1 score, but only beats MetaMap Lite at recall (Table 2.1). An advantage of Bio-YODIE is that, similar to MedCAT, any subset of UMLS can be used in the resource preparation step which means it can also handle non-English languages. It is also fast, because instead of generating candidates through typical pre-processing steps (spelling variation, inflectional forms, derivational forms, abbreviations) it uses the custom synonym acquisition trick to generate a much smaller list of candidates that is still effective. A speed comparison can also be seen in Table 2.1.

	Secs	**Prec L**	**Rec L**	**FIL**	**Acc**	**Scott's Pi**
MetaMap	3811	0.574	0.568	0.571	0.857	0.856
MetaMapLite	986	**0.654**	0.549	**0.597**	0.877	0.876
Bio-YODIE	**573**	0.582	**0.605**	0.593	**0.883**	**0.882**

TABLE 2.1: A performance comparison between MetaMap, MetaMap Lite and Bio-YODIE (Gorrell, Song, and Roberts, 2018)

2.2.4 SemEHR

SemEHR builds upon Bio-YODIE: it generates output by using Bio-YODIE and then applies manual rules to improve the result (H. Wu et al., 2018). However, in the evaluation on several clinical concept extraction tools by Kraljevic et al. (2021), Bio-YODIE still slightly outperforms SemEHR. MedCAT is still the preferred tool, outperforming both SemEHR and Bio-YODIE. Because SemEHR is very similar to Bio-YODIE, we will not discuss it in-depth here.

2.3 Relation Extraction

In general, relation extraction is the task of identifying semantic relationships between two or more entities in a text (Jiang and Zhai, 2007). In this section, we will first describe the types of relation extraction and explain that there are many different methods for it. In the following in subsection we will go over several relation extraction approaches. We will not be using these, the methodology that we use will be described in Section 2.4.5.

First of all, we differentiate between different types of relation extraction:

- *Traditional relation extraction methods* are based on labeled data. A target relation (in our case: 'drug causes event') is defined and fed to a model as input along with annotated samples (Banko and Etzioni, 2008). A sub-task of relation extraction here is slot-filling. In our case slot-filling would mean that for a given 'drug' and the given relation 'drug causes event', the model has to fill in the slot for 'event'. A disadvantage of traditional relation extraction is that domain-specific labelled datasets are scarce and they need to be annotated for the desired relation (Lange Di Cesare et al., 2018). Another disadvantage is that these methods often require several NLP components, such as NER, before slot-filling can take place. So the performance depends on the whole pipeline (Adel, Roth, and Schütze, 2016).

- *Open relation extraction methods* can use any text data. It requires large amounts of text data and can derive a wide variety of relations (Lange Di Cesare et al., 2018). This technique organizes semantically similar relations into clusters (R. Wu et al., 2019). However, this makes acquiring the exact semantics of such a relation difficult.

We see many general-domain relation extraction approaches in the SemEval-2010 Task 8 challenge by Hendrickx et al. (2010). They created a dataset with 10717 annotated examples choosing from a set of 9 semantic relations. The participating teams had to overcome two challenges: predict the semantic relation type, as well as the direction of this relation. The baseline model was a Naïve Bayes classifier, only looking at the local context of 2 words. In an overview of all the models used by the participants, we see a wide variations of machine learning strategies and textual features being used (Figure A.1). Classification strategies included Bayesian networks, support vector machines (both binary and multi-class), maximum entropy models, two-step classification, conditional random fields and decision rules/trees. The models also cover a wide range of textual features: POS, context words, dependencies, paraphrases, capitalization properties, syntactic patterns, and more. A typical relation extraction pipeline is shown in Figure 2.3.

Similar to the SemEval-2010 Task 8 challenge, we want to perform relation extraction, but now in the medical domain. We want to identify causal relations in sentences like "in patient X, [Medication] has led to [Event]". In the sections below we will briefly discuss two relation extraction approaches in the medical domain: MeTAE and a shortest dependency path method. We will not be using these methods for our project because they both focus at short-range dependencies whereas we also want to develop a strategy that is able to capture long-range ADRs in texts.

2.3.1 MeTAE

Abacha and Zweigenbaum (2010) used an approach consisting of two components. First, they used an adapted version of the biomedical concept extraction tool MetaMap. Second, they compiled a list of patterns to include all possible relations between the semantic types of the identified UMLS concepts. An example of such a relation is "[Pharmacological substance] -> causes -> [Medical problem]", where "causes" is the

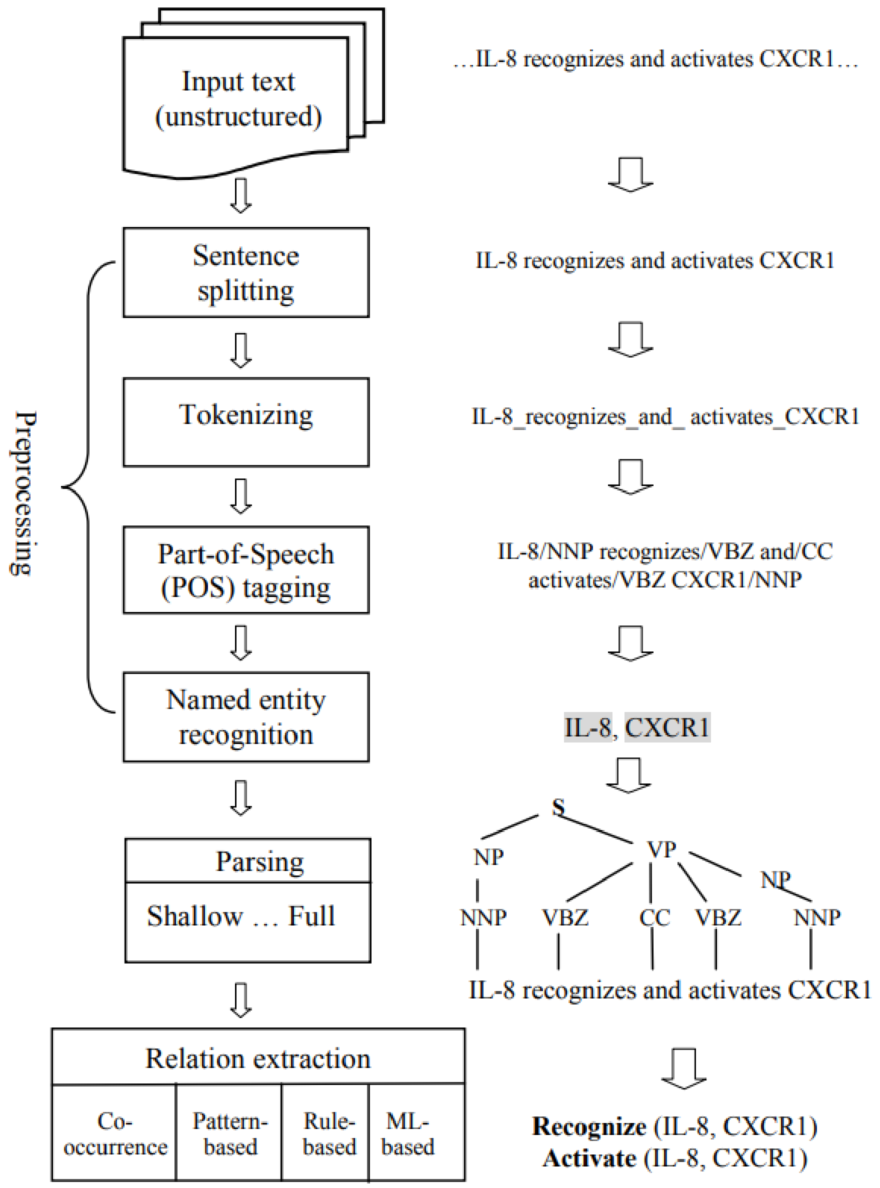

Input text (unstructured)
Sentence splitting
Tokenizing
Part-of-Speech (POS) tagging
Named entity recognition
Parsing
Shallow … Full
Relation extraction
Co-occurrence
Pattern-based
Rule-based
ML-based
Preprocessing
…IL-8 recognizes and activates CXCR1…
IL-8 recognizes and activates CXCR1
IL-8_recognizes_and_ activates_CXCR1
IL-8/NNP recognizes/VBZ and/CC activates/VBZ CXCR1/NNP
IL-8, CXCR1
S
NP
VP
NP
NNP
VBZ
CC
VBZ
NNP
IL-8 recognizes and activates CXCR1
Recognize (IL-8, CXCR1)
Activate (IL-8, CXCR1)

FIGURE 2.3: Typical pipeline for a relation extraction model (Bui et al., 2012).

relation. There are many forms in which a single relation type can appear, "X may trigger Y", "X can cause Y", etc. A challenge is to identify the linguistic patterns in which these relations appear. Together, these two components, clinical concept extraction followed by pattern matching, form MeTAE.

To develop MeTAE, Abacha and Zweigenbaum (2010) extracted articles from PubMed and applied MetaMap for concept extraction. UMLS contains information on semantic relation types that can appear between concepts. They used this to keep only sentences with at least one pair of concepts that has one of the desired UMLS semantic relations. The remaining text contains many sentences serving as examples for how relations appear. From this they were able to construct a set of patterns which were translated to regular expressions for relation extraction. Their model reached 60.5% recall, 75.7% precision and F-score of 67.2%.

2.3.2 Shortest Dependency Path

Ningthoujam et al. (2019) focused on only relation extraction. They used a dataset that was already manually annotated for clinical concepts. Only sentences that contain two entities were selected and put through a POS tagger and subsequently passed on to a dependency parser. The dependency tree that follows from this is used to extract the shortest dependency path (SDP) between two medical concepts in the tree. Next, the words and concept sequence corresponding with the SDP, the dependency labels sequence and the POS tags are used as input for a long-short term memory (LSTM) network. The LSTM is trained to predict the probability of the different possible relations between the two entities.

They used the i2b2b-2010 relation extraction challenge dataset which consists of patients' discharge and progress notes (Uzuner et al., 2011). The whole dataset consists of 394 documents containing 5264 relations for training and 477 documents containing 9069 relations for testing. Ningthoujam et al. (2019) only had access to a subset of this, 170 documents for training and 256 documents for testing. Annotations were present for eight relations in three categories:

- Medical problem - Treatment relations:
  - Treatment improves medical problem (TrIP)
  - Treatment worsens medical problem (TrWP)
  - Treatment causes medical problem (TrCP)
  - Treatment is administered for medical problem (TrAP)
  - Treatment is not administered because of medical problem (TrNAP)
- Medical problem - Test relations:
  - Test reveals medical problem (TeRP)
- Medical problem-Medical problem relations:
  - Test conducted to investigate medical problem (TeCP)
  - Medical problem indicates medical problem (PIP)

The performance was compared to a paper by Sahu et al. (2016), that used the exact same data but based on a domain invariant convolutional neural network (CNN). In this paper they explain that they excluded categories TrWP, TrIP and TrNAP because of the lack of training samples present in the dataset. This might explain why Ningthoujam et al. (2019) only reported performance on the five categories that had more training samples, as can be seen in Table 2.2.

Relation Type	**Proposed model**	**Sahu et al. (2016) model**
TeCP	**59.13**	50.56
TrCP	**62.13**	56.44
PIP	59.38	**64.92**
TrAP	**75.35**	69.23
TeRP	**83.86**	81.25

TABLE 2.2: Comparison of performance in relation extraction (Ningthoujam et al., 2019).

2.4 NLP in Electronic Health Records

An Electronic Health Record (EHR) is a collection of medical data on patients that is stored digitally. It may include any health related data such as medical history, clinical notes, laboratory data, radiology reports and discharge letters. These data are mostly in the form of unstructured text (free text). Medical texts are heterogeneous in nature as they contain (ambiguous) abbreviations, acronyms and spelling mistakes. The challenge is to enable automatic large scale analysis on this unstructured data. An additional challenge is that we target Dutch EHRs. The availability of lexicons, terminologies and annotated corpora for other languages than English is limited (Névéol et al., 2018). Tools such as UMLS are finding increased integration with languages other than English. Another way to avoid the lack of language specific resources is deviating towards unsupervised methods (Névéol et al., 2018).

To show that NLP in EHRs can yield promising results, we provide some (English EHR-based) examples:

- Detection of colorectal cancer (CRC) was done in two steps (Xu et al., 2011): 1) Use of MedLEE for clinical concept extraction of positive CRC concepts (Friedman et al., 2004). 2) Performance comparison of a rule-based approach and four machine learning methods: random forest, ripper, support vector machine and logistic regression.

- Detection of geriatric syndromes by using a rule-based approach based on linguistic patterns (Kharrazi et al., 2018). Experts manually identified geriatric syndromes and analyzed the patterns in which these occurred to detect geriatric syndromes in free text.

- Detection of suicidality in adolescents with autism spectrum disorder (Downs et al., 2017): First experts developed a suicidality terminology. Then they developed a set of rules to classify mentions of suicidality as positive, negated or unknown. They used a majority based approach (positive vs negated mentions) to label a document as positive or negative for suicidality.

2.4.1 Unified Medical Language System (UMLS)

There are a variety of biomedical vocabularies in circulation such as ICD, MeSH, MedDRA and SNOMED-CT (National Library of Medicine, 2020a). The problem is that these all operate separately from each other: they use different terminologies. The unified medical language system (UMLS) is a collection of many biomedical vocabularies. The purpose is to map these different vocabularies to each other and provide a unified terminology system (Humphreys et al., 1998). It contains over 16 million names for over 4 million concepts and draws this data from 157 distinct sources (National Library of Medicine, 2020a). From the most recent 2021 release 70,88% of the names were in English and 1,83% in Dutch, in total 25 languages contribute to the concepts in UMLS. As we will be using UMLS in our research, we will describe its structure.

Concepts that are part of the UMLS Metathesaurus are assigned a unique identifier and added to the UMLS structure. This structure is specified among four levels (for an example, see Figure 2.4), listed below from broad to narrow (National Library of Medicine, 2020b):

- Concept Unique Identifier (CUI): a concept is a meaning, and a meaning can be expressed with different names (also referred to as "terms" in this thesis). One of the main goals of the UMLS is to link all possible names from the different source vocabularies to the corresponding concept. These are unique codes, starting with a "C" followed by 7 digits, these codes themselves do not bear any meaning but stay consistent throughout new releases.

- Lexical Unique Identifier (LUI): are strings that are lexical variants of each other map to the same LUI. Code notation: "L" followed by 7 digits.
- String Unique Identifier (SUI): for each variation in upper-lower case, punctuation or characters a separate SUI is assigned. Code notation: "S" followed by 7 digits.
- Atom Unique Identifier (AUI): each concept names/terms from each of the source vocabularies gets a unique atom identifier. So even if two terms are exactly the same but originate from a different source, they get a different AUI. Code notation: "A" followed by 7 digits.

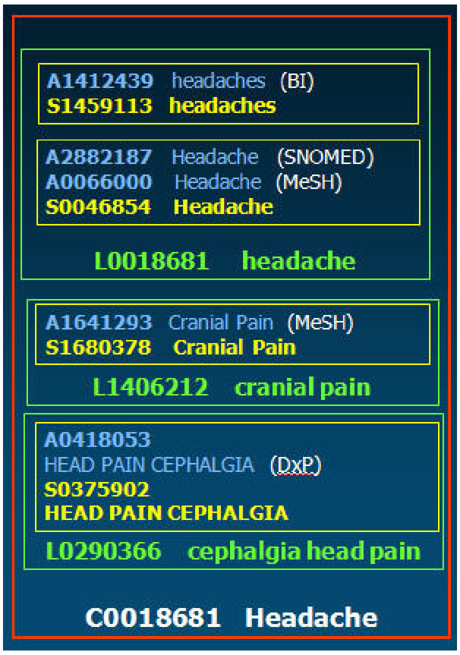


FIGURE 2.4: UMLS Metathesaurus structure, the four levels of specification (National Library of Medicine, 2020b).

2.4.2 MedCAT

MedCAT is the open source Medical Concept Annotation Toolkit (MedCAT, Kraljevic et al., 2021) that we will use in this project. It consists of several components. First, it uses a dictionary approach to match concepts to words in the input text, along with spell-checking and text cleaning. Sometimes, words are mapped to multiple concepts. In these cases, MedCAT uses a unsupervised machine learning algorithm to disambiguate concepts. In this section we will first break down these components, followed by an explanation of why MedCAT may perform well in clinical concept extraction in the medical domain.

2.4.2.1 Vocabulary and Concept Database

MedCAT needs to compile a vocabulary (VCB) and concept database (CDB) to work. The VCB contains all the words that can appear in the clinical documents that are to be annotated. It can be generated from any textual data, Kraljevic et al. (2021) use Wikipedia and add words from UMLS to it, in Section 3.4 we describe our approach. The VCB's main purpose is to serve for spell checking, as will be explained later in this section. The second component, the CDB, consists of a table with CUIs and terms, derived from UMLS as will be explained in Section 3.1.

2.4.2.2 Spell-checking and Text Cleaning

Now that we have a VCB and CDB we start by correcting the spelling of input documents. This is done using a spell corrector tool by Norvig (2007). Words of the input documents are compared to words in the VCB and corrected based on both word frequency and edit distance into the most likely option. Longer words allow a greater edit distance and abbreviations are never corrected. MedCAT uses SpaCy for tokenization, lemmatization and stop word removal (spaCy, 2021b).

2.4.2.3 Entity Candidates

Finding the entities that can potentially be assigned a concept goes as follows for each single given input document:

1. Set two variables, window_length = 1 and word_position = 0.
2. Three possible scenarios:
   (a) Current window is a concept from our CDB, if this is the case, the text is highlighted and we skip step 3.
   (b) The found string is a substring of a longer concept term, go to step 3.
   (c) No match, set window_length = 1 and word_position += 1, repeat step 2.
3. Set window_length += 1, repeat step 2.

Steps 3 and 4 solve the fact that concepts can be substrings of other concepts as shown in Figure 2.5.

2.4.2.4 Unsupervised Training

MedCAT also contains an unsupervised machine learning algorithm, which learns to disambiguate concepts by training on an unlabelled dataset. This component only needs to be applied to those words that can map to multiple concepts. For example "RA", which can map to the concepts "Rheumatoid Arthritis" and "Right Atrium".

In order to train this component, context similarity is used. After the previous step, see Section 2.4.2.3, we first select only those terms that unambiguously link to a concept. For those, we learn the context embeddings, based on the word embedding method Word2Vec (Mikolov et al., 2013). An example of this can be seen in Figure 2.5 where a context embedding (vector) for "Altered Mental Status" is created using 5 words on either side. Now, when we have an entity candidate that links to multiple concepts, its context is compared to the existing concept embeddings and it is linked if the similarity is above a threshold.

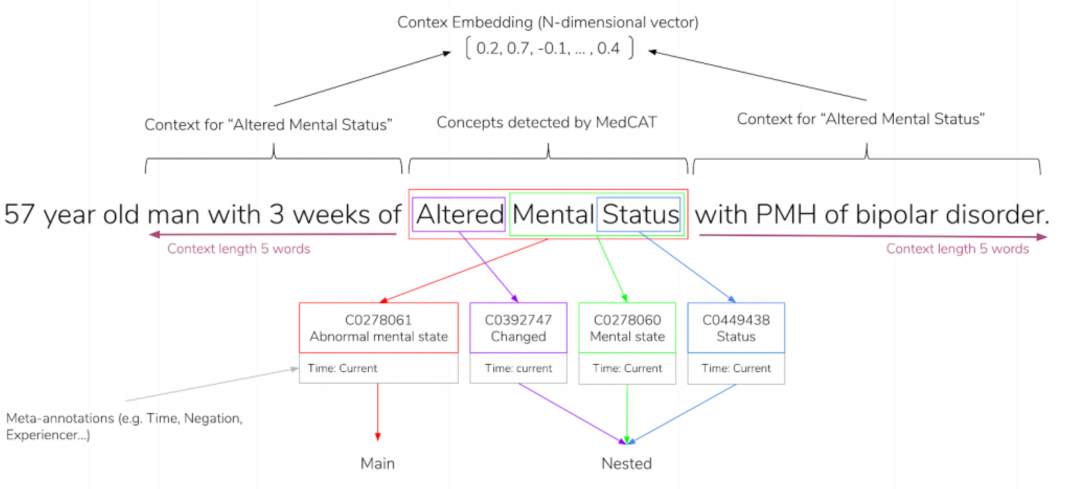

Contex Embedding (N-dimensional vector)
[0.2, 0.7, -0.1, ... , 0.4]
Context for "Altered Mental Status"
Concepts detected by MedCAT
Context for "Altered Mental Status"
57 year old man with 3 weeks of Altered Mental Status with PMH of bipolar disorder.
Context length 5 words
Context length 5 words
C0278061
Abnormal mental state
Time: Current
C0392747
Changed
Time: current
C0278060
Mental state
Time: Current
C0449438
Status
Time: Current
Meta-annotations (e.g. Time, Negation, Experiencer...)
Main
Nested

FIGURE 2.5: Example with nested concepts, meta-annotations and context embeddings (Kraljevic et al., 2021).

2.4.2.5 Performance in the Medical Domain

MedCAT is specifically useful for the digital hospital environment because it integrates in CogStack (Jackson et al., 2018) which is a platform that is developed to enable searching through any clinical data source.

Kraljevic et al. (2021) write that NER+L tools like MetaMap and cTAKES fail to handle ambiguous concepts and spelling mistakes. Because these approaches are based on supervised learning, they need more training data to successfully deal with these issues. Therefore Kraljevic et al. (2021) developed MedCAT as an effective solution to deal with spellings mistakes, form variability and disambiguation. The fact that it is unsupervised eliminates the need for large annotated datasets, which are generally scarce, especially considering the privacy of patients in the medical domain.

In a comparison by Kraljevic et al. (2021), performances for several clinical concept extraction tools were evaluated. All tools were implemented with their default models and an annotation was counted as correct if the exact string was matched and linked to the correct UMLS concept. MedCAT scored the best results overall (based on f1 score), as can be seen in Table 2.3. Kraljevic et al. (2021) argue that all tools perform well on the ShARe/CLEF dataset due to the lack of ambiguity in that dataset. They calculated that 40% of the concepts in MedMentions require disambiguation. This also explains why MedCAT significantly outperforms all other models, especially on the MedMentions dataset.

2.4.3 BERT models

BERT is a pre-trained English language representation model. BERT stands for Bidirectional Encoder Representations from Transformers, a machine learning technique developed by Google (Devlin et al., 2018). BERT is a state-of-the-art NLP technique because it is able to capture a deep bi-directional context of words. Transformer models are non sequential, meaning that the input can be passed into it in parallel, making them effective at memorizing long term relations (Z. Wang et al., 2019). BERT was pre-trained on BooksCorpus and English Wikipedia. The advantage of a pre-trained model is that the model has already created contextual embeddings,

Dataset:	**MedMentions**			**MedMentions (disorders only)**			**ShARe/CLEF**		
Model:	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
SemEHR	0.252	0.165	0.200	0.295	0.499	0.371	0.680	0.623	0.650
Bio-YODIE	0.316	0.143	0.197	0.445	0.366	0.402	0.700	0.607	0.650
cTAKES	0.284	0.129	0.178	0.313	0.375	0.342	0.567	0.640	0.601
MetaMap	0.305	0.465	0.368	0.358	0.460	0.403	0.755	0.540	0.630
ScispaCy*	**0.451**	0.408	0.429	0.487	0.443	0.464	0.711	0.463	0.561
CLAMP*	0.324	0.067	0.110	**0.533**	0.236	0.327	0.772	0.447	0.566
MedCAT BERT	0.386	0.475	0.426	0.459	0.513	0.485	0.788	0.678	0.729
MedCAT	0.406	**0.500**	**0.448**	0.470	**0.523**	**0.495**	**0.796**	**0.688**	**0.738**
+ Δ(MedCAT-Best)	-0.045	0.035	0.019	-0.063	0.024	0.031	0.041	0.048	0.088

TABLE 2.3: Comparison of performance among several clinical concept extraction models (Kraljevic et al., 2021). *The results for ScispaCy/CLAMP are not directly comparable to other tools as they are supervised models.

based on large corpora. As a result, the model only needs to be fine-tuned for a specific task on a relatively small labelled dataset (Delobelle, Winters, and Berendt, 2020).

Because BERT is trained on non-biomedical, non-Dutch texts, we explore other options that better fit our research. First, we discuss BioBERT and PubMedBERT, which do focus on the biomedical domain, but only on English texts. Then we will discuss some BERT models that are trained on non-biomedical Dutch texts. Unfortunately, none of the options meet both criteria: Dutch & biomedical texts. Lastly we discuss SMITH, a recent alternative to BERT.

2.4.3.1 BioBERT

BioBERT is a BERT-based pre-trained English language representation model specifically for biomedical text mining (Lee et al., 2020). BioBERT was pre-trained additionally on PubMed abstracts and PMC full-text articles (Table 2.4).

BioBERT can be used for NLP tasks such as NER, question answering and relation extraction. For this thesis, we are interested in NER and relation extraction. Here an example for both:

- BioBERT-based NER: in Si et al. (2019) multiple English pre-trained models, among which BioBERT, were applied to the 4 different clinical concept extraction tasks. One of the tasks, the 2010 i2b2b/VA challenge (Uzuner et al., 2011), consisted of identifying all problem, test and treatment related entities. BioBERT achieved an f1 score of 84.8% on the task.

- BioBERT-based relation extraction: in Alimova and Tutubalina (2020) various English BERT models were applied to two relation annotated corpora: 1) MADE (Jagannatha et al., 2019): consists of EHRs from 21 cancer patients annotated for 7 types of relations (among which adverse drug reaction). 2) n2c2 (Henry et al., 2020): consists of 505 discharge summaries and annotated for 8 types of relations (also includes adverse drug reaction). They limited the number of characters allowed to be between an entity pair to 1000 and entities allowed to be in between an entity pair to 3. BioBERT's f1 score was 91% on the MADE dataset and 75.2% on n2c2.

Corpus	Number of words	Domain
English Wikipedia	2.5B	General
BooksCorpus	0.8B	General
PubMed Abstracts	4.5B	Biomedical
PMC Full-text articles	13.5B	Biomedical

TABLE 2.4: Text corpora used to train BioBERT (Lee et al., 2020).

2.4.3.2 PubMedBERT

Gu et al. (2021) mention the common assumption that domain specific models should be further pre-trained using general-domain language models as a starting point. Gu et al. (2021) mention that no prior biomedical BERT model has been pre-trained purely on English biomedical text. An example of this is BioBERT, which uses BERT as a base and is then further trained on English biomedical domain texts (Table 2.5). To use a different approach, Gu et al. (2021) propose training a biomedical model from scratch, only on an English PubMed dataset (3.1B words/21 GB). Their model showed state-of-the-art results by outperforming BERT models (Table 2.5) on a variety of biomedical tasks.

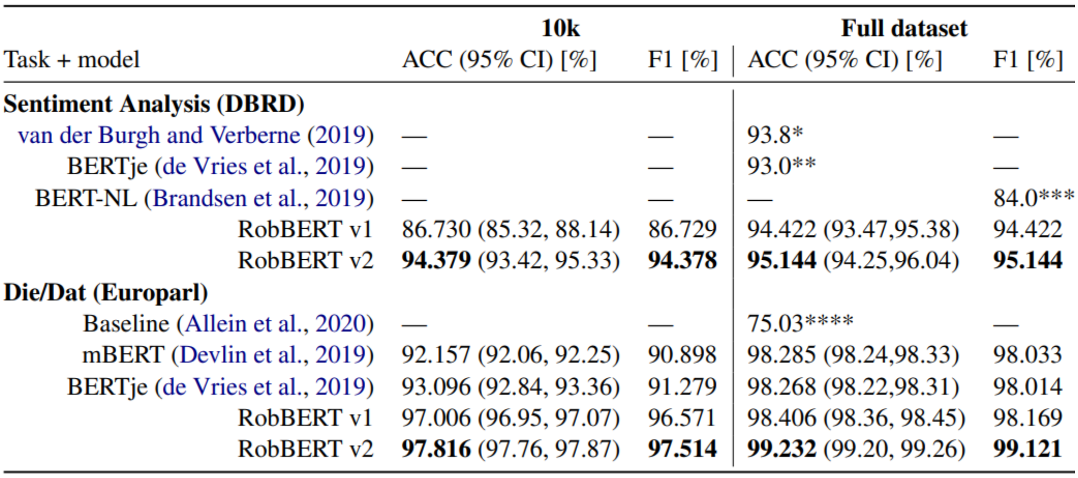
In order to use BioBERT and PubMedBERT for specific use cases, they need to be fine-tuned. Fine-tuning happens through supervised learning on a labelled dataset. This can be problematic when little or no labelled data is available, which is often the case in the medical domain. Another disadvantage of these models is that they are pre-trained on English corpora, meaning performance will significantly drop when applied to other languages. As we will be working with Dutch admission and discharge letters (Section 3.2.2), we require a model that is pre-trained on Dutch language.

	Vocabulary	Pretraining	Corpus	Text Size
BERT	Wiki + Books	-	Wiki + Books	3.3B words/ 16 GB
RoBERTa	Web crawl	-	Web crawl	160 GB
BioBERT	Wiki + Books	continual pretraining	PubMed	4.5B words
SciBERT	PMC + CS	from scratch	PMC + CS	3.2B words
ClinicalBERT	Wiki + Books	continual pretraining	MIMIC	0.5B words/ 3.7 GB
BlueBERT	Wiki + Books	continual pretraining	PubMed + MIMIC	4.5B words
PubMedBERT	PubMed	from scratch	PubMed	3.1B words/ 21 GB

TABLE 2.5: Overview of pre-training details for several BERT models (Gu et al., 2021).

2.4.3.3 Dutch BERT models

Unfortunately, there are no biomedical pre-trained models available in Dutch. There are a few Dutch options pre-trained for the general domain: BERTje (Vries et al., 2019), BERT-NL (Brandsen et al., 2019), RobBERT (Delobelle, Winters, and Berendt, 2020) and belabBERT (Wouts et al., 2021). There are also multilingual models such as multilingual-BERT (Devlin et al., 2018) and XLM-R (Conneau et al., 2019). Multilingual pre-trained models are trained on large corpora containing many different languages: multilingual-BERT is trained on corpora containing 104 languages (Devlin et al., 2018). Although a model like multilingual-BERT performs well across languages on a variety of tasks, models trained specifically on a single language seem to perform better (Pires, Schlinger, and Garrette, 2019; Brandsen et al., 2019; Martin et al., 2019). Delobelle, Winters, and Berendt (2020) compared several Dutch models as shown in Table 2.6, RobBERT v2 performed best among these models. BelabBERT was not included in this comparison, as it was released only quite recently (June 2021). However, in the paper by Wouts et al. (2021), belabBERT is compared to RobBERT v2 (Table 2.7) and the authors conclude that their results show that "belabBERT outperformed the current best text classification network for Dutch, RobBERT". We will dive further into these two Dutch models below.



	10k		**Full dataset**	
Task + model	ACC (95% CI) [%]	F1 [%]	ACC (95% CI) [%]	F1 [%]
Sentiment Analysis (DBRD)				
van der Burgh and Verberne (2019)	—	—	93.8*	—
BERTje (de Vries et al., 2019)	—	—	93.0**	—
BERT-NL (Brandsen et al., 2019)	—	—	—	84.0***
RobBERT v1	86.730 (85.32, 88.14)	86.729	94.422 (93.47,95.38)	94.422
RobBERT v2	**94.379** (93.42, 95.33)	**94.378**	**95.144** (94.25,96.04)	**95.144**
Die/Dat (Europarl)				
Baseline (Allein et al., 2020)	—	—	75.03****	—
mBERT (Devlin et al., 2019)	92.157 (92.06, 92.25)	90.898	98.285 (98.24,98.33)	98.033
BERTje (de Vries et al., 2019)	93.096 (92.84, 93.36)	91.279	98.268 (98.22,98.31)	98.014
RobBERT v1	97.006 (96.95, 97.07)	96.571	98.406 (98.36, 98.45)	98.169
RobBERT v2	**97.816** (97.76, 97.87)	**97.514**	**99.232** (99.20, 99.26)	**99.121**

TABLE 2.6: Performance of several Dutch pre-trained transformers (Delobelle, Winters, and Berendt, 2020). Scores annotated with * are reported as in their original paper.

2.4.3.4 RobBERT v2

RobBERT was pre-trained on Dutch text using the RoBERTa training pipeline, which in turn is an improved version of BERT (Delobelle, Winters, and Berendt, 2020; Y. Liu et al., 2019; Devlin et al., 2018). In the RobBERT v2 version, the original pipeline of RoBERTa was adapted using Dutch versions for both the corpus and the tokenizer. The model was pre-trained on the Dutch OSCAR corpus which contains 6.6B words (Ortiz Suárez, Romary, and Sagot, 2020). What is also interesting about RobBERT is that it can perform well on a relatively small dataset compared to other Dutch pre-trained models (Figure 2.6).

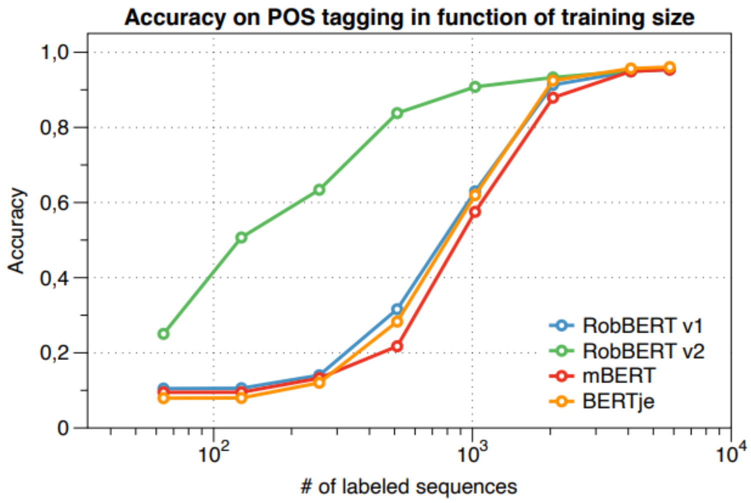


FIGURE 2.6: POS tagging accuracy of several Dutch pre-trained transformers relative to the amount of training data (Delobelle, Winters, and Berendt, 2020).

2.4.3.5 BelabBERT

Similar to RobBERT, belabBERT is based on the RoBERTa architecture (Y. Liu et al., 2019). BelabBERT was pre-trained on the Dutch OSCAR corpus, on >32GB web crawled texts (Ortiz Suárez, Romary, and Sagot, 2020). Whereas RobBERT uses the shuffled and pre-cleaned version of OSCAR, belabBERT uses the non-shuffled version in which the sentence order is preserved. This choice was made to enable belabBERT to learn long-range dependencies in texts. Wouts et al. (2021) evaluated their model on a sentiment analysis task and compared their model to other Dutch BERT models, as can be seen in Table 2.7. They found that their model outperforms RobBERT on several tasks.

Model name	**Pre-train corpus**	**Tokenizer type**	**Acc Sentiment analysis**
belabBERT	Common Crawl Dutch (non-shuffled)	BytePairEncoding	95.92%
RobBERT	Common Crawl Dutch (shuffled)	BytePairEncoding	94.42%
BERTje	Mixed (Books, Wikipedia, etc)	Wordpiece	93.00%

TABLE 2.7: The top 3 performing Dutch BERT models based on their sentiment analysis accuracy (Wouts et al., 2021).

2.4.4 SMITH

Siamese multi-depth transformer-based hierarchical encoder for long-Form document matching, abbreviated as SMITH, is a recently developed model by L. Yang et al. (2020) that contains adaptations to self-attention models to allow for longer text input. SMITH was pre-trained on a random English Wikipedia collection that consists of ±715k documents (±956M words). The max sequence input length of

BERT is limited to 512, for SMITH this is 2048. Whereas BERT is trained by predicting masked words within sentences, SMITH is additionally trained by predicting what the next sentences will be within the document context. BERT's focus is at word-level within sentences, SMITH's focus is at sentence-level within documents.

2.4.5 Relation Extraction with entity information

In our use case, we want to input text along with the entity annotations into a relation extraction model to predict whether a relation exists for the given pair of entities: {drug, event}. Here, relation extraction relies not just on the information of the input text, but also on the two target entities. Before BERT was released in 2018, most relation extraction models were based on convolutional or recurrent neural networks S. Wu and Y. He (2019). However, since the release of BERT, many state-of-the-art relation extraction models are based on BERT or on one of its relatives. Earlier in Section 2.3, we discussed the SemEval-2010 Task 8 dataset, often used as a benchmark for relation extraction models. Paperswithcode (2021) keeps track of state-of-the-art performances on this dataset and among the top 20, many make use of BERT.

We will take an in-depth look at two models that are BERT based and incorporate entity information, as we will be using these for this project. The first one is R-BERT by S. Wu and Y. He (2019), which is among the top performing models on the task of relation extraction on SemEval-2010 Task 8. The second model is provided by Sboev, Selivanov, et al. (2022) and does not report scores on the SemEval-2010 Task 8. This one is particularly interesting because it has only been released recently (April, 2021) and was developed to identify adverse drug reactions. The approaches of these two models are discussed below.

2.4.5.1 R-BERT

In R-BERT, the input text is preceded by a '[CLS]' token, the span of the first entity *e1* is marked by '$' and the span of the second entity *e2* is marked by '#' (S. Wu and Y. He, 2019). By marking the entities, we can use the output vector to implement information on the entities in the model. The architecture of R-BERT is depicted in Figure 2.7. The final hidden state vectors from the BERT model for *e1* (H_i - H_j) and *e2* (H_k - H_m) are averaged and a tanh activation function is applied. A fully connected layer is added to both averaged output layers. The final hidden state for the '[CLS]' token (representing the whole sequence) is also followed by a tanh activation function and a fully connected layer. These 3 output layers (derived from *e1*, *e2* and '[CLS]') are concatenated and then a fully connected layer and a softmax layer are added.

The f1 score of R-BERT on the SemEval-2010 benchmark dataset is 89.25%. Removing the special separation tokens '$' and '#', as well as discarding the hidden output vectors from the two entities drops the f1 performance to 81.09% (S. Wu and Y. He, 2019).

2.4.5.2 RE on Russian user reviews

In the paper by Sboev, Sboeva, et al. (2021) they collected 2800 reviews from a forum dedicated to consumer reviews on medications. The corpus was annotated for 4 types of entities: 1) drugs 2) diseases and symptoms 3) ADRs 4) note: deviating cases such as discontinuous annotations, or annotations with multiple possible entities. This resulted in 33005 medication entities, 17403 for diseases and symptoms,

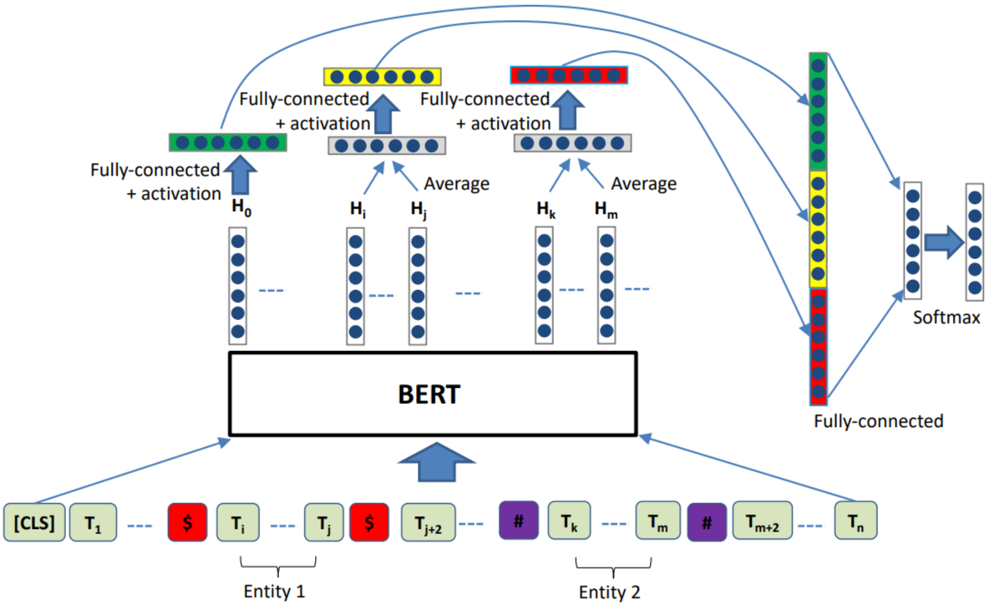


FIGURE 2.7: R-BERT model architecture (S. Wu and Y. He, 2019).

1778 for ADRs and 4490 notes. Inter-annotator agreement scored between 61-71%, depending on how strictly it was judged.

In another paper by Sboev, Selivanov, et al. (2022) they used the above dataset together with the XLM-RoBERTa-base and XLM-RoBERTa-large model (further trained on Russian medical texts by Sboev, Sboeva, et al. (2021)) for relation extraction. Because the model was initially developed for Russian user reviews, we will refer to the model as RUS. The main purpose of their architecture is to preprocess the text in such a way that more info is captured in the vector representation that is used for classification. We will use an example sentence to illustrate the four different preprocessing strategies that were considered: "A hyponatremia was observed, due to hydrochlorothiazide".

1. The whole input text, with the entities marked with special start and end tokens: "[CLS] A <e1> hyponatremia </e1> was observed, due to <e2> hydrochlorothiazide </e2>"

2. Only the target entities: "[CLS] hyponatremia [SEP] hydrochlorothiazide"

3. The target entities and the text in between: "[CLS] hyponatremia [SEP] hydrochlorothiazide [TXTSEP] was observed, due to"

4. The whole input text and the target entities: "[CLS] hyponatremia [SEP] hydrochlorothiazide [TXTSEP] A <e1> hyponatremia </e1> was observed, due to <e2> hydrochlorothiazide </e2>"

We will use the last strategy that they propose, because it was most effective. Here they use the whole input text and the target entities. This was then fed into the XLM-RoBERTa model as shown in Figure 2.8. When looking at the results only for the ADR-Drugname relations, the RUS model achieved an f1 score of 91.9%.

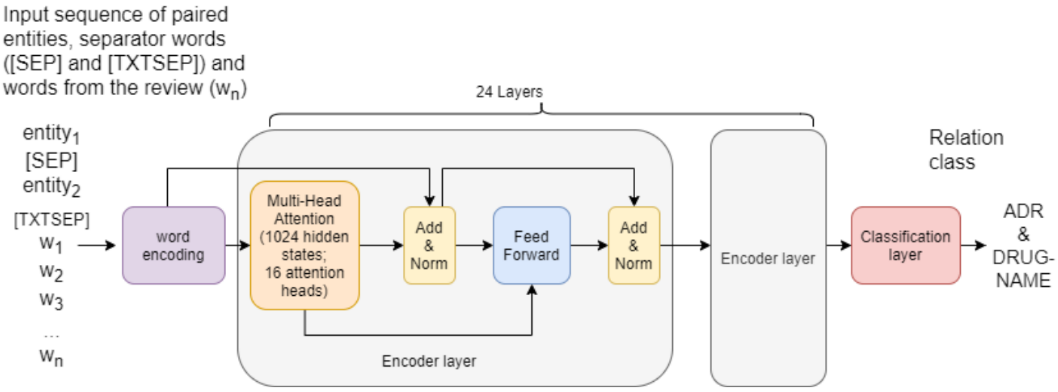


FIGURE 2.8: RUS model architecture (Sboev, Selivanov, et al., 2022).

2.5 NLP in Adverse Drug Reactions (ADRs)

Research in pharmacovigilance, also known as drug safety, is based on a variety of data sources. Some of these data sources like biomedical literature, spontaneous reports and clinical trials are inherently structured and therefore directly viable for research towards ADRs. With the introduction of NLP, we can now also transform data from sources such as search logs, social media and electronic health records (EHRs) into accessible structured data formats, viable for research (Harpaz et al., 2014). For example, social media can be useful for detecting unknown ADRs that are not listed on the drug label (Yates, Goharian, and Frieder, 2015). Our focus will be on EHRs as a data source. In this section we will go over some similar work in this field of research.

EHRs are a promising source for research towards drug safety (Coloma et al., 2011; Luo et al., 2017). Sessa et al. (2020) conducted a systemic review on 77 articles about AI in pharmaco-epidemiology. They found that random forests (RFs), artificial neural networks (ANNs) and support vector machines (SVMs) respectively were the most used techniques. About one fifth of these studies focused on the occurrence/severity of adverse drug reactions. Through the monitoring of EHRs, we can discover new ADRs and discover these sooner, it also enables us to identify patients at risk.

Perera et al. (2013) and Harpaz et al. (2014) mention linguistic and semantic components of sentences that NLP models need to be able to detect in order to understand electronic medical records: 1) Negations, "patient X has no signs of fever". 2) Conditional statements, "if the pain worsens, use naproxen". 3) Uncertainty, "not sure if it is caused by renal failure". 4) Temporal properties, "last week, patient X was suffering from a dry cough". 5) Experiencer, whether the statement is about the patient or about others (family, nurse etc.). Perera et al. (2013) explain how simple cases of these statements can be solved with current NLP techniques, but that some more sophisticated statements require additional attention. They propose to use a knowledge base to resolve some of these more complex statements, which can lead to better analysis of clinical content.

There are many studies that use NLP to detect ADEs and ADRs. L. Chen et al. (2020) used a deep learning, attention-based bidirectional long short-term memory network (biLSTM), to detect medication related information in clinical notes. Their model achieved an f1 score of 81% on identifying relations between drugs and corresponding ADRs. Another deep learning model designed for ADR relation extraction is the one by Sboev, Selivanov, et al. (2022) described in Section 2.4.5.2.

In the n2c2 shared task, teams needed to extract ADEs in 3 steps: concept extraction, relation classification, and end-to-end systems (Henry et al., 2020). The best performing systems had a f1 score of 94.2%, 96.3% and 89.1% for the 3 steps respectively. However, the f1 score of just concept extraction relating to ADEs was significantly lower, with an f1 score between 25.0% and 60.0% for the top 10 best performing teams. The f1 for the relation classification step for ADE-drug interactions scored between 40.0% and 95.0% for the top 10 teams. This shows that extracting concepts and relations that are related to ADEs is one of the most challenging tasks in the field.

Haerian et al. (2012) used a different approach. They developed a tool that could detect whether an adverse event was due to disease etiology rather than drug etiology. It could identify these cases with a high sensitivity and specificity (respectively 93.8% and 91.8%). When filtering EHRs on adverse events such as rhabdomyolysis, this tool can then be applied to filter out most of the cases that are not drug-induced. The documents left over are cases of drug induced adverse events, being ADRs.

Chapman et al. (2019) developed a NLP model that first extracts symptoms and drugs from clinical notes (NER) and subsequently labels the relations between them. The NER component had an f1 score of 80.9%, the relation extraction (RE) component an f1 score of 88.1%.

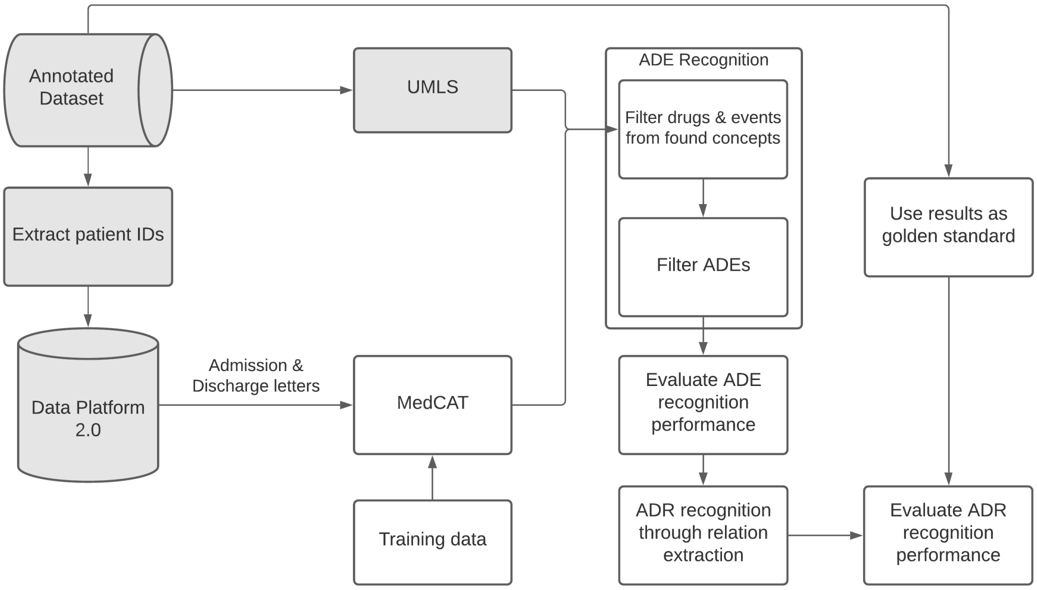
Tang et al. (2019) used a rule-based approach that reached $\geq$90% precision and recall for the entity recognition of drugs and events. They also performed rule-based RE to identify adverse drug events, the relation extraction achieved $\geq$75% precision and $\geq$60% recall.

As we described in this section, many studies have been done with NLP to detect ADRs and many of the mentioned studies used clinical data. A typical approach is implementation of a NER component, followed by an RE component. In Section 3.5 we describe the methodology that we use for our NER component. For RE, we see a variety of techniques used in prior research, ranging from simple rule-based approaches (Tang et al., 2019) to complex deep learning approach such as an attention-based biLSTM (L. Chen et al., 2020). In Section 3.6 we explain our strategy to create an RE component. Additionally, we also identified some of the challenges that come with NLP of clinical notes in this section. We also come across these challenges in our research, which will be addressed in the error analysis (Section 4.2.3, Section 4.3.2 and Section 4.3.4) and in the discussion (Chapter 5).

Chapter 3

Methodology

In this chapter, we will go over the methodology used in this research. In figure 3.1 we present a flowchart of our approach towards a solution. First, we have a dataset that serves as our golden standard and has been annotated for the occurrence of adverse drug events (ADEs) and adverse drug reactions (ADRs), this is described in Section 3.2. This dataset consists of admission and discharge letters. The annotations in our golden standard are only done at document-level, therefore we need to add annotations at word-level, for this we use MedCAT (sub-question 1a in Section 1.3). We need to compile a concept database for Dutch terms (Section 3.1) in order to use MedCAT to annotate Dutch text. From the resulting annotations we keep only the concepts annotated for in our golden standard: this choice is further discussed in Section 3.2.1. These annotations compromise a subset of pharmaceutical drugs (also referred to as **triggers** in this thesis) and events. These marked entities are then used to recognize ADEs (sub-question 1b). Next, we use two models based on belabBERT to perform relation extraction between the annotated triggers and events (sub-questions 1c and 1d).



Annotated Dataset
UMLS
ADE Recognition
Filter drugs & events from found concepts
Filter ADEs
Extract patient IDs
Use results as golden standard
Data Platform 2.0
Admission & Discharge letters
MedCAT
Evaluate ADE recognition performance
Training data
ADR recognition through relation extraction
Evaluate ADR recognition performance

FIGURE 3.1: A flowchart with all the components of the project. The grey boxes contain steps that are not part of this project: *Annotated dataset* originates from previous research and *UMLS* is an existing collection of biomedical vocabularies (Section 2.4.1).

3.1 Concept Database

To perform clinical concept extraction with MedCAT, we need a concept database that has terms linked to concepts (Section 2.1.3). In our research, we have MedCAT use a subset of the UMLS concept database, using the Dutch versions of several biomedical vocabularies: MeSH, MedDRA, ICPC (ICPCDUT, ICPC2ICD10DUT and ICPC2EDUT), ICD10DUT and LOINC (https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html). An overview is provided in table Section 3.1.

Vocabulary	**Version**	**Concepts**
MeSH	2005	20615
MedDRA	2020	56914
ICPCDUT	1999	722
ICPC2ICD10DUT	2005	35466
ICPC2EDUT	2005	685
ICD10DUT	2005	10697
LOINC	2020	53938
	Total:	179037

TABLE 3.1: An overview of the Dutch UMLS vocabularies used including their versions and amount of concepts.

With ≈179k concepts, the selected Dutch UMLS vocabularies are relatively small. The UMLS 2021AA release contains over 4.4M concepts, of which the majority is in English (National Library of Medicine, 2020a). To solve this, we added the Dutch version of SNOMED, which contains ≈219k concepts (SNOMED, 2021). UMLS does support SNOMED US and not the Dutch version, therefore we had to perform some extra steps to add this:

- The Dutch SNOMED and SNOMED US share the same concept identifier and SNOMED US is mapped to UMLS concept unique identifiers (CUIs). This allows us to create a mapping from the Dutch SNOMED to UMLS. Sometimes Dutch SNOMED contains only one term for a concept, while in SNOMED US, the same concept can have several terms and it is possible that each of these terms map to different UMLS concepts. Then it is problematic to decide a mapping for the Dutch SNOMED term to a UMLS concept. Mapping to both will cause ambiguity and choosing to which one to map manually is very labor intensive and may require expert knowledge. Therefore we decided to exclude these from the merge.

- A lot of SNOMED terms already exist in UMLS and are therefore also skipped during merging.

After these steps, ≈123k (56%) CUIs were removed from the Dutch SNOMED, meaning the other ≈96k CUIs were added from Dutch SNOMED to our concept database. The final concept database consists of ≈635k terms mapped to ≈275k UMLS CUIs.

3.2 Data: Relation Extraction

For this research we will use the same data as used by Noorda et al. (2022). The dataset consists of admission and discharge letters from geriatric patients. The trigger and event combinations in these letters that form ADEs were manually identified. Next, each ADE was assessed for being an ADR or not. This results in a dataset with admission and discharge letters annotated for the presence of ADEs and ADRs.

More specifically, they collected data on a random subset of patients aged ≥70 with polypharmacy (≥5 chronic prescription drugs), admitted through the emergency department to the geriatric department between 01-01-2011 and 01-08-2017 at the UMC Utrecht. This population choice has consequences for the generalisation of the results. First of all, the presence of polypharmacy makes it difficult to find a causal relation between triggers and events. On the other hand, presence of polypharmacy provides us with a dataset that contains a higher density of events and triggers as compared to other populations. Secondly, in this specific population age range (≥70) the distribution of triggers and events among different categories (as seen in Table 3.2) may be skewed compared to populations of different age. This can cause our model to be less sensitive to categories that are more prevalent in different age groups. Lastly, these patients are admitted to the geriatric department, meaning that care is provided by geriatricians. ADRs play an important role in geriatrics and cause geriatricians to pay more attention to events and triggers while writing clinical notes. As an example, surgeons on the other hand, focus on different aspects of health care in their writings. Therefore, performance of our model may be different across different branches of specialized care.

The selected inclusion criteria resulted in a collection of 589 patients. 211 patients were excluded because of incomplete records (missing admission or discharge letter) and 33 patients were excluded because the admission was not the patient's first admission within the study period. From those patients, only those with at least one drug-event combination in the admission letter were selected. Drug-event combinations are those as described by the ADR trigger tool as defined in the Dutch national geriatric guideline "Polypharmacy in the elderly" (Federatie Medisch Specialisten (FMS), 2020). This tool provides an overview of the problems that most often lead to ADR-related hospital admissions in older patients. In the tool, medication related events that occur often are linked to the pharmaceutical drugs that cause them. The trigger tool is compiled from the 10 most prevalent medication related problems according to HARM-, IPCI- and QUADRAT studies (Warlé-van Herwaarden et al., 2014; Warlé-van Herwaarden et al., 2015; Hooft et al., 2008), complemented with information from several other studies as described in Federatie Medisch Specialisten (FMS) (2020). The tool is not exhaustive and does not cover the set of all possible triggers and events. This also means that the dataset does not contain annotations on triggers and events that are not listed in this tool. However, as our model will learn to recognize how relations between triggers and corresponding events occur in these documents, it might also be able to generalize well on triggers and events outside the scope of the tool. In Section 5.3, we discuss generalization for relation extraction.

Noorda et al. (2022) slightly modified this ADR trigger tool, called "explicated ADR trigger tool", as can be seen in Table 3.2. They made three types of changes: 1) Events that represent different symptoms that may link to a different drug class are split up. 2) Events that are used interchangeably and difficult to separate are merged. 3) Drug categories are specified on ATC-classification.

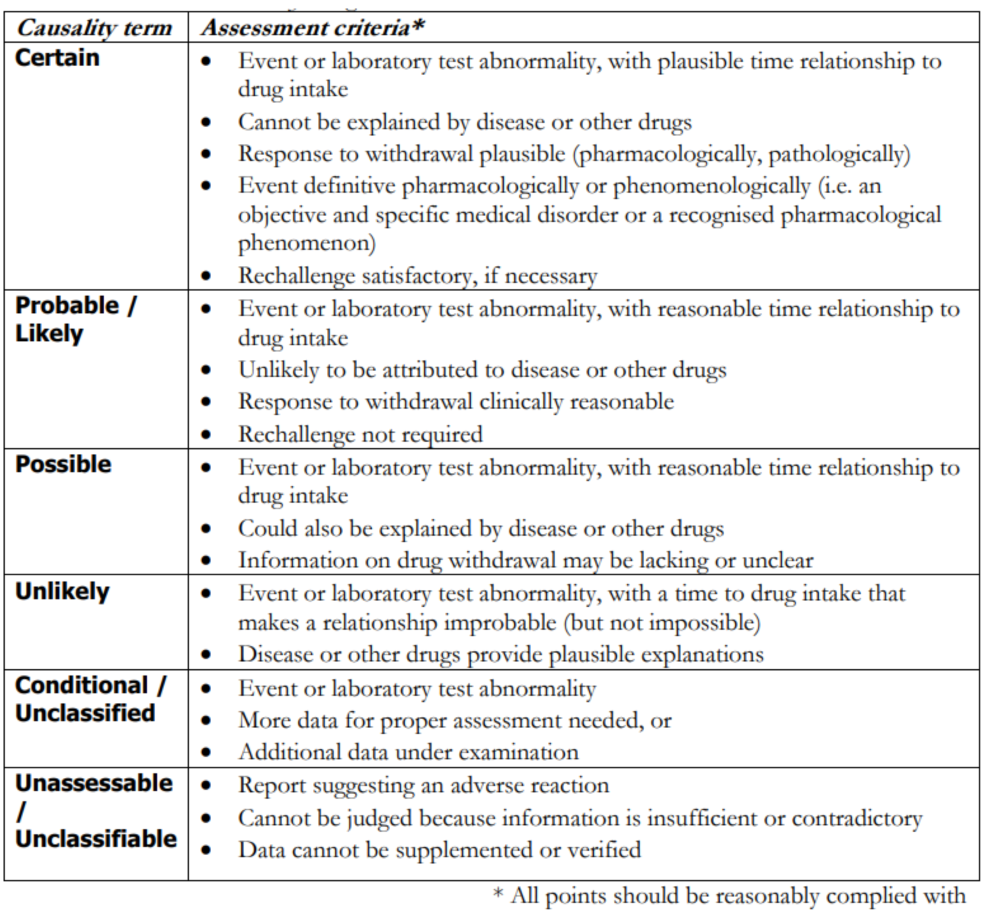
Explicated ADR Trigger Tool	
Events	**Drug class**
1. Fracture	A. Systemic corticosteroids
2. Fall(*) / collaps / (orthostatic) hypotension / dizziness / syncope	A. Antihypertensive agents: ACE-inhibitors, ATII-antagonists, calcium antagonists, beta blockers, thiazide diuretics, loop diuretics, potassium sparing diuretics, alpha-1-blockers, long acting nitrates. Antiarrhythmic agents: digoxin, class I, II and III antiarrhythmic agents B. Psychotropic drugs: benzodiazepines, antipsychotics, antidepressants (i.e. SSRI, TCA and miscellaneous: duloxetine, venlafaxine and mirtazapine)
3.1 Gastro-intestinal bleeding 3.2 Intracranial bleeding 3.3 Other bleedings	A. Vitamin K antagonists, DOACs, heparins, other anticoagulants B. Thrombocyte aggregation inhibitors C. NSAIDs
3.4 Supratherapeutic INR	A. Vitamin K antagonists
4.1 Hyponatraemia	A. Thiazide diuretics, loop diuretics, potassium sparing diuretics B. ACE-inhibitors, ATII-antagonists C. Antidepressants (i.e. SSRI, TCA and miscellaneous: duloxetine, venlafaxine and mirtazapine)
4.2 Hypokalaemia	A. Thiazide diuretics, loop diuretics
4.3 Hyperkalaemia	A. Potassium sparing diuretics B. ACE-inhibitors, ATII-antagonists
5. Renal insufficiency and/or dehydration (*)	A. ACE-inhibitors, ATII-antagonists B. NSAIDs C. Thiazide diuretics, loop diuretics, potassium sparing diuretics
6.1 Hypoglycaemia	A. Oral antidiabetics, insulin and analogues
6.2 Hyperglycaemia	B. Systemic corticosteroids
7. Acute heart failure	A. NSAIDs
8. Constipation / ileus (based on constipation)	A. Opioids B. Calcium channel blockers
9. Vomiting / diarrhea	A. Antibiotics
10. Delirium / confusion / drowsiness	Drugs with anticholinergic and sedative properties, digoxin, anti-Parkinson drugs

TABLE 3.2: The explicated version of the ADR trigger tool (Noorda et al., 2022)

Application of the explicated ADR trigger tool resulted in identification of 941 drug-event combinations in 253 patients. These drug-event combinations were assessed using the WHO-UMC causality assessment system (World Health Organization, 2009). In this assessment each ADE is assigned a score of how likely it is to be an ADR: conditional, unclassifiable, unlikely, possible, probable and certain (Table 3.3). Noorda et al. (2022) considered the last three categories as ADRs: possible, probable and certain. This resulted in the identification of 393 (41.8%) ADRs among 941 ADEs. 88.9% (837) of the 941 ADEs, occurred in just four of the event categories (see full category specification in Table 3.2): fall (32.4%), delirium (24.0%), electrolyte disturbances (hyponatraemia, hypokalaemia and hyperkalaemia, in total 16.4%) and renal insufficiency and/or dehydration (16.2%). These 839 ADEs in the top 4 categories, included 313 (79.6%) of all 393 ADRs. For each event category that we analyse, we need to compile a specific list of all medications that fall in the categories mentioned in Table 3.2. This list is compiled by two clinical pharmacologist and a geriatrician. Because compiling this list is labour intensive and most ADRs occur in the top 4 categories (79.6%), we focus only on those categories. Imitating the results of the WHO-UMC causality assessment with a relation extraction model is difficult, because it requires a deep understanding of all the drugs, events and interactions involved in the patient's admission, as can be seen in Table 3.3. However, Noorda et al. (2022) also assessed ADR recognition by usual care. Recognition by usual care was defined as "an explicit documented drug-event combination by the treating physician (i.e. geriatric resident, supervised by a geriatrician) in the admission and/or discharge letter, implying that the drug-event combination was considered an ADR by usual care". Additionally, when a drug was withdrawn or its dose adjusted, in combination with explicitly mentioning a corresponding event, this was also considered recognition by usual care. These labelled ADRs seem a more realistic approach to develop an AI model capable of capturing these relations. Therefore we also train and test a model using only the usual care instances. The top 4 categories contain 244 usual care instances that correctly identify ADRs, however these are annotated at document-level so we cannot deduce to which specific sentence(s) these annotations correspond. Therefore we generate our own dataset of usual care instances as described in Section 3.6.1. An important note is that not all usual care instances are ADRs (68.2% is an ADR) and that not all ADRs are recognized by usual care (16.5% is missed).

3.2.1 Relevant Concepts

In the above described dataset, each letter is annotated for whether and which drug-event combination occurs. However, the dataset does not specify where in a letter the trigger and the event occur (also, these do not necessarily have to be in close proximity to each other). For that purpose, MedCAT will serve as a tool to disambiguate and label entities with the proper concept unique identifier (CUI), as described in Section 2.4.2. As our dataset focuses on specific drug-event combinations, we will only be looking at a subset of all the CUIs recognized by MedCAT. This means that we will not be recognizing every drug-event combination, but only those specified by the ADR trigger tool. This means we use a task-specific approach, it allows us to match exactly those ADEs that we are looking for (present in the Table 3.2), but not more. The advantage is that this causes less false positives (Section 4.2.3) and it makes the manual concept annotation process less labour intensive (Section 3.3). The downside, is that our methodology is only evaluated for a subset of ADEs, so we do not know how well it performs when generalizing to other



Causality term	*Assessment criteria**
Certain	• Event or laboratory test abnormality, with plausible time relationship to drug intake • Cannot be explained by disease or other drugs • Response to withdrawal plausible (pharmacologically, pathologically) • Event definitive pharmacologically or phenomenologically (i.e. an objective and specific medical disorder or a recognised pharmacological phenomenon) • Rechallenge satisfactory, if necessary
Probable / Likely	• Event or laboratory test abnormality, with reasonable time relationship to drug intake • Unlikely to be attributed to disease or other drugs • Response to withdrawal clinically reasonable • Rechallenge not required
Possible	• Event or laboratory test abnormality, with reasonable time relationship to drug intake • Could also be explained by disease or other drugs • Information on drug withdrawal may be lacking or unclear
Unlikely	• Event or laboratory test abnormality, with a time to drug intake that makes a relationship improbable (but not impossible) • Disease or other drugs provide plausible explanations
Conditional / Unclassified	• Event or laboratory test abnormality • More data for proper assessment needed, or • Additional data under examination
Unassessable / Unclassifiable	• Report suggesting an adverse reaction • Cannot be judged because information is insufficient or contradictory • Data cannot be supplemented or verified

* All points should be reasonably complied with

TABLE 3.3: WHO-UMC causality categories (World Health Organization, 2009)

ADE categories. In Section 5.3, we also discuss how this impacts generalization for relation extraction.

The next step is to translate the ADR trigger tool as proposed by Noorda et al. (2022) into a list of CUIs.

3.2.1.1 Events to CUIs

First for the events, we used the UMLS browser to find the CUI corresponding to each of the event terms in the top 4 categories in the explicated ADR trigger tool from Noorda et al. (2022). These are listed in table A.1.

3.2.1.2 Triggers to CUIs

Then for the triggers a clinical pharmacologist translated these from the explicated ADR trigger tool into anatomical therapeutic chemical (ATC) codes (World Health Organization, 2021). These were then reviewed by a geriatrician. This is necessary because the drugs listed in Table 3.2 are higher order concepts and we need to translate these to pharmaceutical drug names, the lower order concepts. For example, we need to translate NSAIDs into generic drug names like naproxen, ibuprofen, diclofenac, but also into the branded names such as advil, nurofen and aleve. ATC codes follow a hierarchical classification system and can consist of up to 7 characters (Figure 3.2). When more characters are specified, this means a more specific category and lower in the hierarchical structure. For example: 'C' stands for 'cardiovascular system', 'C01' for 'cardiac therapy' (a child of cardiovascular system), 'C01A' for 'cardiac glycosides'. Using UMLS, we translate the provided ATC codes (and all their descendants) into a list of CUIs. The resulting list of CUIs correspond only to the generic names of the drugs.

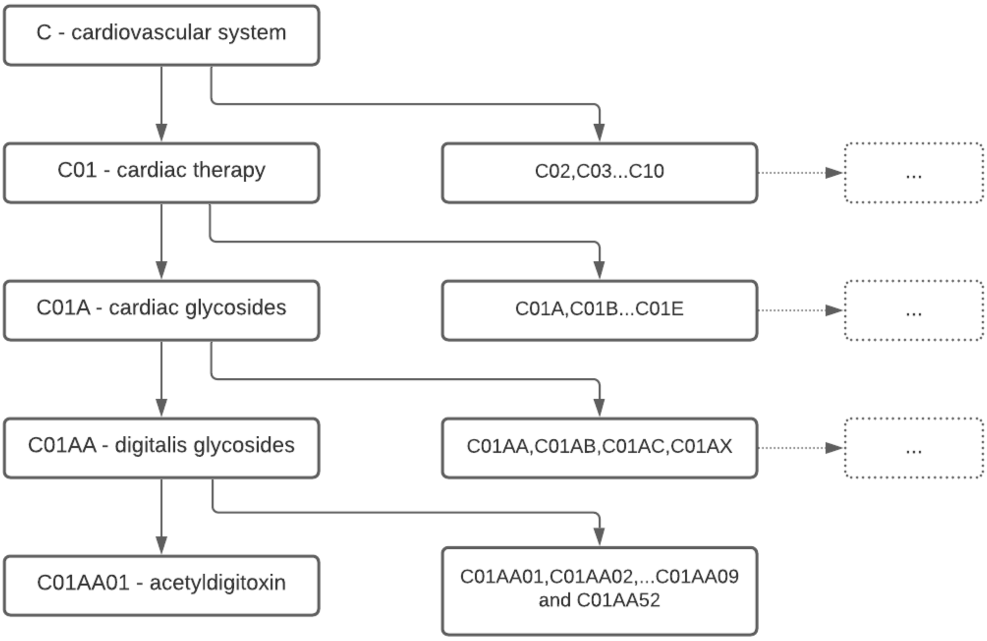

C - cardiovascular system
C01 - cardiac therapy
C02,C03...C10
...
C01A - cardiac glycosides
C01A,C01B...C01E
...
C01AA - digitalis glycosides
C01AA,C01AB,C01AC,C01AX
...
C01AA01 - acetyldigitoxin
C01AA01,C01AA02,...C01AA09 and C01AA52

FIGURE 3.2: Showing the left most branch for every level in the hierarchical tree structure of WHO's ATC system.

In clinical notes, doctors also use brand names of drugs. To solve this, we use the relation 'tradename_of' as specified in the RXNORM vocabulary which is also linked to UMLS CUIs (`https://www.nlm.nih.gov/research/umls/rxnorm/index.html`). This relational property links generic names of medication to the branded names. We use this to expand our list of CUIs with the CUIs of the branded names. Unfortunately, this will not result in an exhaustive list of all possible mentions of drugs. For example, "paracetamol" is often abbreviated as "pcm" which is not listed as a term for the concept. Another example of variation in drug spelling is "Prozac", which can also be written as "Prozak" or their lowercase versions: "prozac" and "prozak". We will discuss how MedCAT potentially solves such cases in Section 3.5. After this step we have 1348 CUIs for drug names (both generic and brand names).

3.2.2 Admission and Discharge Letters

The dataset used in Noorda et al. (2022) contains the admission and discharge letters of 345 patients. Noorda et al. (2022) had direct access to the letters present in the EHRs, this means they could use the HiX interface to access each patient individually and obtain the correct documents. In contrast, we had to obtain the letters directly from the database using specific queries. Because the letters were spread across the database, it was very hard to retrieve all letters and replicate the dataset. Due to these limitations we were only able to achieve the letters for 140 patients. These letters have a mean length of 874 tokens and show great variety in token length between different letters as can be seen in Table 3.4.

Most letters follow a similar structure. As can be seen in the examples below, the use of language is summarily and the letters contain a high density of medical concepts. Examples of sections that are managed in most letters are:

- Reason for admission: short mention of one or more reasons for admission structured as follows "1) Stomach ache 2) ..."
- Patient history: a list of all medical events listed like "1988 Hysterectomy, 2000 ..."
- Medication: a list of all current medication: "acetylsalicylic acid 80 mg 1dd1, metroprolol 50 mg 1dd1, ..."
- Allergies
- Anamnesis: contains a summary of the story told by the patient. When patients bring someone with them there can also be a hetero-anamnesis containing their story.
- Examinations: several types of examinations depending on the medical problem: radiological, neurological and physical examination. An example of physical examination if the medical problem is stomach related: "Abdomen: swollen belly, active peristaltis, ..."
- Laboratory research: very dense summation of laboratory values "Na 136 K 4,1 ureum 7,6 hb 6,5..."
- Discussion/Course of events: an interpretation of all the information by the clinician. Includes the course of events, what actions were taken and whether these were useful.

For each patient, the admission letter was used to determine the presence of ADEs. Whenever a drug and event that are related according to the explicated trigger tool (meaning that they are in the same row in Table 3.2) appeared together in an admission letter, it was marked as an ADE. This ADE was further specified in trigger and event. The event was annotated as one of the following: fall, delirium, dehydration or electrolyte disturbance (hyponatraemia, hypokalaemia or hyperkalaemia). The annotated trigger was provided in ATC format. A single letter can have multiple ADEs, for each ADE it is provided how likely it is to be an ADR (as explained in Section 3.2) and whether it was explicitly mentioned to be an ADR in text (usual care instances). Notable was that usual care instances are often located in the discussion and/or conclusion sections of the letters and the trigger and event mentions appear close together. In contrast, non-usual care related ADEs and ADRs can have their trigger and event components spread all throughout the letters.

3.3 Manual Annotation

In the dataset from Noorda et al. (2022), annotations for triggers and events are provided at document level. This enables us to test performance of MedCAT at document level. Because we are also interested in MedCAT's performance at concept level, we manually annotated the admission and discharge data for the relevant trigger and event concepts. In total, we annotated 280 documents for 140 patients

		Amount:
Letters	**Mean length (Characters)**	6407 (±3392)
	Mean length (Tokens)	874 (±471)
ADEs (337)	**ADRs**	129
	Fall	130
	Delirium	78
	Electrolyte Disturbances	71
	Renal Insufficiency	58

TABLE 3.4: Descriptive statistics on the admission and discharge letters. "±" denotes the standard deviation.

(one admission and one discharge document for each patient). Our approach is as follows:

1. We used our custom concept database as described in Section 3.2.1, together with the patient data, as input for MedCATTrainer (Searle et al., 2019). MedCATTrainer is an annotation tool that provides a web interface to easily confirm, alter, add or delete MedCAT annotations.

2. Run MedCATTrainer to annotate all the documents (for which it uses MedCAT). This saves time as many concepts are now already pre-annotated and only require review.

3. Manually go over all the 280 documents and annotate for the relevant concepts. For each document, we followed these guidelines:

   (a) Assume a patient is annotated by Noorda et al. (2022) to only have a single ADE, for example: "fall" as event and "furosemide" as trigger. Then we only annotate for all instances of "falling" (all concepts corresponding to trigger tool event category "2" in Table A.1) and all occurrences of "furosemide" (so also for its brand names, such as "Lasix") in the documents belonging to that patient.

   (b) We go over all MedCAT-provided annotations and review these, marking them as either correct or incorrect, or altering the assigned CUI.

   (c) Next, we read through the document for CUIs missed by MedCAT and add annotations for these.

   (d) When the document is finished, we submit it for training. This way, MedCATTrainer will learn from our confirmed, corrected, altered and added annotations. This means that annotating gets progressively easier when going through the documents.

Annotation was performed by a single annotator, so no inter-annotator agreement could be calculated. The annotations of events involved only a small set (14 CUIs) of concepts. Therefore it was not difficult to determine which concept each event should be linked to. For the triggers, the set of concepts was more extensive (1348 CUIs). However, drug names are unambiguous by nature and are very easy to map to the correct concept. The "Farmacotherapeutisch Kompas" was used to classify drugs in case of doubt (Zorginstituut Nederland, 2021). In total, we ended up with 3301 annotations.

One of the shortcomings of MedCAT that we point out in Section 4.1.3, is that MedCAT is unable to capture concepts that consist of non-adjacent words. It is not possible to annotate for these terms in MedCATTrainer. Because there are only around 10-20 occurrences of these, we decided not to include annotations for these.

While annotating, some documents turned out to be incomplete or empty. We excluded 10 patients because all the relevant information was missing from both their admission and discharge letters, leaving us with 130 patients. For 27 patients, the admission data was incomplete, but their discharge letters were complete. Because our dataset is already limited in size, we decided to keep these patients and potentially use their discharge letters as well to identify ADEs (as described in Section 4.2).

As we are only annotating for 4 categories from the explicated trigger tool (Section 3.2), we only annotated the documents that contain one or more ADEs belonging to these categories. 37 patients were exclusively labelled with ADEs not among the 4 chosen categories. So, from the 130 patients, 37 were excluded. Therefore we ended up annotating admission and discharge letters for 93 patients (186 letters).

3.4 Data: Unsupervised MedCAT Training

For MedCAT's unsupervised training procedure we need medical training corpora, these are used to create contextualized concept embeddings for the concepts in the concept database (Section 3.1). There is a wide range of multilingual sources available to obtain medical texts (Section 2.1). However, as discussed in Section 2.4, there is only a limited availability of open sources for Dutch medical texts. As we will be using MedCAT for NER+L tasks within admission and discharge letters, the provided texts should contain many medical terms with context. One corpus that is publicly available is Wikipedia, which is also available in Dutch (Section 3.4.1). Two closed sources that we were able to use for this project are the "Nederlands Tijdschrift voor Geneeskunde" (NTvG) (Section 3.4.2) and the EMC Dutch Clinical Corpus (Section 3.4.3). Besides these three corpora we also add the terms from our concept database to compile the VCB (Section 2.4.2.1). In the sections below follow descriptions for each of these corpora. In Section 3.5 we will describe how these corpora are further pre-processed and used for MedCAT's unsupervised learning step.

3.4.1 Wikipedia

Twice a month, a backup of the Dutch Wikipedia is made available at `https://dumps.wikimedia.org/nlwiki/`. From here we downloaded the dump file containing the text of all the latest versions of Wikipedia articles. The version downloaded and used for training was released on June 3th, 2021.

We are interested in a subset of all Dutch Wikipedia articles, the medical part. PetScan (`https://petscan.wmflabs.org/`) is a tool that can retrieve all Wikipedia pages within a certain category. The category we are interested in is "Geneeskunde", Dutch for "Medicine". PetScan requires users to determine the depth of the search, as Wikipedia categories follow the structure of a directed acyclic graph (DAG). Setting the depth at 5 would include 1256 categories including more than 200 categories about the paralympics, not necessarily the type of medicine-related categories we are looking for, therefore depth 4 seems to be a better cut-off value for the depth. This results in 873 categories (June 3th, 2021).

Based on this list of categories, a selection of articles was extracted from the dump file using WikiExtractor (`https://github.com/attardi/wikiextractor`). This resulted in a single text file containing only the text from the articles (no images and tables) with $\approx$ 5M tokens.

3.4.2 NTvG

The NTvG is a journal with Dutch articles that focuses on recent developments in the medical domain. It contains articles of various types: patient case descriptions, quizzes, guidelines, news, research, technical developments, editorials and more (`https://www.ntvg.nl/`). Their articles have been digitized from 1986 onwards. These documents were provided to us (June, 2021) in a MySQL format. After removing duplicates we were left with a dataset of 42247 articles. Because not all of the article types contain medical texts, we decided to remove "10 tips" (4 articles) and "editorial" (616 articles) from the selection, resulting in 41627 articles. We used BeautifulSoup to clean the html codes from the text and regular expressions to clear some metadata from the articles (Richardson, 2007). This resulted in a single text file containing over 30M tokens.

3.4.3 EMC Dutch Clinical Corpus

The EMC Dutch Clinical Corpus is a collection of 7500 anonymized clinical documents from four sources: general practitioner entries, specialist letters, radiology reports and discharge letters. EMC stands for Erasmus Medical Centre and is a Dutch university hospital in Rotterdam. The dataset was initially developed to identify contextual properties of identified concepts (Afzal et al., 2014). The recognized concepts in the dataset contain annotations on three contextual properties: negation, temporality and experiencer. These are also the domain-specific challenges in EHRs as pointed out by Harpaz et al. (2014), which we described in Section 2.5.

Every clinical document in the EMC corpus was provided in both json (with the annotations) and plain text (without annotations) format. We picked the plain text documents because we are only interested in computation of contextualized concept embeddings for MedCAT's unsupervised learning step. Some of the clinical documents from the general practitioners category seemed to contain non-sensible text giving a decoding error, this resulted in the removal of 18 documents. All other documents were merged into a single text file by pasting them after each other, separated by a newline. The EMC corpus is relatively small compared to Wikipedia and NTvG and contains a little over 200k tokens. Although it is the smallest of the three, it does best represent the use of language in clinical documents.

3.5 Concept Extraction and Linking

In order to answer how well we can identify concepts from our clinical letters and in what ways we can improve this process (Section 1.3), we start by creating our MedCAT model (Kraljevic et al., 2021). For the first step we use the text data compiled by combining (a Dutch subset of) Wikipedia, NTvG and EMC DCC, as described in Section 3.4. This is used to create our VCB and CDB files (Section 2.4.2.1). Next, we feed MedCAT our input data, consisting of our admission and discharge letters (Section 3.2.2). For each patient in the dataset we combine the admission and discharge letter into a single document before entering MedCAT.

When entered into MedCAT, the documents undergo pre-processing as described in Section 2.4.2.2. The Dutch SpaCy model 'nl_core_news_lg' was used as we are tokenizing and lemmatizing Dutch letters. This model has a list with stopwords that are removed by default. We also enabled the configuration to check whether a token is fully uppercased or not, indicating an abbreviation. This helps differentiate between terms such as "als" ("if" in Dutch) and "ALS" ("amyotrophic lateral sclerosis") or "is" (same in English) and "IS" ("infantile spasm").

MedCAT offers an additional tool, MetaCAT. This is a bidirectional LSTM that can be trained to provide meta annotations on contextual features, such as negations, temporality etc. We considered adding meta annotations on negations as we do have access to a trained MetaCAT negation model (Es et al., 2022). To decide whether this is of added value to our model, we evaluated a random sample of 4 letters from the dataset. In these letters we looked at the mentions of ADRs and considered whether negations play an important role in these. From these 4 letters only two ADR mentions contain negations, these are as follows: "Na staken losartan geen collaps meer" (After discontinuing losartan there was no collapse) and "Amiodaron gestaakt bij hyperthyeroïdie" (Discontinued amiodaron because of hyperthyroidism). The presence of negations in these sentences do not change the fact that they indicate an ADR. Because of this and additionally the low number of negations, we decided not to implement the MetaCAT negation model.

Although we do not have access to trained models on temporality and experiencer contextual properties (Section 3.4.3), we did assess the necessity of such models. We evaluated a random sample of 4 letters to assess whether these properties impact our results if not accounted for. When a letter mentioned other persons than the patient, it was never in context with both a trigger and event, making it unlikely that the presence of different experiencers influence the results. Temporality in the sense of whether a trigger-event combination occurred in the past or now is not relevant, as we want to identify it in both cases: we want to identify each ADE mention in the letters. However, relative temporality (the order of events in sentences) could be useful. Take this example: "Tijdens opname werd de novomix die patiente tweemaal daags gebruikte omgezet in eenmaal daags Lantus. Hiermee zien we geen hypoglycemieën meer en zijn de glucoses goed onder controle." (During admission the novomix, which the patient used twice a day, was changed to Lantus once a day. With this change, there were no further hypoglycemias and the glucoses were well maintained.). In this case, we need our model to understand that the hypoglycemias disappeared after Novomix was stopped, and that it has nothing to do with Lantus. Using a model to add annotations on temporality would improve performance (Section 4.2.3) and is something for future work (Section 5.2).

Before we can do any evaluation, we have to convert the found CUIs for the triggers and events into a different format to be able to compare our results to those provided by Noorda et al. (2022). The labelled data uses event categories such as in Table A.1. So we map all CUIs to their corresponding trigger tool event category, as shown in Figure 3.3.

For the triggers, we want to map the CUIs to their ATC codes, as explained in Section 3.2.1.2. An example of such a mapping is provided in Figure 3.4. This way, we arrive at the desired ATC codes.

3.5.1 Evaluation

To check how well our model performs (thereby further answering sub-question 1 as formulated in Section 1.3), we evaluate using two approaches. First, we use a

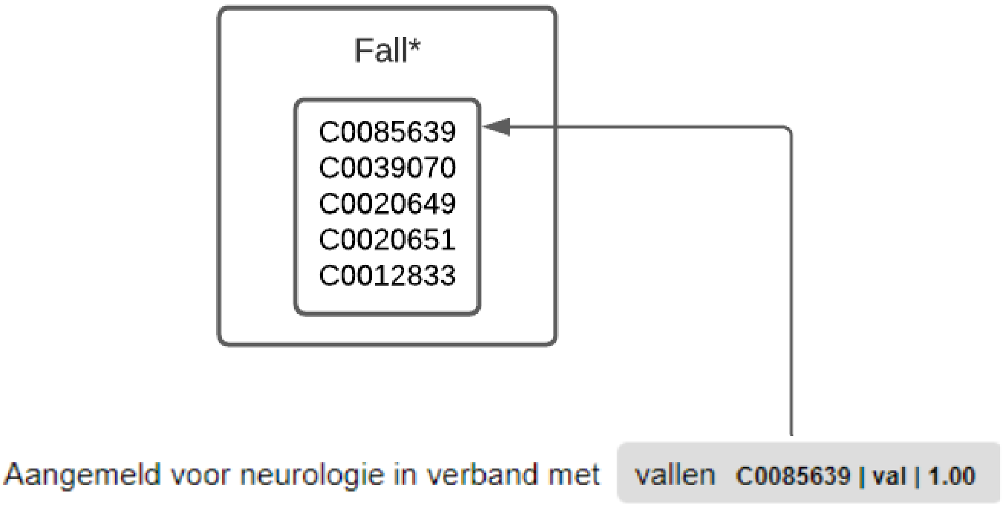

Fall*
C0085639
C0039070
C0020649
C0020651
C0012833
Aangemeld voor neurologie in verband met
vallen C0085639 | val | 1.00

FIGURE 3.3: Mapping CUIs recognized by MedCAT to their trigger event category. *Note that 'fall' is in the same category as collaps/syncope, hypotension, orthostatic hypotension and dizziness, as listed in Table A.1.

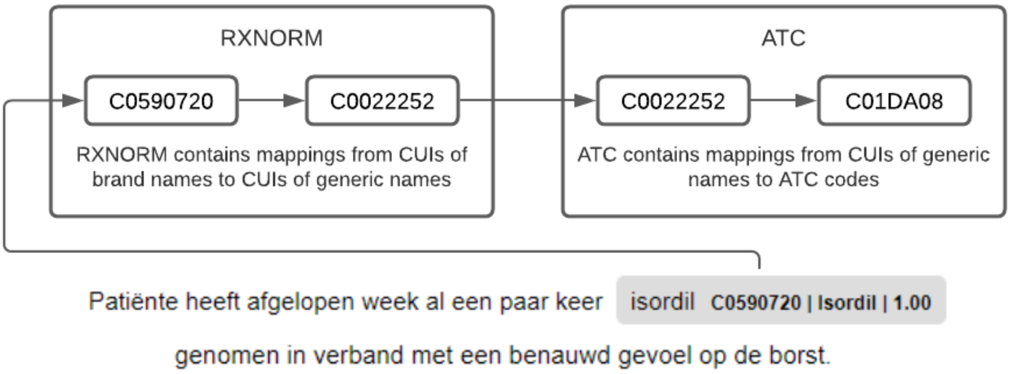

RXNORM
C0590720
C0022252
RXNORM contains mappings from CUIs of brand names to CUIs of generic names
ATC
C0022252
C01DA08
ATC contains mappings from CUIs of generic names to ATC codes
Patiënte heeft afgelopen week al een paar keer
isordil C0590720 | Isordil | 1.00
genomen in verband met een benauwd gevoel op de borst.

FIGURE 3.4: Steps to map MedCAT trigger CUIs to ATC codes.

document-level approach to see how many of the ADEs are recognized at document-level. So if a relevant trigger-event combination is detected at least once in a letter, it is counted as correct. We can use the labels assigned to the documents (Section 3.2.2) as a golden standard for the presence of an ADE at document-level. Secondly, we use a concept-level approach, this evaluates how many of the individual triggers and events are correctly identified. To evaluate this we use this the manually annotated dataset (Section 3.3). We calculate the recall on ADEs, but also on drugs and events separately. Because collecting precision scores is labour intensive, we only provide precision scores in some cases.

3.5.2 Error Analysis

We have a fully annotated dataset, this allows us to thoroughly identify mistakes made by MedCAT. We can then see what mistakes are most common: are concepts missed because their words are not in the concept database? Are concepts linked to the wrong CUIs?

Additionally, we pick 10 random letters and evaluate the concept recognition in these letters. We will attempt to provide an explanation for the errors in labelling, and define different error-categories. Such an error analysis can prove useful for improving the model in the future.

3.6 Relation Extraction

We will be analyzing admission and discharge letters from EHRs. In these letters, relations between drugs and events are not necessarily in the same sentence or even in the neighbouring sentences. As explained in Section 2.3, methods that are rule-based and shortest dependency paths might not be able to capture long-range dependencies. Therefore, we need to select models that are able to capture long term dependencies. Transformer models are non sequential, meaning that the input can be passed into it in parallel, making them effective at memorizing long term relations (Section 2.4.3).

Relation extraction relies not just on the information of the whole input text, but also on the two target entities, the trigger and the event in our case. To incorporate target entity information in this process we use the architectures from the R-BERT (S. Wu and Y. He, 2019) and RUS models (Sboev, Selivanov, et al., 2022) and adapt them for Dutch language by replacing the underlying BERT models with belabBERT.

As we are analyzing Dutch text, we choose to use belabBERT (Section 2.4.3.5). belabBERT is pre-trained on Dutch text and we can use our dataset (Section 3.2.2) to fine-tune it on our relation extraction task. BelabBERT has its own tokenizer for Dutch text, constructed using the same BPE algorithm as RoBERTa but now on Dutch web crawled texts (Y. Liu et al., 2019).

The default parameters of belabBERT are inherited from RoBERTa. We used the same hyperparameters that were used to fine-tune RoBERTa on RACE (Table 3.5) and kept all other parameters at their default values. In Mosbach, Andriushchenko, and Klakow (2020) they propose some guidelines for fine-tuning BERT on small datasets. One of these is to increase the number of epochs and train to almost zero training loss. We started at 3 epochs but ended up training all models for 10 epochs, which achieves a near zero training loss. According to Mosbach, Andriushchenko, and Klakow (2020), training for more epochs results in increased stability of the models.

Hyperparameter	RACE	SQuAD	GLUE
Learning Rate	1e-5	1.5e-5	{1e-5, 2e-5, 3e-5}
Batch Size	16	48	{16, 32}
Weight Decay	0.1	0.01	0.1
Max Epochs	4	2	10
Learning Rate Decay	Linear	Linear	Linear
Warmup ratio	0.06	0.06	0.06

TABLE 3.5: Hyperparameters used for fine-tuning RoBERTa (Y. Liu et al., 2019).

We then train and test two relation extraction models, one using the R-BERT architecture and one using the RUS architecture. We do this twice, once for the usual care data (Section 3.6.1) and once for the full letter data (Section 3.6.2) using 5-fold cross-validation. Additionally, we introduce three baseline models, two that fit the usual care data, and one that fits the full letter data. An overview of all these models is presented in Figure 3.5 and the sections below.

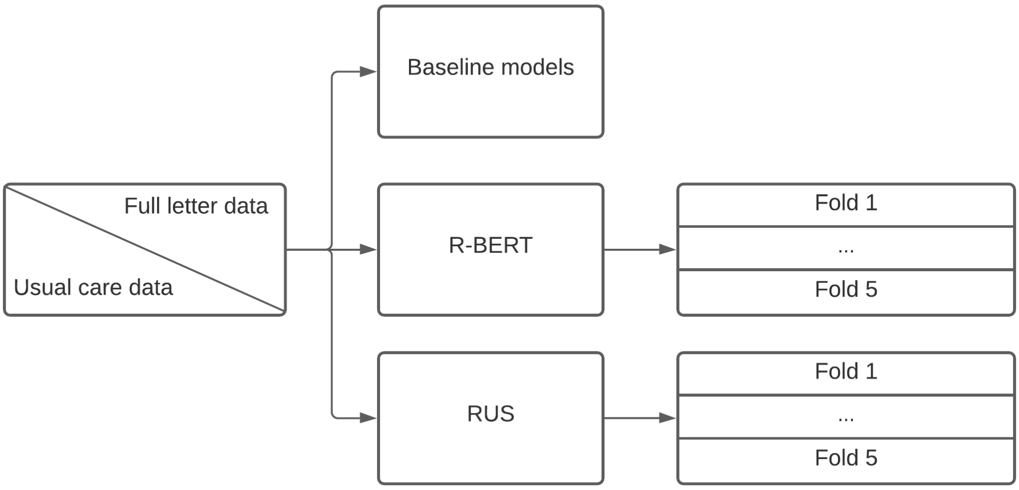


FIGURE 3.5: An overview of all the models that we will train and evaluate.

3.6.1 Usual Care models

Usual care has a two-fold definition that is explained in Section 3.2. We are mainly interested in explicit mentions of ADR recognition by usual care. With this, we mean the subset of usual care entries that follow the definition provided by Noorda et al. (2022): "an explicit documented drug-event combination by the treating physician (i.e. geriatric resident, supervised by a geriatrician) in the admission and/or discharge letter, implying that the drug-event combination was considered an ADR by usual care".

The triggers and events in these samples always appear close together in a single or two adjacent sentences. This means that for the usual care models we will not be using full letters as input, but only single or double sentences. We need both positive and negative samples. To use a less labour intensive approach without the need to mark all entities manually, we used the following method to generate our dataset of usual care instances: we collect every instance where a trigger and event appear in the same or two adjacent sentences. "Sentences" were obtained by splitting the texts by line breaks. This means that a sentence can sometimes consists of more than 1 sentence, although we set the sentence length to contain a maximum of 40 words. We chose splitting by line breaks over splitting by sentence boundaries, because the latter resulted in many mid-sentence splits, due to the custom writing style used in clinical notes.

If multiple triggers and events are found, we add each combination to our dataset. An example, "The patient uses [trigger 1] and [trigger 2], because she suffers from [event 1]. Patient also suffers from [event 2]." Now we have 4 entries:

1. trigger 1 with event 1
2. trigger 1 with event 2
3. trigger 2 with event 1
4. trigger 2 with event 2

We used the labelled triggers and events as provided by the MedCAT output. This results in a list of 270 samples, we manually went through the list and annotated whether the samples were positive or negative. Because cases could be quite ambiguous, this process was done by two annotators: a medical student and a clinical pharmacologist. Both annotators agreed to the following guidelines:

1. Score "0": When there is no relation at all or when it cannot be deducted that the author suspects a relationship.
2. Score "1": The text fragment clearly shows that the author suspects a causal relationship between the marked trigger and event. Note; we only look at a relation between the **marked** entities, so if the marked entity is "fall" and it appears twice, we only check whether the marked instance indicates recognition by usual care.
3. Score "2": Not sure if either of the other definitions fits.

Additionally, both annotators discussed some examples beforehand with the researchers that originally labelled the dataset for usual care instances, to ensure the same definition was maintained. After both individually annotating the samples, the inter annotator agreement was 88.8% (240). The other 21 cases were discussed and 17 of these were assessed as negative (scored 0), 3 as positive (scored 1) and 2 were excluded (scored 2). This resulted in 268 samples among which 95 (35.4%) positive and 173 (64.6%) negative.

3.6.1.1 Baseline model 1: Close Proximity

For our first baseline model, which we will call the close proximity model, we use a simple rule-based approach: when a trigger and event appear in the same or two adjacent sentences, mark it as an ADR.

Because the way we created the usual care dataset is equal to this rule-based approach, the recall is 100%. In addition, we can evaluate the performance of this baseline model by checking precision on the annotated usual care instances. From the precision and recall we calculate the f1 score.

3.6.1.2 Baseline model 2: Close Proximity with ADE Filter

In our second baseline model, we take the above baseline, but add an additional step: when a trigger and event appear in the same or two adjacent sentences, check whether it is an ADE. If the combination exists in the trigger tool, we mark it an ADE. Now we mark all these identified ADEs as ADRs.

Similar to the model above, the recall is still 100%, however the precision increases. The reason for providing both baselines (and not just the best one) is that the better option requires a knowledge base with information on which combinations are ADEs. Building such a knowledge base takes time, so it is also interesting to know how a simpler baseline model, one that is not dependant on something like the trigger tool, performs.

Note that for both baselines, performance will decrease when the target categories are expanded. Currently, the baselines are applied only on the triggers and events relating to 4 selected trigger tool categories (Section 3.2). When more ADEs are included, more entities will be marked as triggers and events. The amount of non-relevant trigger-event combinations will increase relatively more (compared to relevant combinations). In these baseline models, by design, this leads to an increased amount of false positives.

3.6.1.3 RUS and R-BERT

In order to train and test R-BERT and RUS on usual care instances, we pass the dataset (Section 3.6.1) with marked triggers and events as input. We use 5-folds cross-validation with a 4:1 (train:test) split and the models are trained on a GPU.

We then evaluate the performances of both models on all different folds and report the precision, recall and f1 score. We evaluate performance with and without the ADE filter, to be able to compare with both baseline 1 and baseline 2. To provide a fair comparison with baseline 1 and 2, their performances are also provided separately for each fold. Additionally, we check how well the models perform within each trigger tool category (only possible for the models with ADE filter).

3.6.2 Full Letter models

In our dataset, we were able to extract 266 ADEs among which 96 ADRs in the categories we focus on (and based on the manual annotated dataset, Section 3.3). These are within-letter ADEs and ADRs, meaning that these matches cannot occur cross-document, a cross-document case would be: event in the admission letter, trigger in the discharge letter. This is based on the assumption that combining cross-document text fragments results in incoherent texts. While the usual care models focus on finding ADRs in a span of one or two sentences, the full letter model approach is meant to identify triggers and events while providing more context. Here we give as much context as possible within the 512 tokens that are allowed by BERT models.

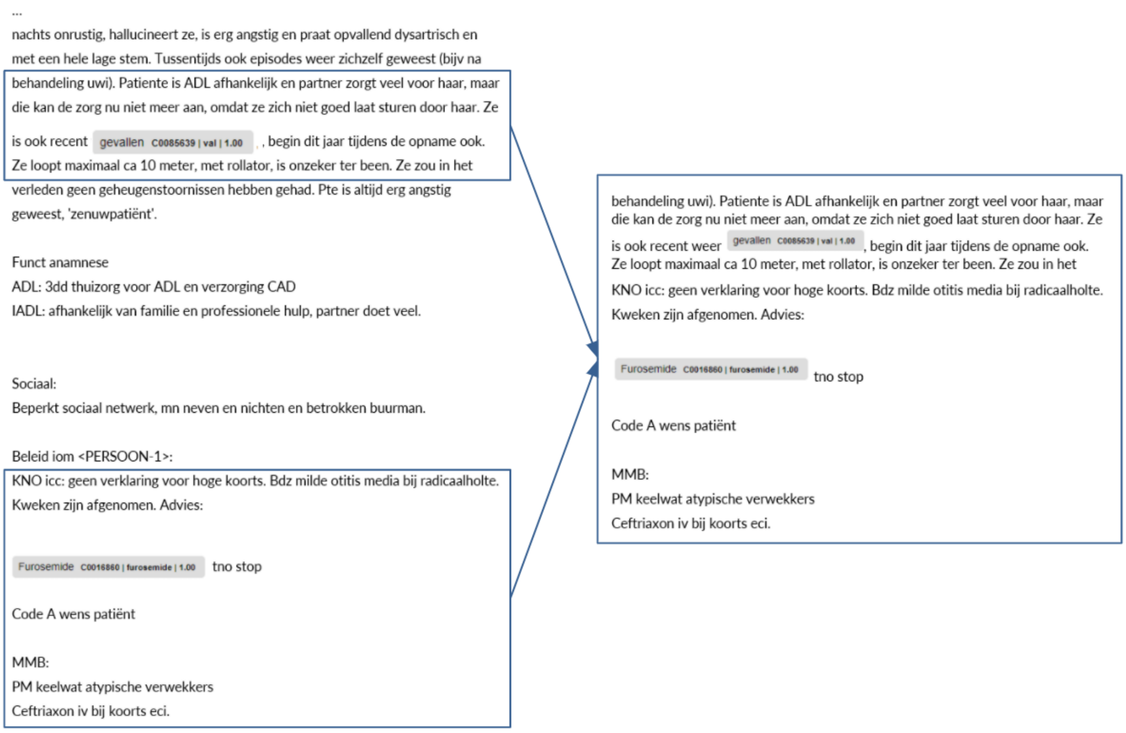
We use the following steps to pre-process the existing data into training and test sets that are appropriate for BERT models, Figure 3.6 shows a schematic representation of this process. For each letter we do the following:

1. Identify all mentions of triggers and events.

2. For every mention we keep the text in a span of 700 characters (350 to the left, and 350 to the right).

3. For every trigger and event that are ADEs according to the trigger tool, we generate a text fragment that consists of both texts concatenated in the order of appearance (and overlapping characters are corrected).

4. If for this patient the ADE has been marked as an ADR (Section 3.2), the generated sample is labelled as positive and otherwise negative.



...

nachts onrustig, hallucineert ze, is erg angstig en praat opvallend dysartrisch en
met een hele lage stem. Tussentijds ook episodes weer zichzelf geweest (bijv na
behandeling uwi). Patiente is ADL afhankelijk en partner zorgt veel voor haar, maar
die kan de zorg nu niet meer aan, omdat ze zich niet goed laat sturen door haar. Ze
is ook recent gevallen C0085639 | val | 1.00 , begin dit jaar tijdens de opname ook.
Ze loopt maximaal ca 10 meter, met rollator, is onzeker ter been. Ze zou in het
verleden geen geheugenstoornissen hebben gehad. Pte is altijd erg angstig
geweest, 'zenuwpatiënt'.

Funct anamnese
ADL: 3dd thuizorg voor ADL en verzorging CAD
IADL: afhankelijk van familie en professionele hulp, partner doet veel.

Sociaal:
Beperkt sociaal netwerk, mn neven en nichten en betrokken buurman.

Beleid iom <PERSOON-1>:
KNO icc: geen verklaring voor hoge koorts. Bdz milde otitis media bij radicaalholte.
Kweken zijn afgenomen. Advies:

Furosemide C0016860 | furosemide | 1.00 tno stop

Code A wens patiënt

MMB:
PM keelwat atypische verwekkers
Ceftriaxon iv bij koorts eci.

behandeling uwi). Patiente is ADL afhankelijk en partner zorgt veel voor haar, maar
die kan de zorg nu niet meer aan, omdat ze zich niet goed laat sturen door haar. Ze
is ook recent weer gevallen C0085639 | val | 1.00, begin dit jaar tijdens de opname ook.
Ze loopt maximaal ca 10 meter, met rollator, is onzeker ter been. Ze zou in het
KNO icc: geen verklaring voor hoge koorts. Bdz milde otitis media bij radicaalholte.
Kweken zijn afgenomen. Advies:

Furosemide C0016860 | furosemide | 1.00 tno stop

Code A wens patiënt

MMB:
PM keelwat atypische verwekkers
Ceftriaxon iv bij koorts eci.

FIGURE 3.6: Example of concatenating text fragments containing fall (event) with furosemide (trigger). Left is the (sample) letter, right is the concatenated text, that will be labelled and used as input for the full letter model. The trigger and event are marked in grey.

This way, we generated a dataset of 3712 samples, of which 1385 (37.3%) positive and 2327 (62.7%) negative. To illustrate the large number of samples in the dataset: if "fall" and "metoprolol" (these form an ADE) respectively occur 5 and 4 times in a single letter, this will result in 20 (5 · 4) samples. Whether this ADE was originally marked as an ADR determines whether all these 20 samples are labelled positive or negative. We are aware that not all these 20 samples contain the relevant information to deduct whether we can speak of an ADR. It would be better to annotate all 3712 samples independently, but due to time constraints we did not perform this step. For evaluation, we do not report performance on how well the model predicts each individual instance in the test set. We first map the individual instances back to their original annotations (the 266 ADEs and 96 ADRs). If one of the instances is positive, the ADE is marked as an ADR. So for the example: all the 20 samples lead back to a single annotation for "fall" and "metoprolol" for the corresponding patient, if ≥1 samples is positive, "fall" and "metoprolol" is marked as an ADR in that patient. This makes sense, because no matter how many negative mentions there are in a letter; if an ADR is mentioned only once in a letter, it exists for that patient.

3.6.2.1 Baseline model 3: Trigger Tool

Our third baseline model is based on the explicated trigger tool (Table 3.2). When a trigger and event of the same category appear in a letter, it is marked as an ADR. This will always result in a recall of 100%. This baseline actually relies on the a priori probability that an ADE is an ADR in our dataset. We further evaluate the performance by calculating precision and the resulting f1 score.

3.6.2.2 RUS and R-BERT

We use 5-folds cross-validation with a 4:1 (train:test) split to train and evaluate RUS and R-BERT. Both models are evaluated in terms of precision, recall and f1 score. The models are compared to each other and the baseline. We also provide an overview of the performances on the different ADE categories.

3.6.3 Error Analysis

We will analyse the errors for the usual care models and the full letter models separately. For the usual care models we will pick one of the folds and go through all the samples in the test set. We will then try and identify patterns in what is correctly labelled and what is not. Some examples will be more thoroughly examined: we generate variations on a sample to see how much changes are necessary to make the model correctly label the sample.

For the full letter model, we also pick the test results of a specific fold. However, because each fold consists of ±742 samples (mapped to ±40 ADEs/ADRs), we are not able to discuss all these and will highlight a subset of these.

Chapter 4

Results

In this chapter, we will evaluate the performance of the models as described in Chapter 3. First, we evaluate the performance of step 1: concept extraction and linking using MedCAT (Section 3.5). We were able to achieve an f1 score of 91.4% on concept recognition. Using these concepts for ADE recognition led to an f1 score of 71.4% and proved very useful in identifying additional ADEs that were missed during manual annotation. In this chapter, these results will be further analyzed and we will provide an error analysis with examples from the admission and discharge letters. Secondly, we evaluate the performance of step 2: relation extraction in several models as described in Section 3.6.1 and Section 3.6.2. The full letter models only outperformed the baseline by a little bit. However, the usual care models are promising as RUS (f1: 76.9%) and R-BERT (f1: 74.0%) perform much better than baseline (f1: 58.9%). These results, including an error analysis, will be discussed in more detail in this chapter.

4.1 Triggers and Event Recognition: Sample

Now, we will start the concept extraction and linking using MedCAT (Section 3.5). This is where we apply MedCAT to the admission and discharge letters to identify which words in the texts should be marked as triggers and events. MedCAT's role here is to match words to the concepts in our concept database, whilst also checking for spelling mistakes and applying disambiguation if necessary. We want MedCAT to recognize as many relevant triggers and events as possible. As we already noticed some shortcomings of MedCAT while exploring the data, we provide some explorational improvements in Section 4.1.1. These vary from simple adjustments to implementations into the MedCAT source code itself. Subsequently, we evaluate MedCAT's performance on a smaller sample and go through an error analysis (Section 4.1.3). The results of this error analysis can then be used to tweak MedCAT to improve its performance on the full dataset. For this smaller sample, we randomly selected 5 letters from patients that we were unable to retrieve both letters of. So for those patients we have either an admission or discharge letter, but not both. Our dataset is already limited in size, this way we do not have to exclude additional letters from the process.

4.1.1 Explorational Improvements

While setting up the pipeline to run the letters through MedCAT, we already came across some errors. We addressed two of these before moving to the next step: 1) Unrecognized drugnames. 2) MedCAT's inability to handle diacritics.

The first problem is caused by the lack of drug names present in the Dutch concept databases that we used to compile our own concept database as described

			Recall		Precision	
		Count	**Diacritics**	**No Diacritics**	**Diacritics**	**No Diacritics**
Terms	**All**	63	**0.81**	0.76	0.98	0.98
	Triggers	23	0.91	0.91	1.00	1.00
	Events	40	**0.74**	0.67	**0.97**	0.96
Concepts	**All**	37	**0.83**	0.78	0.97	0.97
	Triggers	15	0.87	0.87	1.00	1.00
	Events	22	**0.81**	0.71	0.94	0.94

TABLE 4.1: An overview of MedCAT's performance on the sample letters.

in Section 3.1. We described in Section 3.2.1.2 that we used RXNORM to retrieve the CUIs of the brand names, we did however not include the concept database of RXNORM itself because it is not specifically for Dutch language. To solve the problem of unrecognized drug names, we reason that drug names are less language-bound and that we can add international drug databases without decreasing our model's performance on Dutch language. So we decided to add the concepts of ATC (12k terms), DRUGBANK (59k) and RXNORM (196k) to our concept database.

The second problem occurs because MedCAT is originally made for English language. In English, diacritics do not occur, only in words borrowed from other languages. In Dutch, diacritics are very common and while setting up the pipeline we found that trivial words (for our use-case) like "hyponatriëmie", "hyperkaliëmie" and "hypokaliëmie" were not recognized. To solve this, we added an extra feature to MedCAT so that during pre-processing of the concept database and during spell-checking of input-text, diacritics can also be taken into account (`https://github.com/CogStack/MedCAT/pull/125`).

4.1.2 Concept Extraction on Sample: Performance

In Table 4.1 the results of MedCAT on these 5 letters are presented, we see a decent performance: MedCAT is able to annotate most of the relevant concepts. For each letter we only took into account the terms that are relevant for the triggers and events it was annotated for in the dataset provided by Noorda et al. (2022). Take for example a letter which is annotated to contain the events "Fall" and "Delirium" and the triggers "Furosemide" and "Loperamide". Then we only have MedCAT consider a subset of CUIs relevant to these triggers and events. Otherwise, we would end up with a very large amount of CUIs that we need to check manually, and now we use the annotations by Noorda et al. (2022) to only focus on the CUIs relevant to this research. This may also explain the low number of false positives (illustrated by the high precision in Table 4.1), as MedCAT has a relatively small set of CUIs to choose from.

4.1.3 Concept Extraction on Sample: Error Analysis

In Table 4.2 we provide an overview of the errors in the 5 letters. First we look at the errors made overall: MedCAT performs slightly better in recognition of triggers as compared to events. Triggers are medicines and often carry many (brand)names. On the other hand, events are subject to more ambiguity ("To fall" versus "To fall asleep") and different verb tenses. These aspects are more difficult to solve for MedCAT, which may explain its better performance on triggers.

The errors can be divided into several categories:

1. Disambiguation: "To fall asleep", here "fall" is recognized as "to fall". There is no UMLS concept that links to "fall asleep", so MedCAT will never learn to disambiguate this case.

2. Inherent to MedCAT: in the case that a concept consists of non-adjacent words ("raakte ... weg"), MedCAT is incapable of capturing the concept.

3. Lemmatization error: "Valt" should be lemmatized to "Val", which would be recognized.

4. Terms that are not in the concept database: "Wegrakingen", "Emselex", "Temgesic" and "Delier".

This disambiguation error is not something we can easily solve: as UMLS continues to expand, this problem (and similar problems) may eventually solve itself.

For the lemmatization error with "Valt", the problem is that MedCAT uses a minimal normalization length of 5 by default. We use SpaCy models specifically for Dutch language (spaCy, 2021a), these are able to correctly lemmatize "Valt". Lowering this value may proof useful for our use case, as "valt" is a common term in our ADEs, in Section 4.2 we evaluate the results with different normalization lengths.

MedCAT currently only looks at adjacent words to link these to concepts. To capture dependencies of non-adjacent words, MedCAT would need to rely on extra information such as dependency trees and POS tags.

The last error, where concepts are missing from the database, can be solved easily by adding those terms to our concept database. This is not the most appealing solution as this requires manual adjustments every time a new dataset is introduced. On the other hand, it does help to identify trivial concepts that are missing from the concept database. In our case "Wegrakingen" and "Delier" are important terms that we want to add to our database.

Doc	Recognized	Term	Count	Mapped CUI	Correct CUI	Concept Name (of correct CUI)	Explanation
1	Mapped	In slaap was [gevallen]	1	C0085639	-	-	To "fall asleep" does not have a CUI in UMLS, so MedCAT cannot disambiguate between "falling" and "falling asleep".
		Hypotensie	1	C0020649		Hypotensie	
		Dehydratie	2	C0011175		Dehydratie	
		Tamsulosine	1	C0257343		Tamsulosine	
		Citalopram	2	C1099456		Escitalopram	
		Hyponatriëmie	2	C0020625		Hyponatriëmie	Is correctly identified with the diacritics feature enabled, with diacritics disabled, no CUI is assigned.
	Missed	[Raakte] patiënt om 15.30 gedurende 45 minuten [weg]	1	-	C0039070	Syncope	Two reasons: 1) These terms are not present in the concept database 2) MedCAT is not able to catch non-adjacent terms.
		Wegrakingen	5	-	C0039070	Syncope	"Wegrakingen" is not in the concept database.
2	Mapped	Enalapril	1	C0014025		Enalapril	
		Hydrochloorthiazide	1	C0020261		Hydrochloorthiazide	
		Nierfunctiestoornissen	1	C1565489		Nierfunctiestoornis	
		Nierfunctiestoornis	1	C1565489		Nierfunctiestoornis	
		Dehydratie	1	C0011175		Dehydratie	
	Missed	-					
3	Mapped	Lanoxin	2	C0699988		Lanoxin	
		Lormetazepam	1	C0065185		Lormetazepam	
		Loperamide	1	C0023992		Loperamide	
		Digoxine	1	C0012265		Digoxine	
	Missed	Temgesic	1	-	C0006404	Subutex	Temgesic appears in RXNORM, but is somehow marked as an obsolete / suppressible term.
		Emselex	1	-	C0529351	Darifenacine	Emselex does not occur in our drug concept vocabularies: RXNORM, DRUGBANK and ATC.
4	Mapped	Verwardheid	5	C0009676		Toestand van verwardheid	
		Keppra	2	C0876060		Keppra	
		Citalopram	2	C1099456		Escitalopram	
		Verward	2	C0009676		Toestand van verwardheid	
		Levetiracetam	2	C0377265		Levetiracetam	
	Missed	Delier	2	-	C0011206	Delier	"Acuut delier" and "Delier acuut" are terms that are present in the concept database. However, "Delier" is not.
5	Mapped	Val	3	C0085639		Val	
		Nierfunctiestoornissen	1	C0035078		Nierfunctiestoornis	
		Gevallen	2	C0085639		Val	
		Vallen	1	C0085639		Val	
		Duizeligheid	1	C0012833		Duizeligheid	
		Hyperkaliëmie	1	C0020461		Hyperkaliëmie	Is correctly identified with the diacritics feature enabled, with diacritics disabled, no CUI is assigned.
		Lisinopril	2	C0065374		Lisinopril	
		Metoprolol	3	C0025859		Metoprolol	
		Orthostatische hypotensie	1	C0020651		Orthostatische hypotensie	
		Dehydratie	1	C0011175		Dehydratie	
	Missed	Valt	2	-	C0085639	Val	The verb "Valt" is not properly processed during lemmatization.

TABLE 4.2: Error Analysis of 5 random discharge letters. Lists all recognized and missed relevant terms in each letter and whether these are mapped correctly. Only unique terms are listed for each document. In case of an error we provide the explanation that we deem most likely.

4.2 Triggers and Event Recognition: Full Dataset

Now we have MedCAT annotate the rest of the data, 186 letters from 93 patients. Performance is evaluated in two different ways: at concept-level and at document-level. The concept-level approach evaluates how well MedCAT recalls individual trigger and event concepts (can be multiple per document). This is evaluated by comparing against our manually annotated dataset (Section 3.3). Document-level is a similar approach to the one used in Noorda et al. (2022), as they labelled for each admission letter whether and what ADE occurred in the document (Section 3.5.1).

First we will go over performance of the concept-level approach, followed by an evaluation of how well these concepts can be translated into ADE recognition at document-level. Both approaches are based on the same annotation strategy, it is just the evaluation methodology that is different. Therefore, the error-analysis that follows, covers both.

4.2.1 Concept-Level Performance

For the concept-level approach, we compare the output of MedCAT with the manual annotations that we created (Section 3.3). Because we have annotations for all admission and discharge letters, we run MedCAT once on the full dataset, containing 186 letters from 93 patients.

In Table 4.3 the performance of MedCAT is presented and compared to our manually created golden standard, which contains 3301 annotations. In our best model, 86% of all concepts are recognized. Within these concepts, triggers (93% recall) are significantly better recognized than events (79% recall). When lowering the minimal normalization length we see a performance increase from 84.3% to 86.1%. With minimal normalization value at 5, the strings "valt" and "Valt" (lower and uppercase) were missed respectively 35 and 19 times. Lowering the minimal normalization value to 3 allows MedCAT to lemmatize "valt". Consequently, "valt" and "Valt" are missed 0 times, which increases the concept-level performance. The reason that this performance increase does not show up in the document-level approach (Table 4.5) is because there are multiple event mentions in each letter, therefore missing a case of "valt" is compensated for by detecting another similar instance. Because the recall with minimum normalization length "3" and diacritics "on" is much higher, we choose to further evaluate using that model in Section 4.2.3.

	Recall (%)			
min_len_normalize:	**5**		**3**	
Diacritics:	**on**	**off**	**on**	**off**
All concepts (3301)	84.3	80.2	**86.1**	82.0
Events (1660)	75.2	67.5	**78.7**	71.0
Triggers (1641)	**93.3**	93.0	**93.3**	93.0

TABLE 4.3: Concept-level performance: An overview of MedCAT's performance on detecting concepts. We used the letters of 93 patients which included 3301 annotated concepts for triggers and events. We also included scores on the diacritics feature that we added to MedCAT (Section 4.1.1).

We only report the precision score for the run with diacritics on and minimum normalization length at 3. There are 147 cases marked as false positives, this results in a precision score for of at least 95.1%. The reason why we say "at least" is because many of these false positives turn out to be true positives. This is the case for all 65 triggers that are marked as false positives. If we were to correct for this, precision score raises even higher to 97.3%. This is however not completely explainable as we initially used MedCAT as a starting point for our manual annotation process. It may be explained by the fact that MedCATTrainer was quite buggy at time of use. We will further elaborate on the false positives (that really are false positives) in Section 4.2.3. For evaluation, we only evaluated the triggers and events that we knew to be ADEs in each letter (Section 3.3). An example: a letter is marked to have only one ADE, "fall" and "metoprolol". It may be that the word "delirium" also appears in the document, but we only added annotations in our "golden standard" for "fall" and "metoprolol", as these are the concepts corresponding to the labelled ADE. Concepts irrelevant to the ADEs in each letter were disregarded and not taken into evaluation.

	R	**P**	**f1**
All concepts (3301)	86.1	97.3	91.4
Events (1660)	78.7	95.2	86.2
Triggers (1641)	93.3	100.0	96.5

TABLE 4.4: Concept-level performance with precision and f1 score for the run with diacritics on and minimum normalization length at 3.

In Figure 4.1 the most frequently identified strings and concepts are depicted. The strings are the literal strings before they are mapped to their concept and are case-sensitive. So in this example, "delier" and "Delier" are two different instances. The figure shows that some words appear relatively frequent, for example "delier" makes up 7.5% (249) of all the concepts in our dataset. Our concept database consists of 1362 CUIs, but in practice our model detected only 108 correct different concepts. This means that only a relatively small part of these CUIs are used in practice. The top 20 concepts (Figure 4.1) make up 1945 (68.5%) of 2840 correctly identified cases. This shows that a small number of concepts appear very frequently while a large number of concepts only appear in a few cases.

4.2.2 Document-Level Performance

For the document-level approach, we ran MedCAT multiple times on slightly different datasets. As described in Section 3.2, we can only use the admission letters to detect ADEs, following the methodology of Noorda et al. (2022). However, during annotation it became clear that the admission letters in our dataset were not always complete. In Noorda et al. (2022), the researchers had direct access to the EHRs and could therefore manually find, for example, the medication list of a patient when this was missing in a letter. We do not have such access. This means that if we have MedCAT detect ADEs in only the admission letters, it is impossible to achieve a 100% recall score, as some of the letters we have do not contain the ADEs they are labelled for. Although we still provide the recall when we run MedCAT exclusively on the admission letters, we also decided to run it with additional data. So, to compensate for the lack of data in the admission letters, we decided to introduce 2 more versions of the dataset that we offer to MedCAT. In the first additional run, we use the admission letters, but also the discharge letters for those patients that have incomplete admission letters (Section 3.3. This is a fairer approach, as the ADEs are often named in the discharge letters (although not always, so achieving 100% recall

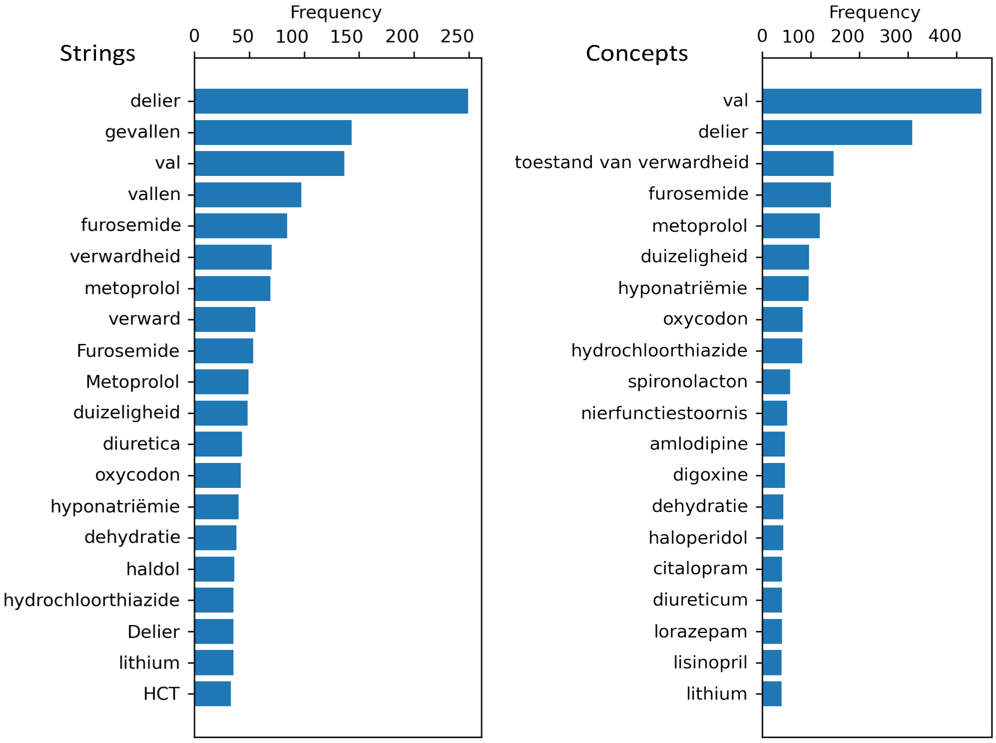


FIGURE 4.1: On the left: top 20 correctly identified strings (case-sensitive). On the right: top 20 correctly recognized CUIs, expressed using their preferred Dutch name.

is still impossible). In the second additional run, we provide the admission and discharge letters for all the patients. To see if we can improve the recall even further by using all letters to detect ADEs.

In Table 4.5 the results of MedCAT on the letters of 93 patients are presented. In the best model, the ADE recall rises significantly from 65.0% to 83.7% when we add just the 37 discharge letters for the patients with incomplete admission records. The highest recall is naturally obtained when we input the most data, when running annotation on all admission and all discharge letters, 87.8% of all the ADEs are recognized at document-level.

We also see a great increase in performance when we enable our self-implemented diacritics feature (Section 4.1.1). This makes sense in the context of our research as we specifically focus on ADEs relating to "hyponatriëmie", "hypokaliëmie" and "hyperkaliëmie", which all contain diacritics. In the trigger category the diacritics feature only causes a minor performance increase, because diacritics are rarely present in drug names. We do not see any increase in performance when we lower the minimal normalization length, this will be further explained in Section 4.2.1.

An important metric that is missing in Table 4.5, is precision. The reason is that we noticed that some false positives were actually true positives. This means that they were originally missed in the annotations provided by Noorda et al. (2022). To report a fair precision score, it is therefore required to go through all documents and screen the legitimacy of the false positive label. This is labour intensive and therefore we decided to do this for only the smallest run, the one that relies just on the admission letters. In the original run (with diacritics on and minimum normalization length at 3) recall was 65.0%, because 219 out of 337 ADEs were recognized. In our case, MedCAT labelled 512 ADEs, meaning we have 293 false positives. This

	Recall in % (total n=337)					
	ADEs		**Events**		**Triggers**	
min_len_normalize:*	**3 & 5**		**3 & 5**		**3 & 5**	
Diacritics:	**On**	**Off**	**On**	**Off**	**On**	**Off**
Admission letters	**65.0**	48.7	**89.9**	70.0	**68.2**	68.0
Admission & selected discharge letters	**83.7**	64.7	**95.8**	76.0	**86.9**	86.6
Admission & all discharge letters	**87.8**	68.0	**95.8**	76.0	**91.4**	90.8

TABLE 4.5: Document-level performance: An overview of MedCAT's performance on detecting triggers, events and ADEs. We used the letters of 93 patients which included 337 annotated trigger-event combinations (ADEs). The 'ADEs' column shows how often the full trigger-event combination was recognized. We also included scores on different minimal normalization lengths and the diacritics feature that we added to MedCAT (Section 4.1.1). *Scores for min_len_normalize "3" and "5" were equal in all cases.

results in very low precision scores (Table 4.6). However, after screening, we found that out of those 293 false positives, 131 ADEs were actually true positives. This means that we now recall 350 (219+131) out of 468 (337+131) ADEs and precision also increases. We refer to this as the "corrected" version in Table 4.6. It is impossible to achieve a 100% recall score for because the admission letters are incomplete.

	ADEs		
Dataset:	**R**	**P**	**f1**
Admission letters	65.0	42.8	51.6
Admission letters (corrected)	74.8	68.4	71.4

TABLE 4.6: MedCAT scores on document-level recognition of ADEs in the admission letters with precision scores.

To further evaluate the document-level performance we pick the best performing model and further examine it. So, for the next tables and statistics we use the model that is based on all admission and discharge letters with diacritics on. In Table 4.7 the recall of ADEs, triggers and events within each category is provided. There is a significant difference in performance between the categories, "Renal Insufficiency" for example, is among the worst-scoring categories in both events and triggers (and consequently also in ADEs). On the other hand, "Fall" seems to be an easier category in terms of both events and triggers that correspond to it, as scores within this category are very high (>95%).

4.2.3 Triggers and Event Recognition: Error Analysis

In the error analysis we will also continue to analyze the results of the model applied to the full dataset. From the 3301 annotated concepts, 2841 were correctly recognized and 460 concepts were missed. Figure 4.2 shows the most common mistakes of our model. There are a variety of reasons why the different strings are missed.

		Recall		
Event Category*	**Total**	**ADEs**	**Events**	**Triggers**
Fall	130	95.4 (124)	100.0 (130)	95.4 (124)
Delirium	78	82.1 (64)	100.0 (78)	82.1 (64)
Electrolyte Disturbances	71	88.7 (63)	91.5 (65)	95.8 (68)
Renal Insufficiency	58	77.6 (45)	86.2 (50)	89.7 (52)

TABLE 4.7: MedCAT scores on document-level recognition of ADEs, events and triggers within different event categories. *We only give one term corresponding to the event category here ("Fall" also covers "collaps", "hypotension" etc.), the other terms belonging the event categories can be found in Table A.1.

We will first look at the most frequently missed trigger strings. "Oxynorm", "monocedocard" and "natriumvalproaat" are all drug brand names that are missed because they do not occur in our concept database (their generic variants are included though). Similarly "levodopa/carbidopa" and "levodopa/benserazide" are not in the concept database but are also missed because MedCAT splits words separated by "/" and will never be assigned to a single concept: within "levodopa/...", "levodopa" is recognized. "Tiotropium" is missed because the ATC code (R03BB04) for "Tiotropium" is linked to "Tiotropium bromide". So in our concept database we have "Tiotropium bromide" but not "Tiotropium". When looking at the WHO ATC description for tiotropium bromide, it says "expressed as tiotropium". So although the term in our concept database is correct, "tiotropium" is a more common expression in practice and it would be beneficial to add such terms to the concept database. "Amitryptiline" is not identified because the spelling is too far off, the correct word is "amitriptyline", the "i" and "y" are swapped. Lastly, "amlodipine" is not annotated in 4 cases. This may be due to the context of its appearance, because the term is mapped correctly in our concept database, illustrated by the fact that it is recognized correctly 46 times in the dataset.

Now for the event strings, "orthostase", "sufheid", "in de war", "suf", "Sufheid", "bewustzijnsverlies" and "buiten bewustzijn" lack annotations because the terms are not in the concept database. The missed event strings appear a lot more frequent than the trigger strings, so adding those to create a more complete concept database would be very beneficial. The reason that "nierinsufficientie" and "nierinsufficiëntie" are missed will be explained later this section when elaborating on false positives. Lastly, "val" is not recognized in 13 cases, similar to "amlodipine" this may be due to the context of its appearance.

Another frequent cause of errors is when a single word contains a drug name followed by "gebruik" (usage) such as "metoprololgebruik", "citalopramgebruik" and "diureticagebruik". These are not recognized as concepts, causing several drugnames to be missed. A quick solution for our use case would be to identify strings ending with "gebruik" and seperate all preceding characters with a space. Another option would be to add a concept for each of these cases, this is however much more labour intensive.

In Table 4.8 and Table 4.9 we do an error analysis of 10 random letters. For each of these letters we show which concepts are assigned by our model and whether these are correct. We also provide an overview of the concepts that should have been recognized but were missed, in 5 of the letters, no concepts were missed (although this can still mean there are incorrect mappings). The mistakes made can be divided into a few categories:

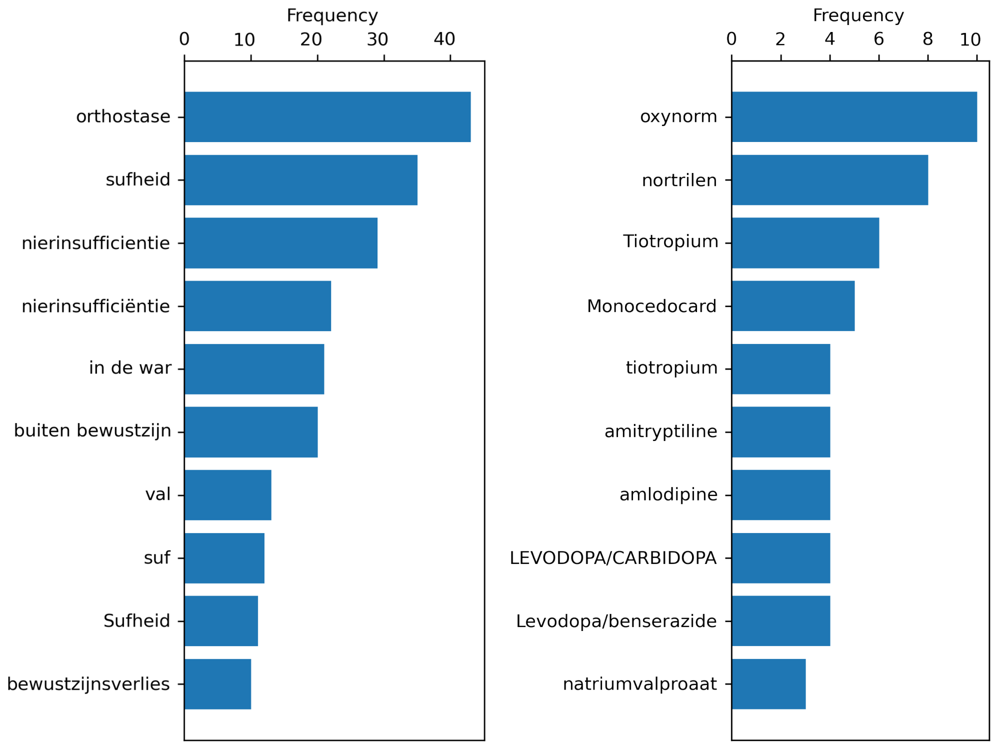


FIGURE 4.2: The top 10 most frequently missed strings for events (left) and triggers (right).

1. Terms not in our concept database: Dutch concept databases are still a work in progress and continue to improve. For example, SNOMED CT Netherlands edition is updated twice a year, and in the last update (30 September 2021) around 1000 new concepts and 5000 new translations were added. For now, we will have to manually add missing terms for specific use cases.

2. Wrong mapping of concepts: This will be explained further down in this section, when we elaborate on false positives.

3. Ambiguous language: A difficult case is "NF" (letter 9 in Table 4.9). Health practitioners normally use "NF" to express "nierfunctie" (renal function), but in this context the doctor used it to express "nierfunctiestoornis" (renal dysfunction).

Another step that could potentially lead to errors is where the CUIs are linked to the ADE categories. For the event CUIs this step is not sensitive to errors because there are only 18 possible event CUIs (Table A.1). However, there are 1348 trigger CUIs in our concept database of which 94 were detected in the dataset (Table 4.10). These are mapped to corresponding ATC codes using the methodology described in Section 3.5. This mapping step successfully linked all trigger terms to the correct ATC code.

In the predictions by our model, there were also false positives. Here it is important to distinguish between the document- and concept-level approach. In the concept level approach, we look at how many concepts are correctly labelled, despite presence of negations or other contextual properties. For example: marking "drowsiness" in the sentence "patient did not suffer from drowsiness" is correct and not a false positive here. In contrast, for the document-level approach where we are actually labelling for ADEs, this is very relevant because it indicates that drowsiness

Doc	Recognized	Term	Count	Mapped CUI	Corrected CUI	Concept Name (of correct CUI)	Explanation
1	Mapped	enalapril	1	C0014025		Enalapril	
		hydrochloorthiazide	1	C0020261		Hydrochloorthiazide	
		atenolol	1	C0004147		Atenolol	
		temazepam	1	C0039468		Temazepam	
		paroxetine	4	C0070122		Paroxetine	
		gevallen	1	C0085639		Val	
	Missed	buiten bewustzijn	1	-	C0039070	Syncope	Term not in concept database
		nortrilen	1	-	C0028420	Nortriptyline	Term not in concept database
		Collaps	2	-	C0039070	Syncope	Unknown
2	Mapped	nierinsufficiëntie	2	C0035078	C1565489	Nierfunctiestoornis	Explained this section
		furosemide	5	C0016860		Furosemide	
		inspra	2	C1144054		Inspra	
	Missed	-					
3	Mapped	furosemide	2	C0016860		Furosemide	
		perindopril	1	C0136123		Perindopril	
		spironolacton	1	C0037982		Spironolacton	
		hyponatriëmie	1	C0020625		Hyponatriëmie	
	Missed	-					
4	Mapped	lisinopril	3	C0065374		Lisinopril	
		metoprolol	2	C0025859		Metoprolol	
		temazepam	1	C0039468		Temazepam	
		nierinsufficiëntie	1	C0035078	C1565489	Nierfunctiestoornis	Explained this section
		dehydratie	1	C0011175		Dehydratie	
		hypotensie	1	C0020649		Hypotensie	
		hyponatriëmie	1	C0020625		Hyponatriëmie	
	Missed	-					
5	Mapped	verwardheid	4	C0009676		Toestand van verwardheid	
		lorazepam	3	C0024002		Lorazepam	
		verward	6	C0009676		Toestand van verwardheid	
		hyponatriëmie	4	C0020625		Hyponatriëmie	
		delier	1	C0011206		Delier	
		oxycodon	1	C0030049		Oxycodon	
	Missed	desoriëntatie	2	-	C0009676	Toestand van verwardheid	
		ipratropium	2	-	C0027235	Ipratropium	
		in de war	1	-	C0009676	Toestand van verwardheid	
		oxynorm	2	-	C1602113	Oxynorm	
		suf	1	-	C0013144	Suf voelen	

TABLE 4.8: Error Analysis of random documents 1-5.

Doc	Recognized	Term	Count	Mapped CUI	Corrected CUI	Concept Name (of correct CUI)	Explanation
6	Mapped	hypokaliëmie	2	C0020621		Hypokaliëmie	
		furosemide	1	C0016860		Furosemide	
		delier	1	C0011206		Delier	
	Missed	-					
7	Mapped	dehydratie	1	C0011175		Dehydratie	
		delier	1	C0011206		Delier	
		furosemide	2	C0016860		Furosemide	
		oxycodon	1	C0030049		Oxycodon	
	Missed	sufheid	1	-	C0013144	Suf voelen	
		achteruitgang nierfunctie	1	-	C1565489	Nierfunctiestoornis	Explained this section
8	Mapped	hyponatriëmie	2	C0020625		Hyponatriëmie	
		triamtereen	4	C0040869		Triamtereen	
		HCT	1	C0020261		Hydrochloorthiazide	
		hydrochloorthiazide	3	C0020261		Hydrochloorthiazide	
	Missed	-					
9	Mapped	irbesartan	1	C0288171		Irbesartan	
		ibuprofen	1	C0020740		Ibuprofen	
		HCT	3	C0020261		Hydrochloorthiazide	
	Missed	nierfunctiestoornissen	1	-	C1565489	Nierfunctiestoornis	Explained this section
		NF	3	-	C1565489	Nierfunctiestoornis	Normally NF simply means "nierfunctie", but this is a rare case where it is used to abbreviate "nierfunctiestoornis".
10	Mapped	captopril	3	C0006938		Captopril	
		hyponatriëmie	2	C0020625		Hyponatriëmie	
		hydrochloorthiazide	2	C0020261		Hydrochloorthiazide	
		HCT	2	C0020261		Hydrochloorthiazide	
	Missed	hydrochloorthiazode	1	-	C0020261	Hydrochloorthiazode	

TABLE 4.9: Error Analysis of random documents 6-10.

	Events	Triggers
CUIs in concept database	18	1348
CUIs that occur atleast once	14	94

TABLE 4.10: Total number of unique CUIs that are included in our concept database and the amount of unique CUIs actually detected in our letters.

cannot be part of an ADE. Below, we will first discuss false positives at concept-level, then at document-level.

In the predictions by our concept-level model, there were 147 false positives. 65 of these false positives are actually true positives, these were corrected for, so 82 remain. In 56 of these cases, the correct word span is recognized, but the wrong CUI is assigned. This is the case for 56 (38.1%) of the false positives. One of these cases is "Temazepam" which is actually assigned the right CUI, but we falsely annotated it during the annotation process, so it is actually a true positive (this has been taken into account in the "corrected" version in Table 4.4). The other 55 cases are all string variations of "nierinsufficiëntie", and are all mapped to the "C0035078" CUI for kidney failure, instead of the true CUI "C1565489" for renal insufficiency. This mistake is made because our concept database (Section 3.1) contains the word "nierinsufficiëntie" only for the concept of kidney failure and not for renal insufficiency. This example shows that concept databases are not always complete and do not always contain perfect mappings. However, this mismapping of "nierinsufficiëntie" does not pose a problem in our use case as renal insufficiency and kidney failure both map to the same event category (Table A.1). The remaining 26 cases were ambiguity-related. The majority of these were caused by "viel" being interpreted as the past tense of "fall" in cases where it related to either "opvallen" (stands out) or "fell asleep".

For the document-level approach, we manually went over all the false positives in the admission letter run (Table 4.6). We performed a manual correction on the false positives, as some concepts were missed during the annotation process. After manual correction, there are still a 162 false positives. For each of these, we marked the error cause(s) (Table 4.11). Negations most often lead to false positives. In Section 3.5 we decided not to use a negation model, even though we had access to the negation model by Es et al. (2022). Now that we see that negations are the most common cause for false positives, we do run the model once more with this negation model. As a result, the amount of false positive ADEs was reduced by 12, but it also causes 1 of the true positives to be missed. The letters may contain medical history and prescribed medication for future use, which often leads to errors. Additionally, ambiguous use of words such as "fall" (this one in particular) and the presence of conditional statements result in false positives.

4.3 Identification of Relations

We will now evaluate the results for the relation extraction part. We will start by analyzing the performance of the usual care models RUS and R-BERT (Section 3.6.1) on the 268 samples using 5-fold cross-validation and compare them to each other and the baseline. This is followed by an error analysis, to see if we can identify common causes of errors. Next, we evaluate performance for the full letter models RUS and R-BERT (Section 3.6.2) against baseline and each other. Similarly, this is followed by an error analysis, with the aim to identify structural mistakes.

Error Category	Total	Examples
Negations	72	"stopped using sotalol" "never falls"
Medical History	40	"patient fell in may" "2007 delirium"
Prescribed Medication	20	"start using haloperidol"
Disambiguation	20	"falls asleep"
Conditional Statements	17	"if sleep problems persist, we may consider oxazepam"

TABLE 4.11: An overview of the reasons for false positive identifications of ADEs.

4.3.1 Usual Care models: Performance

First, we evaluate the performance of baseline model 1, the close proximity model (Section 3.6.1.1). We are aiming to identify cases of ADR recognition by usual care of which the definition is explained in Section 3.2. In our dataset there are 268 usual care instances of which 95 (35.4%) are positive, the other 173 (64.6%) are negative. In table Table 4.12, the precision score concerns the percentage of entries that are positive usual care instances. The recall is 100% as explained in Section 3.6.1.1.

The performance of baseline model 2 (Section 3.6.1.2) is also shown in Table 4.12. Here we apply an additional ADE filter; all samples that are not ADEs (as defined by the chosen trigger tool categories), are removed from evaluation. Now the dataset consists of 196 usual care instances of which 84 (42.9%) positive and 112 (57.1%) negative samples. The proportion of positive samples has now increased from 35.4% to 42.9%. Because the baseline depends on the a priori probability that the marked instance is a positive usual care instance, baseline 2 scores higher than baseline 1.

	Samples	Recall	Precision	f1 Score
Baseline 1 (without ADE filter)	268	100.0	35.4 (95)	52.3
Baseline 2 (with ADE filter)	196	100.0	48.5 (95)	65.3

TABLE 4.12: Performance of our close proximity baseline models on identification of usual care instances.

Table 4.13 shows the performance of all models on the unfiltered usual care dataset (268 samples) over 5 different folds. Both BERT models outperform the baseline model by >20% (based on average f1 scores). The standard deviations for RUS and R-BERT are 7.5% and 10.3% respectively so the difference of 3.2% in f1 score is not significant. However, because RUS has both a higher f1 score and a smaller standard deviation, this seems to be the preferred model.

Table 4.14 shows the performance of all models on the filtered usual care dataset (196 samples) over 5 different folds. All models perform better with the ADE filter in terms of average f1 scores. However, performance of RUS and R-BERT increases only slightly by 0.3% and 0.6% respectively, whereas the baseline improves by 6.6%. Although both BERT models still outperform the baseline model, the difference is

	Samples in test set		**RUS**			**R-BERT**			**Baseline 1**		
Fold	**Pos**	**Neg**	**R**	**P**	**f1**	**R**	**P**	**f1**	**R***	**P**	**f1**
1	17	36	76.5	56.5	65.0	47.1	66.7	55.2	100.0	32.1	48.6
2	16	37	81.3	76.5	78.8	87.5	87.5	87.5	"	30.2	46.4
3	15	38	66.7	76.9	71.4	66.7	66.7	66.7	"	28.3	44.1
4	28	25	85.7	85.7	85.7	89.3	71.4	79.4	"	52.8	69.1
5	19	37	84.2	80.0	82.0	84.2	72.7	78.0	"	33.9	50.6
Average:			78.9	75.1	**76.6**	75.0	73.0	73.4	"	35.4	52.3
Std of average:			±6.9	±9.9	±7.5	±16.1	±7.6	±10.3	±0	±8.9	±8.9

TABLE 4.13: A comparison between the usual care models: RUS, R-BERT and the baseline model. *As explained in Section 3.6.1.1, the baseline recall is always 100%.

smaller: RUS by 17.3%, R-BERT by 15.1%. For the BERT models, standard deviations also increase significantly, making R-BERT's superiority over the baseline disputable with a standard deviation of ±15.7%. Because RUS significantly outperforms the baseline, and has a higher f1 score and lower standard deviation than R-BERT, this is the preferred model.

ADE samples in test set:			**RUS**			**R-BERT**			**Baseline 2**		
Fold	**Pos**	**Neg**	**R**	**P**	**f1**	**R**	**P**	**f1**	**R***	**P**	**f1**
1	14	25	71.4	58.8	**64.5**	50.0	70.0	58.3	100.0	35.9	52.8
2	14	24	92.9	86.7	89.7	92.9	92.9	**92.9**	"	36.8	53.8
3	9	22	55.6	62.5	**58.8**	55.6	50.0	52.6	"	29.0	50.0
4	28	18	85.7	88.9	**87.3**	89.3	75.8	82.0	"	60.9	75.7
5	19	23	84.2	84.2	**84.2**	84.2	84.2	**84.2**	"	45.2	62.3
Average:			80.0	76.2	**76.9**	74.4	74.6	74.0	"	41.6	58.9
Std of average:			±13.2	±12.9	±12.7	±17.9	±14.5	±15.7	±0	±11.0	±9.3

TABLE 4.14: A comparison between the usual care models, but now with ADE filter applied. Positive means those recognized by usual care. *As explained in Section 3.6.1.1, the baseline recall is always 100%.

We combined all 5 test sets from the different folds and accumulated these to report the performance of RUS and R-BERT per event category, as shown in Table 4.15. Performance is quite similar among the different categories and between the RUS and R-BERT models. The only odd result is that of R-BERT in the "fall" category. Here, the f1 score is only 50.0%, which is 26.2% lower than RUS' f1 and also much lower than R-BERT's performance on other categories.

4.3.2 Usual Care models: Error Analysis

In this error analysis we will analyze a subset of the results of RUS and R-BERT on the ADE filtered results. We analyze the errors in the test set of the first fold. The test set of the first fold contains 39 examples, of which 14 positive and 25 negative (Table 4.14). An overview of the mistakes is provided in Table 4.16. RUS made more mistakes than R-BERT (11 vs 10). The errors by RUS in this fold are mostly false

	Samples:		RUS			R-BERT			Baseline 2		
Event Category:	**Pos**	**Neg**	**R**	**P**	**f1**	**R**	**P**	**f1**	**R***	**P**	**f1**
Fall	10	36	80.0	72.7	76.2	50.0	50.0	50.0	100.0	21.7	35.7
Electrolyte Disturbances	50	21	86.0	78.2	81.9	88.0	81.5	84.6	"	70.4	82.6
Renal Insufficiency	12	8	83.3	83.3	83.3	83.3	71.4	76.9	"	60.0	75.0
Delirium	12	47	58.3	87.5	70.0	58.3	87.5	70.0	"	20.3	33.7

TABLE 4.15: A comparison between the usual care models for each of the event categories.

positives, whereas R-BERT produces false negatives more often. 7 errors are shared by both models (examples 1-7 in Table 4.16).

For the first example, both models falsely label it as a positive instance. By design, RUS and R-BERT are not able to distinguish which instance of (in this case) "Hydrochloorthiazide" is targeted, the models simply incorporate extra information on the entity "Hydrochloorthiazide" in the vectors (and do so in different ways). During annotation, we agreed that only the marked entities should be relevant in deciding whether it is a true usual care instance (Section 3.6.1), that is why example 1 was annotated as negative by the annotators. We will provide two examples from the dataset to illustrate this:

1. Example 1 from Table 4.16: "Milde <e1> hyponatriëmie </e1>, geduid bij Hydrochloorthiazide; deze werd gestaakt en het natrium herstelde. Hypocalciëmie: bij <e2> Hydrochloorthiazide </e2> en nierfunctiestoornissen"
   (a) Mild <e1> hyponatremia </e1>, interpreted as caused by hydrochlorothiazide; which was stopped and lead to recovery of sodium levels. Hypocalcemia: caused by <e2> Hydrochloorthiazide </e2> and renal dysfunction
2. A subsentence of above example that also exists in our dataset with a different marked entity for "Hydrochloorthiazide": "Milde <e1> hyponatriëmie </e1>, geduid bij <e2> Hydrochloorthiazide </e2>; deze werd gestaakt en het natrium herstelde"
   (a) Mild <e1> hyponatremia </e1>, interpreted as caused by <e2> hydrochlorothiazide </e2>; which was stopped and lead to recovery of sodium levels

The first one is annotated by the annotators as negative and the second one as positive. Both cases are marked as positive by RUS and R-BERT. Inherent to their methodology, these models do not take into account which instance of "Hydrochloorthiazide" is relevant in the first case. A different evaluation methodology on our side could bypass the problem as follows: for each sample, keep track of the document it corresponds to. If just one instance that contains both "Hydrochloortiazide" and "hyponatriëmie" is marked as positive, the document is labelled as positive for the ADR.

Inherent to deep learning is that it is difficult to track down the origins of mistakes. Take example 6, the fragment "<e1> hyponatriemie </e1> bij <e2> HCT </e2>" (Hyponatremia caused by HCT) clearly denotes a positive instance but is missed by both. However, when asking the models to label just the fragment, they label it correctly. This shows that too much irrelevant context, noise, disrupts performance.

Example	Mistake made by	Predicted Label	Correct Label	Text
1	Both	1	0	Milde <e1> hyponatriëmie </e1>, geduid bij Hydrochloorthiazide; deze werd gestaakt en het natrium herstelde. Hypocalciëmie: bij <e2> Hydrochloorthiazide </e2> en nierfunctiestoornissen
2	"	0	1	indien ondanks halveren <e2> pramipexol </e2> nog hallucinaties (ikv <e1> delier </e1>) dan clozapine, indien motorische onrust dan lorazepam overwegen.
3	"	0	1	indien ondanks halveren pramipexol nog hallucinaties (ikv <e1> delier </e1>) dan clozapine, indien motorische onrust dan lorazepam overwegen. pm cholinesterase remmer starten als hallucinaties na halveren <e2> pramipexol </e2> niet verbeteren
4	"	0	1	Hyponatriëmie: patiënte had een <e1> hyponatriëmie </e1> bij opname, passend bij een absoluut zouttekort en citalopramgebruik. De <e2> citalopram </e2> is afgebouwd en haar intake is verbeterd. Het natrium normaliseerde.
5	"	1	0	Bumetanide 1d1 mg (was <e2> furosemide </e2> 2d 40 mg; verlaagd ivm mgl betere opname uit darm bij rechts decompensatio cordis)KCL drank 1d 30 mmol t/m 6-8-2015 (ivm <e1> hypokaliëmie </e1>)
6	"	0	1	Opname neurologie ivm epileptisch insult wv depakine opgehoogd 2dd 125 <e1> hyponatriemie </e1> bij <e2> HCT </e2>. Respiratoire insufficientie bij pneumonie en dec. cordis. influenza type A en enterobacter cloacae. Delier. Afspraak niet reanimeren
7	"	1	0	- Bij <e1> hypokaliëmie </e1> kan dosering <e2> spironolacton </e2> worden opgehoogd
8	RUS	1	0	pm <e2> citalopram </e2> halveren indien natrium niet stijgt met betere intake (vooralsnog <e1> hyponatriemie </e1> geduid bij absoluut zouttekort bij urine Na van 28)
9	"	1	0	in VG bij huisarts somberheid wv <e2> citalopram </e2>, nog steeds sombere stemmingFunctioneel: <e1> vallen </e1> en niet meer kunnen opstaan uit stoel DD verminderde kracht benen bij DD vitamine D deficientie (onwaarschijnlijk gezien vitamine D suppletie), hyponatriemie (hyponatriemie is niet zo laag en verder geen klachten die daarbij passen)
10	"	1	0	<e1> Hypokaliemie </e1>. DD: bij verminderde intake / bij lisdiuretica/ hypomagnesiemie (dd bij eerder PPI, lisdiuretica, matige intake) CAVE: digoxine gebruik Beleid: ophogen <e2> spironolacton </e2>, start orale kaliumsuppletie en magnesiumsuppletie, stop PPI, verlagen dosering lisdiuretica. Geen ACE-remmer ivm lage bloeddrukken.
11	"	1	0	Bumetanide 1d1 mg (was <e2> furosemide </e2> 2d 40 mg; verlaagd ivm mgl betere opname uit darm bij rechts decompensatio cordis)KCL drank 1d 30 mmol t/m 6-8-2015 (ivm <e1> hypokaliëmie </e1>)
12	R-BERT	0	1	Verder: status na <e1> orthostatische hypotensie </e1>, vitamine B12 deficiëntie, wisselende glucosewaarden bij DM2, hyponatriëmie en rigiditeit bij <e2> citalopram </e2> gebruik.
13	"	0	1	Verder: status na orthostatische hypotensie, vitamine B12 deficiëntie, wisselende glucosewaarden bij DM2, <e1> hyponatriëmie </e1> en rigiditeit bij <e2> citalopram </e2> gebruik.
14	"	0	1	<e1> Val </e1> DD bij sepsis (UWI en pneumonie bij vieze urine en verdichting li basaal op X-thorax), orthostatisch (bij <e2> propanolol </e2>), cardiaal (bij ischemie danwel ritmestn), cave longembolie (bij maligniteit in VG en afw re op X-thorax)

TABLE 4.16: An overview of the errors made by both BERT models in the test set of fold 1. "1" stand for a positive label, and "0" for negative.

Example 14 is something we also came across during annotation; inferences. In this example "fall" is caused by "low blood pressure" which is caused by "propanolol", and so the fall is caused by propanolol. R-BERT is unable to recognize that it is a positive instance. Further experimenting with the following inputs (shortened variations of example 14):

1. <e1> Val </e1> DD bij sepsis (UWI en pneumonie bij vieze urine en verdichting li basaal op X-thorax), orthostatisch (bij <e2> propanolol </e2>)
    (a) <e1> Fall </e1> DD sepsis (UTI and pneumonia with dirty urine and densification left basal on X-thorax), orthostatic (caused by <e2> propanolol </e2>)
2. <e1> Val </e1> DD bij sepsis (UWI en pneumonie bij vieze urine en verdichting li basaal op X-thorax), <e2> propanolol </e2>
    (a) <e1> Fall </e1> DD sepsis (UTI and pneumonia with dirty urine and densification left basal on X-thorax), <e2> propanolol </e2>
3. <e1> Val </e1> DD bij sepsis, orthostatisch (bij <e2> propanolol </e2>)
    (a) <e1> Fall </e1> DD sepsis, orthostatic (caused by <e2> propanolol </e2>)

To empirically experiment whether inference is the cause of failure, we presented R-BERT with 3 variations of example 14. In the first one we took just the relevant part of the sentence, in the second one we removed the inference and in the third part we removed the noise in between the relevant entities but kept the inference. Only the prediction for the third variation was successful.

Although it is difficult to deduct the cause of errors, the presence of irrelevant context seems to play a role here. Experimenting with different pre-processing steps could help reduce the noise in the input samples,such as picking only the span from first entity - second entity (and maybe a range of tokens around these) or removing text between parenthesis that does not contain relevant entities.

4.3.3 Full Letter models: Performance

In the third baseline model, we match all cases that are ADEs according to the trigger tool (Table 3.2) in both the admission and discharge letters and mark them as ADRs. In our dataset, there are 96 ADRs among the 266 ADEs. Logically, marking all of these as ADRs results in a 100% recall score (Table 4.17). As a result, the precision score is much lower, at 36.1%. The precision score is in line with expectations, as the ADE trigger tool has a precision of 41.8%, as shown by Noorda et al. (2022).

	Samples	**Recall**	**Precision**	**f1 Score**
Baseline 3	266	100.0	36.1 (96)	53.6

TABLE 4.17: Performance of our trigger tool baseline model on identification of ADRs.

Table 4.18 shows the performance of all full letter models on the 266 samples over 5 folds. RUS and R-BERT outperform the baseline by respectively 7.6% and 6.9% in terms of f1 score. RUS has a very low standard deviation (±3.7%), whereas R-BERT has a high standard deviation (±14.0%) making its improved f1 score over the baseline insignificant.

ADEs in test set:			**RUS**			**R-BERT**			**Baseline 3**		
Fold	**Pos**	**Neg**	**R**	**P**	**f1**	**R**	**P**	**f1**	**R***	**P**	**f1**
1	23	41	91.3	46.7	**61.8**	69.6	51.6	59.3	100.0	35.9	52.8
2	26	40	73.1	44.2	55.1	84.6	51.2	**63.8**	"	39.4	56.5
3	19	39	78.9	48.4	**60.0**	63.2	40.0	49.0	"	32.8	49.4
4	7	14	57.1	80.0	66.7	71.4	100.0	**83.3**	"	33.3	50.0
5	21	32	95.2	43.5	**59.7**	47.6	38.5	42.6	"	39.6	56.7
Average:			79.1	52.6	**60.7**	67.3	56.3	60.0	"	36.2	53.1
Std of average:			±13.6	±13.8	±3.7	±12.1	±22.5	±14.0	±0	±2.9	±3.1

TABLE 4.18: A comparison between the full letter models. *Recall is always 100% for the baseline (Section 3.6.2.1)

In Table 4.19 the results for the 5 test sets from the folds are combined and split in the performance per event category. For every event category, performance among the models is quite similar. However, performance between different event categories show fluctuations in performance: categories "fall" and "delirium" have lower f1 scores than "electrolyte disturbances" and "renal insufficiency". This might be linked to the distribution of positive and negative samples present within the classes, as the lower scoring categories contain relatively few positive examples. In the evaluation method for the full letter models, if a single instance that corresponds to an ADE is marked positive, the ADE is marked positive (Section 3.6.2. On average, there are ∼14 instances per ADE, there is a reasonable chance that (in a negative sample) one of these 14 results in a false positive, thereby marking the ADE as an ADR, despite the other 13 instances being correctly labelled as negative. This is also visible in Table 4.18 where the recall scores are much higher than the precision scores.

	Samples:		**RUS**			**R-BERT**			**Baseline 3**		
Event Category:	**Pos**	**Neg**	**R**	**P**	**f1**	**R**	**P**	**f1**	**R***	**P**	**f1**
Fall	26	78	73.1	29.2	41.8	17.6	23.1	20.0	100.0	25.0	40.0
Electrolyte Disturbances	26	24	88.5	57.5	70.0	96.2	56.8	71.4	"	52.0	68.4
Renal Insufficiency	34	14	88.2	71.4	79.0	82.3	66.7	73.7	"	70.8	82.9
Delirium	10	50	70.0	30.4	42.4	40.0	60.0	48.0	"	16.7	28.6

TABLE 4.19: A comparison between the full letter models for each of the event categories. *Recall is always 100% for the baseline (Section 3.6.2.1)

4.3.4 Full Letter models: Error Analysis

In total, the full letter models used 3712 text fragments (of up to 512 tokens) for training and testing. Because of this large amount of inputs that also contain long texts per input, it is difficult to go through all of these manually and come to unequivocal answers to as of why certain errors occur. We start by comparing the labelling of RUS and R-BERT, then we will take a closer look at the specific inputs.

We will take a look at the first fold, which contains 742 samples of which 363 are positive and 379 are negative. These samples correspond to 64 ADEs, of which 23 positive and 41 negative (Table 4.18). One thing that stands out is that RUS manages a more aggressive positive labelling strategy. RUS assigns 434 positive labels while R-BERT assigns 311 positive labels. There is a lot of overlap between both models, they agree on 536 (72.2%) labels.

Both models incorporate entity information on the trigger and event. In some cases the entity seems to largely influence the outcome. For example, "hydrochlorothiazide" is the marked trigger in 84 samples (in the test set of fold 1) and is positive in 32 of these cases. However, RUS and R-BERT both mark these as positive in 83 out of 84 cases. This is not always the case though, "temazepam" occurs in 51 samples and these are all negative, still, RUS assigns a positive label to 11 of these (R-BERT labels them all negative).

Example	**Fragment of input**
1	... Analyse geriatrie valpoli: evenwichtsstoornis wv balanscursus en medicatie optimalisatie. <e1>Collaps </e1>... simvastatine 1dd 40 mg, <e2>nortrilen </e2>25 mg 2dd1, atenolol 25 mg 1dd1, Calci-Chew 500 mg 1dd1 ...
2	... <e1>Duizeligheid </e1>. Afgelopen dagen geen klachten gehad. Hoest meer slijm op sinds enkele maanden. Pijn is begonnen na val ... Thuismedicatie Datum overzicht: 5-4- (Medicatie; Toedieningsweg; Schema; Zo nodig; Opmerking) <e2>Lisinopril </e2>tablet 20mg ; oraal; 1 x per dag 20 milligram. Metoprolol tablet mga 25mg (succinaat) (retard) ...

TABLE 4.20: Examples from the full letter model test sets in fold 1.

If we look at example 1 from Table 4.20, the input is marked negative by R-BERT, whereas the same input with "atenolol" marked instead of "nortrilen", results in a positive label. The same happens in example 2, when we change the marked trigger from "lisinopril" to "metoprolol", both models change their annotation from positive to negative. Here the two inputs are almost similar, except for the marked entities, which must therefore be the cause of the different label assignment.

While doing a manual exploration of the results in fold 1, it seems that entity information has a disproportionate weight on the outcomes of both models. Because the input texts are long, it is hard to detect what textual differences between inputs causes the models to assign different labels. It seems that the long texts contain too much noise, causing the models to rely too much on the entity information.

Chapter 5

Discussion and Future Work

The main goal of this thesis was to detect adverse drug reactions (ADRs) in Dutch admission and discharge letters of geriatric patients. To accomplish this goal, we formulated several sub-questions in Section 1.3. In this section we provide the answers to those sub-questions. First, we will go over concept extraction, followed by how this translates to adverse drug events (ADE) recognition. Then, we discuss relation extraction, ultimately followed by the relevance of this research to clinical practice.

5.1 Concept Extraction

In Section 1.3, our first formulated sub-question was: *How well can our concept extraction model identify concepts from Dutch admission and discharge letters of geriatric patients?*

Our model is able to achieve an f1 score of 91.4% on recognition of all the concepts. In previous research, where MedCAT, MetaMap, cTAKES, Bio-YODIE and SemEHR were applied on several UMLS concept extraction datasets, f1 scores varied between 17.8% and 73.8%, with the highest scores being achieved by MedCAT (Kraljevic et al., 2021). Our f1 score is significantly higher, mainly caused by the high f1 score in the recognition of pharmaceutical drugs (triggers).

We explored several options to achieve high recall scores. This answers the next question: *In what ways can we improve concept extraction?* First, the exploration of several admission and discharge letters (that were excluded thereafter) led to early identification of terms that are missing in the filtered concept database. The second step was to adapt parameters specifically to our use case. In our dataset "val" was a very common word and therefore setting minimum normalization length to "3" was necessary. Additionally, we introduced our own parameter "diacritics", to enable recognition of terms with diacritical marks, which are common in Dutch language and in our dataset. Our self-developed diacritics parameter increased recall on events by ≈20%, as words like "hyponatriëmie", "hypokaliëmie" and "hyperkaliëmie" are common in our dataset. Our "diacritics" parameter has now also been implemented into the official MedCAT tool.

It is also noticeable that triggers are significantly better recognized than events (93.3% vs 78.7%). We argue that the cause of this is twofold. First, ambiguity is rarely a problem in triggers because drug names are designed to be unambiguous. Secondly, triggers are nouns hence linguistically more simplistic. To improve our model, primary focus should be on boosting event recognition performance, as here is most to gain. There are several options that could improve clinical concept extraction on events even further. One is the expansion of concept databases with re-occurring ambiguous non-medical words. For example the Dutch term "valt op" (stands out) was often labelled as "valt" (falls) in our use case. Adding ambiguous words to the concept database before training MedCAT, allows MedCAT to distinguish such cases. Preferably such expansions would be implemented UMLS-wide, but for smaller use cases (such as ours) it is also achievable to do this manually. Finally, during error analysis we manually corrected concepts that were missed during the annotation process. This shows that the annotations can contain errors, reducing the number of annotation mistakes would be beneficial to our model.

5.2 ADE Extraction

Subsequently, the concepts can be used to extract ADEs from medical texts. *The question that follows from this: Can we use these identified concepts to recognize ADEs?*

In Noorda et al. (2022) admission letters were manually annotated for presence of ADEs. They did not annotate for all possible ADEs, but only for ADEs relating to 10 categories. We used MedCAT to automatically detect ADEs in the same admission letters and used their annotated dataset as golden standard. We focus on 4 of these ADE categories: fall, delirium, electrolyte disturbances and renal insufficiency and/or dehydration. At first, our model achieved a precision score of only 42.8% (and f1: 51.6%) in extraction of ADEs from admission letters. However, it turned out that many false positives were actually true positives. This means that our model recognizes cases that were missed during manual annotation. These cases show the added value of our model on top of manual annotation. After correcting for this, the precision score improved to 68.4% (and f1 to 71.4%). This is an excellent score, especially considering that a 100% f1 score was not achievable; some admission letters in our dataset missed crucial segments that contained mentions of the relevant triggers and events required for ADE recognition. We estimate the maximum achievable f1 score to be ∼85%.

Manually identifying ADEs is labour intensive and also causes missed ADEs as mentioned in the previous paragraph. We show that using a concept extraction tool for ADE recognition could be used complementary to human work. The first step to identify ADEs (both manually and automatically) is to compile a list of trigger-event combinations that form ADEs. In Noorda et al. (2022) they annotate for 10 ADE categories. Even though there are only 10 categories, already several ADEs are missed during manual annotation. This is because the annotators have to keep track of many concepts that potentially form ADEs, making the process prone to errors. This will be even more so when the dimensionality of the problem increases (when more ADE categories are included in the assessment). Because the mapping between ATC codes (drugs) and events need to be established only once, this methodology can be translated into an algorithm that detects ADEs automatically, thereby speeding up the process.

Our methodology means that detecting ADEs in texts can be automatized, removing the necessity to read through all texts carefully. However, not all steps can be automatized. First, defining ADEs by providing a mapping between ATC codes and events remains a manual task. Next, it is still necessary to define which words in texts correspond to ATC codes and events. Although instead of creating a list of words ourselves, we take a more generic approach by using existing ontologies, these are not exhaustive. This means that some of the former problems remain. One example is that in the original compilation of our ATC codes, an instance was missing. Additionally, the concepts that were derived from the events and ATC codes were not always well-represented in UMLS. Again, expansion of medical ontologies for Dutch language or access to existing databases (such as Farmacotherapeutisch Kompas, Zorginstituut Nederland, 2021) could prove useful for future work.

Our methodology also brings new problems as becomes apparent when we look at the false positive results. 72 (42.6%) of all 169 false positive errors were caused by negations (table 4.11), a few of these were solved by using the model provided by Es et al. (2022), but there is still room for improvement in our use case. The prescribed medication false positives (20 errors) can be subdivided into two problems with different solutions: 1) There is a section at the bottom of the letter with new prescriptions, the solution here would be to remove this part entirely. 2) It is mentioned in running text, at an arbitrary position in the letter. For this last one, using a model to detect temporality could be useful. ContextD is an algorithm that can do this in Dutch clinical texts (Afzal et al., 2014). Another option to reduce false positives is to remove the medical history sections (40 errors) while pre-processing, because these contain mentions of triggers and events that do not refer to present state. The disambiguation problem (20 errors) is also a common cause of false positives and a solution was already mentioned in previous section (Section 5.1). Lastly, conditional statements (17 errors) are a cause of mistakes. To our knowledge, there is no model that is capable of capturing this contextual property in Dutch clinical text.

5.3 Relation Extraction

The final step in our research is to determine whether an ADE is an ADR. In our dataset, all ADEs were labelled for their probability of being an ADR (table 3.3). We translated this into a binary problem, causality categories "unclassifiable" and "unlikely" as non-ADRs, "possible", "probable" and "certain" as ADRs. This raised the following question: *Can we use a belabBERT-based model for relation extraction or do we need to use less data intensive methods?*

We took two approaches to relation extraction. Our first approach was trained and tested on large input texts (up to 512 tokens). The advantage of this approach is that it may capture ADRs over long-range in text, as for each ADE we are combining every corresponding trigger-event combination within a letter, the trigger and event are combined including a context-span around each of them. So each input entry consists of a text fragment along with the marked trigger and event and is assigned a label (ADR or ADE). The label is based on whether the ADE is annotated as an ADR in the letter where the text fragment originates from. This may mean that a text fragment is annotated as an ADR, whereas this is not deducible from that specific fragment. Therefore, during evaluation we accumulate all results per ADE per letter and mark it as ADR if $\geq$1 of the text fragments is marked as ADR. We trained and tested two belabBERT-based models with this input, RUS and R-BERT. These models incorporate information on the entities themselves along with context to determine a relationship. These models are compared to a baseline model that marks all detected ADEs as ADRs. This resulted in a slight improvement of RUS over the baseline, 60.7% vs 53.1%, and R-BERT not being significantly better than baseline. Although the models seem to have learned something, these results are not very promising. One reason may be the way that labels are assigned and how output is evaluated. It would be better to determine the label for each of the input text fragments individually. For the output, an ADR label is assigned when just one ADE instance in the letter is marked as ADR. Alternatively, changing this to 10%, 20% or a majority vote may produce more solid outcomes. On top of that, by generating inputs up to 512 tokens, the models have to deal with a lot of noisy texts. A different and more effective strategy to capture long-range ADRs might be to reduce the input size. This can still be used to detect long-range dependencies, however the context-span of each trigger and event will contain less text.

In the second approach we use short text fragments, which explicitly show recognition of ADRs by clinicians in the clinical letters. These usual care models, use short input texts, consisting of only one or two sentences. With this approach, it is not possible to catch all ADRs in a letter, because these may rely on long-range dependencies. In the research by Noorda et al. (2022) they note that 16.5% of the ADRs are missed through recognition by usual care, and that only 68.2% of usual care instances translate to ADRs. The advantage of this approach is that we feed the model with only short texts which contain relatively much useful information. As a consequence, the results are much better than in the full letter models. Here, RUS significantly outperforms the baseline model with an f1 score of 76.9% (table 4.14). R-BERT also achieved decent performance, with an f1 score of 74.0%, although it did not significantly outperform baseline due to large performance fluctuations across folds. This points out a shortcoming in our research: a lack of train and test instances. We believe that most performance can be gained by increasing the amount of training samples. Now, we are only relying on 268 samples, but in other similar research these numbers are often several orders of magnitude larger. Text data augmentation could prove useful in this case. By switching terms for their synonyms and abbreviations (and vice versa) we can generate variants that increase the size of our dataset. Moreover, the models could also benefit from further cleaning the input texts as irrelevant text in the input is causing errors.

A rule-based method could be used to compete against the proposed usual care models. The samples in our usual care dataset contain patterns. Examples such as "Hyponatriëmie DD HCT" or "Milde hyponatriëmie geduid bij HCT" are common and could easily be matched using rules. A hybrid method, where entities are recognized using our methodology and then consequently the words in between the entities are tested against rules, could also prove useful. An advantage of rule-based is that it is less data intensive and that it does not have to rely on a knowledge base. Oppositely, if there is enough data at hand, a deep learning strategy may be the preferred option. Whereas a rule-based model certainly misses a never-seen pattern, a well-trained version of our usual care approach may well do it correct.

We were not able to evaluate how well the models generalize to other cases. Generalization here has two aspects: 1) Generalization on different texts, for example on letters from the surgery department. 2) Generalization on different ADE categories. The second one would require additional changes to our work: adding the relevant concepts to the concept database. Our relation extraction models are trained to incorporate entity information for determining ADRs. Therefore, generalization to new ADE categories without having samples to retrain on, may not be successful. However, retraining our relation extraction models and replacing all triggers and events by placeholders ("[trigger]" and "[event]" for e.g.) may result in a version that achieves decent performance when applied to unseen ADEs.

The next question we seek to answer is: *What is the best way of incorporating entity information for relation extraction?* In all our results, RUS outperforms R-BERT. In short, R-BERT generates three vectors: one for the whole input, one for the trigger and one for the event. These three are concatenated and used for the final prediction. This assigns a lot of weight to the marked entities. RUS changes the input by adding the marked entities in front of the input text separated by separator tokens. These separator tokens help BERT understand that these are different sentences and adds a learned embedding to every token that indicates to which sentence it belongs. In this case, the marked entities make up a smaller part of the vector. We argue, that in the R-BERT models, the vectors of the marked entities are too influential. As a result, the outcome is largely based on the marked entities. So when the same combination of triggers and events occur repetitively, R-BERT almost always assigns the same label. We have no hard evidence to support this claim, but our results may be indicative for this.

5.4 Clinical Practice

Finally, our last question: *What insights are gained about NLP in Dutch clinical practice?* First of all, our research shows that available Dutch medical ontologies (UMLS and SNOMED NL) are sufficiently developed to perform well at recognition of (a subset of) drugs and events in Dutch medical texts. We came across well-established issues such as negations and conditional statements, of which we emphasize the importance, but we also identified other pitfalls that are especially important when considering Dutch language, such as the usage of diacritics. Furthermore, we achieve promising results on ADR relation extraction while only having the availability of a small dataset. This shows the potential of these BERT-architectures together with the Dutch belabBERT model for ADR relation extraction in Dutch clinical texts. The main insight we gain here, is that future work in the medical NLP domain should first focus on providing larger annotated datasets, as this is the main bottleneck. Thereafter comes eliminating the false positives (negations, conditional statements, etc.) by tracking multiple contextual properties. Lastly, comes expanding the concept database or fine-tuning it to specific use cases.

Now we will discuss the relevance of this work towards clinical practice. We are able to reach f1 scores of 91.4% and 71.4% on concept and ADE recognition respectively. Therefore, we believe that MedCAT combined with our methodology is capable of capturing ADEs in clinical texts. For relation extraction, we believe our full letter models fall short. Performance improvement over baseline is minimal. In spite of that, our usual care models are promising. Picking the best usual care model, RUS, gives us an f1 score of 76.9% on determining recognition by usual care of two marked entities. There are however three downsides to our implementation of the usual care model. The first one is that the definition used by Noorda et al. (2022) for recognition by usual care is two-fold: 1) Explicit mentions in text 2) Drug withdrawal or adjustment in response to an event. We only acquired a subset of all possible usual care instances because we only keep track of the first part of the definition. Secondly, recognition by usual care does not necessarily imply an ADR. In the dataset by Noorda et al. (2022), 68.2% of all usual care instances are ADRs and 16.5% of all ADRs go undetected by usual care. Finally, we do not know if these numbers apply to our usual care dataset, as we gathered our own usual care instances. So although these numbers are presumably in the same range, we cannot know for sure. This means that, although our usual care models are promising, there remains uncertainty as to what extent this can be translated into ADR detection. Therefore, it would be useful to further investigate the relationship between recognition by usual care and ADRs.

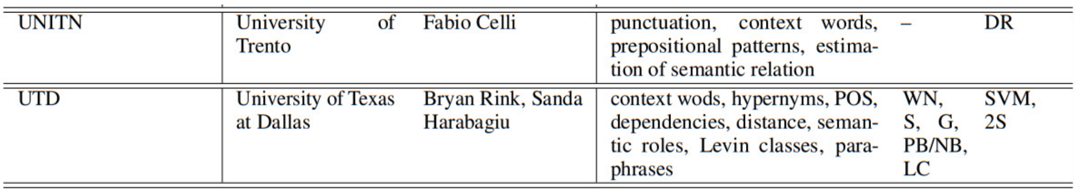
In practice, a pipeline that uses concept extraction, followed by relation extraction, could be used to detect ADRs in previous medical records. Deploying such a pipeline could help against under-identification of ADRs and can consequently lead to more data for ADR research. Under-reporting rates are currently so high (Hazell and Shakir, 2006), that arguably every recognized and reported ADR is a bonus. So even though our best model (RUS, usual care model) will not retrieve all ADRs, and not all predicted ADRs are correct, it can still be used to increase reporting rates for ADRs. It could also be implemented as an additional safeguard when prescribing medicines, that whenever a patient has had an ADR to the drug being prescribed, it triggers a warning. Such an implementation would require more evaluation, as this may directly influence patient treatment (for e.g. withdrawal of a drug).

In conclusion, to bring our best model (RUS, usual care model) into practice, a few steps have to be taken. First, we need to increase the size of our usual care dataset (currently 268 samples). This requires a larger extract of clinical letters/notes from the hospital database and additional manual annotation efforts by clinical pharmacologists. Secondly, recognition by usual care is not always a correct ADR, and also not every ADR is recognized by usual care. Therefore we need to further investigate the relationship between recognition by usual care and ADRs, to be able to further evaluate the performance. With these steps, this project can be further developed for clinical implementation.

Appendix A

A.1 SemEval-2010 Task 8 Models

System	Institution	Team	Description	Res.	Class.
Baseline	Task organizers		local context of 2 words only		BN
ECNU-SR-1	East China Normal University	Man Lan, Yuan Chen, Zhimin Zhou, Yu Xu	stem, POS, syntactic patterns	S	SVM (multi)
ECNU-SR-2,3			features like ECNU-SR-1, different prob. thresholds		SVM (binary)
ECNU-SR-4			stem, POS, syntactic patterns, hyponymy and meronymy relations	WN, S	SVM (multi)
ECNU-SR-5,6			features like ECNU-SR-4, different prob. thresholds		SVM (binary)
ECNU-SR-7			majority vote of ECNU-1,2,4,5		
FBK_IRST-6C32	Fondazione Bruno Kessler	Claudio Giuliano, Kateryna Tymoshenko	3-word window context features (word form, part of speech, orthography) + Cyc; parameter estimation by optimization on training set	Cyc	SVM
FBK_IRST-12C32			FBK_IRST-6C32 + distance features		
FBK_IRST-12VBC32			FBK_IRST-12C32 + verbs		
FBK_IRST-6CA, -12CA, -12VBCA			features as above, parameter estimation by cross-validation		
FBK_NK-RES1	Fondazione Bruno Kessler	Matteo Negri, Milen Kouylekov	collocations, glosses, semantic relations of nominals + context features	WN	BN
FBK_NK-RES 2,3,4			like FBK_NK-RES1 with different context windows and collocation cutoffs		
ISI	Information Sciences Institute, University of Southern California	Stephen Tratz	features from different resources, a noun compound relation system, and various feature related to capitalization, affixes, closed-class words	WN, RT, G	ME
ISTI-1,2	Istituto di scienca e tecnologie dell'informazione "A. Faedo"	Andrea Esuli, Diego Marcheggiani, Fabrizio Sebastiani	Boosting-based classification. Runs differ in their initialization.	WN	2S
JU	Jadavpur University	Santanu Pal, Partha Pakray, Dipankar Das, Sivaji Bandyopadhyay	Verbs, nouns, and prepositions; seed lists for semantic relations; parse features and NEs	WN, S	CRF
SEKA	Hungarian Academy of Sciences	Eszter Simon, Andras Kornai	Levin and Roget classes, n-grams; other grammatical and formal features	RT, LC	ME
TUD-base	Technische Universität Darmstadt	György Szarvas, Iryna Gurevych	word, POS n-grams, dependency path, distance	S	ME
TUD-wp			TUD-base + ESA semantic relatedness scores	+WP	
TUD-comb			TUD-base + own semantic relatedness scores	+WP,WN	
TUD-comb-threshold			TUD-comb with higher threshold for OTHER		



UNITN	University of Trento	Fabio Celli	punctuation, context words, prepositional patterns, estimation of semantic relation	–	DR
UTD	University of Texas at Dallas	Bryan Rink, Sanda Harabagiu	context wods, hypernyms, POS, dependencies, distance, semantic roles, Levin classes, paraphrases	WN, S, G, PB/NB, LC	SVM, 2S

FIGURE A.1: Overview of different participants and the used tools for their models (Hendrickx et al., 2010). WN: WordNet data; WP: Wikipedia data; S: syntax; LC: Levin classes; G: Google n-grams, RT: Roget's Thesaurus, PB/NB: PropBank/NomBank). Class: Classification style (ME: Maximum Entropy; BN: Bayes Net; DR: Decision Rules/Trees; CRF: Conditional Random Fields; 2S: two-step classification.

A.2 Event CUIs

Trigger Tool event categories	**Events**	**CUI**
2	Fall	C0085639
	Collaps / Syncope	C0039070
	Hypotension	C0020649
	Orthostatic Hypotension	C0020651
	Dizziness	C0012833
4	Hyponatraemia	C0020625
	Hypokalaemia	C0020621
	Hyperkalaemia	C0020461
5	Renal Insufficiency	C1565489
	Kidney Failure	C0035078
	Dehydration	C0011175
10	(Still) Delirium	C0011206
	Confusion	C0009676
	Drowsiness	C0013144

TABLE A.1: Overview of the UMLS CUIs linked to each of the 4 most prevalent event categories specified in the explicated ADR trigger tool.

Bibliography

Abacha, Asma Ben and Pierre Zweigenbaum (2010). “Automatic Extraction of semantic relations between medical entities: Application to the treatment relation.” In: *Semantic Mining in Biomedicine*.

Adel, Heike, Benjamin Roth, and Hinrich Schütze (2016). “Comparing convolutional neural networks to traditional models for slot filling”. In: *arXiv preprint arXiv:1603.05157*.

Afzal, Zubair, Ewoud Pons, Ning Kang, Miriam CJM Sturkenboom, Martijn J Schuemie, and Jan A Kors (2014). “ContextD: an algorithm to identify contextual properties of medical terms in a Dutch clinical corpus”. In: *BMC bioinformatics* 15.1, pp. 1–12.

Alimova, Ilseyar and Elena Tutubalina (2020). “Multiple features for clinical relation extraction: a machine learning approach”. In: *Journal of biomedical informatics* 103, p. 103382.

Alomar, Muaed, Ali M Tawfiq, Nageeb Hassan, and Subish Palaian (2020). “Post marketing surveillance of suspected adverse drug reactions through spontaneous reporting: Current status, challenges and the future”. In: *Therapeutic Advances in Drug Safety* 11, p. 2042098620938595.

Amisha, Paras Malik, Monika Pathania, and Vyas Kumar Rathaur (2019). “Overview of artificial intelligence in medicine”. In: *Journal of family medicine and primary care* 8.7, p. 2328.

Apache Software Foundation (2014). *openNLP Natural Language Processing Library*. URL: http://opennlp.apache.org/.

Aronson, Alan R (2006). “Metamap: Mapping text to the umls metathesaurus”. In: *Bethesda, MD: NLM, NIH, DHHS* 1, p. 26.

Aronson, Alan R and François-Michel Lang (2010). “An overview of MetaMap: historical perspective and recent advances”. In: *Journal of the American Medical Informatics Association* 17.3, pp. 229–236.

Banko, Michele and Oren Etzioni (2008). “The tradeoffs between open and traditional relation extraction”. In: *Proceedings of ACL-08: HLT*, pp. 28–36.

Baxt, William G (1991). “Use of an artificial neural network for the diagnosis of myocardial infarction”. In: *Annals of internal medicine* 115.11, pp. 843–848.

Beijer, HJM and CJ De Blaey (2002). “Hospitalisations caused by adverse drug reactions (ADR): a meta-analysis of observational studies”. In: *Pharmacy World and Science* 24.2, pp. 46–54.

Bodenreider, Olivier (2004). “The unified medical language system (UMLS): integrating biomedical terminology”. In: *Nucleic acids research* 32.suppl_1, pp. D267–D270.

Brandsen, A, A Dirkson, S Verberne, M Sappelli, D Manh Chu, and K Stoutjesdijk (2019). “BERT-NL a set of language models pre-trained on the Dutch SoNaR corpus”. In: *Dutch-Belgian Information Retrieval Conference (DIR 2019)*.

Bui, QC, P.M.A. Sloot, U. Lesser, C.A.B. Boucher, A.H.C. Kampen, M.T. Bubak, M. Rijke, and J.A. Kaandorp (2012). “Relation extraction methods for biomedical literature”. In: *UvA-DARE*.

Chapman, Alec B, Kelly S Peterson, Patrick R Alba, Scott L DuVall, and Olga V Patterson (2019). "Detecting adverse drug events with rapidly trained classification models". In: *Drug safety* 42.1, pp. 147–156.

Chen, Long, Yu Gu, Xin Ji, Zhiyong Sun, Haodan Li, Yuan Gao, and Yang Huang (2020). "Extracting medications and associated adverse drug events using a natural language processing system combining knowledge base and deep learning". In: *Journal of the American Medical Informatics Association* 27.1, pp. 56–64.

Chiu, Billy, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo (2016). "How to train good word embeddings for biomedical NLP". In: *Proceedings of the 15th workshop on biomedical natural language processing*, pp. 166–174.

Chowdhury, Gobinda G (2003). "Natural language processing". In: *Annual review of information science and technology* 37.1, pp. 51–89.

Cohen, KB (2013). "Biomedical natural language processing and text mining". In: *Methods in biomedical informatics: a pragmatic approach* 141.

Coloma, Preciosa M, Martijn J Schuemie, Gianluca Trifiro, Rosa Gini, Ron Herings, Julia Hippisley-Cox, Giampiero Mazzaglia, Carlo Giaquinto, Giovanni Corrao, Lars Pedersen, et al. (2011). "Combining electronic healthcare databases in Europe to allow for large-scale drug safety monitoring: the EU-ADR Project". In: *Pharmacoepidemiology and drug safety* 20.1, pp. 1–11.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2019). "Unsupervised cross-lingual representation learning at scale". In: *arXiv preprint arXiv:1911.02116*.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020). "RobBERT: a Dutch roBERTa-based language model". In: *arXiv preprint arXiv:2001.06286*.

Demner-Fushman, Dina, Willie J Rogers, and Alan R Aronson (2017). "MetaMap Lite: an evaluation of a new Java implementation of MetaMap". In: *Journal of the American Medical Informatics Association* 24.4, pp. 841–844.

Denny, Joshua C, Jeffrey D Smithers, Randolph A Miller, and Anderson Spickard III (2003). ""Understanding" medical school curriculum content using KnowledgeMap". In: *Journal of the American Medical Informatics Association* 10.4, pp. 351–362.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Doğan, Rezarta Islamaj, Robert Leaman, and Zhiyong Lu (2014). "NCBI disease corpus: a resource for disease name recognition and concept normalization". In: *Journal of biomedical informatics* 47, pp. 1–10.

Downs, Johnny, Sumithra Velupillai, Gkotsis George, Rachel Holden, Maxim Kikoler, Harry Dean, Andrea Fernandes, and Rina Dutta (2017). "Detection of suicidality in adolescents with autism spectrum disorders: developing a natural language processing approach for use in electronic health records". In: *AMIA annual symposium proceedings*. American Medical Informatics Association, p. 641.

Es, Bram van, M.P. Schraagen, S. Tan, M.M. Hemker, L. Reteig, S.R.S. Arends, M.A.R. Gaona, and S. Haitjema (2022). "Negation Detection of Dutch Clinical Texts". In: *Work in Progress*.

Federatie Medisch Specialisten (FMS) (Dec. 2020). *Richtlijnendatabase*. URL: https://richtlijnendatabase.nl/richtlijn/polyfarmacie_bij_ouderen/polyfarmacie_bij_ouderen_2e_lijn/medicatiegerelateerde_opname.html.

Friedman, Carol, Lyudmila Shagina, Yves Lussier, and George Hripcsak (2004). "Automated encoding of clinical documents based on natural language processing". In: *Journal of the American Medical Informatics Association* 11.5, pp. 392–402.

Fu, Sunyang, David Chen, Huan He, Sijia Liu, Sungrim Moon, Kevin J Peterson, Feichen Shen, Liwei Wang, Yanshan Wang, Andrew Wen, et al. (2020). "Clinical concept extraction: a methodology review". In: *Journal of Biomedical Informatics*, p. 103526.

Gautier, Sophie, Hélène Bachelet, Régis Bordet, and Jacques Caron (2003). "The cost of adverse drug reactions". In: *Expert opinion on pharmacotherapy* 4.3, pp. 319–326.

Gorrell, Genevieve, Xingyi Song, and Angus Roberts (2018). "Bio-yodie: A named entity linking system for biomedical text". In: *arXiv preprint arXiv:1811.04860*.

Gu, Yu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon (2021). "Domain-specific language model pretraining for biomedical natural language processing". In: *ACM Transactions on Computing for Healthcare (HEALTH)* 3.1, pp. 1–23.

Haerian, K, D Varn, S Vaidya, L Ena, HS Chase, and C Friedman (2012). "Detection of pharmacovigilance-related adverse events using electronic health records and automated methods". In: *Clinical Pharmacology & Therapeutics* 92.2, pp. 228–234.

Hahn, Udo and Michel Oleynik (2020). "Medical information extraction in the age of deep learning". In: *Yearbook of Medical Informatics* 29.1, p. 208.

Hakkarainen, Katja M, Khadidja Hedna, Max Petzold, and Staffan Hägg (2012). "Percentage of patients with preventable adverse drug reactions and preventability of adverse drug reactions–a meta-analysis". In: *PloS one* 7.3, e33236.

Härmark, Linda, Florence van Hunsel, and Birgitta Grundmark (2015). "ADR reporting by the general public: lessons learnt from the Dutch and Swedish systems". In: *Drug safety* 38.4, pp. 337–347.

Harpaz, Rave, Alison Callahan, Suzanne Tamang, Yen Low, David Odgers, Sam Finlayson, Kenneth Jung, Paea LePendu, and Nigam H Shah (2014). "Text mining for adverse drug events: the promise, challenges, and state of the art". In: *Drug safety* 37.10, pp. 777–790.

Hazell, Lorna and Saad AW Shakir (2006). "Under-reporting of adverse drug reactions". In: *Drug safety* 29.5, pp. 385–396.

Hendrickx, Iris, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz (July 2010). "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals". In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pp. 33–38. URL: https://aclanthology.org/S10-1006.

Henry, Sam, Kevin Buchan, Michele Filannino, Amber Stubbs, and Ozlem Uzuner (2020). "2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records". In: *Journal of the American Medical Informatics Association* 27.1, pp. 3–12.

Hooft, Cornelis S van der, Jeanne P Dieleman, Claire Siemes, Albert-Jan LHJ Aarnoudse, Katia MC Verhamme, Bruno HCH Stricker, and Miriam CJM Sturkenboom (2008). "Adverse drug reaction-related hospitalisations: a population-based cohort study". In: *Pharmacoepidemiology and drug safety* 17.4, pp. 365–371.

Huang, Chung-Chi and Zhiyong Lu (2016). "Community challenges in biomedical text mining over 10 years: success, failure and the future". In: *Briefings in bioinformatics* 17.1, pp. 132–144.

Humphreys, Betsy L, Donald AB Lindberg, Harold M Schoolman, and G Octo Barnett (1998). “The unified medical language system: an informatics research collaboration”. In: *Journal of the American Medical Informatics Association* 5.1, pp. 1–11.

Jackson, Richard, Ismail Kartoglu, Clive Stringer, Genevieve Gorrell, Angus Roberts, Xingyi Song, Honghan Wu, Asha Agrawal, Kenneth Lui, Tudor Groza, et al. (2018). “CogStack-experiences of deploying integrated information retrieval and extraction services in a large National Health Service Foundation Trust hospital”. In: *BMC medical informatics and decision making* 18.1, pp. 1–13.

Jagannatha, Abhyuday, Feifan Liu, Weisong Liu, and Hong Yu (2019). “Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (MADE 1.0)”. In: *Drug safety* 42.1, pp. 99–111.

Jiang, Jing and ChengXiang Zhai (2007). “A systematic exploration of the feature space for relation extraction”. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pp. 113–120.

Johnson, Alistair EW, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark (2016). “MIMIC-III, a freely accessible critical care database”. In: *Scientific data* 3.1, pp. 1–9.

Kaul, Vivek, Sarah Enslin, and Seth A Gross (2020). “The history of artificial intelligence in medicine”. In: *Gastrointestinal endoscopy*.

Kharrazi, Hadi, Laura J Anzaldi, Leilani Hernandez, Ashwini Davison, Cynthia M Boyd, Bruce Leff, Joe Kimura, and Jonathan P Weiner (2018). “The value of unstructured electronic health record data in geriatric syndrome case identification”. In: *Journal of the American Geriatrics Society* 66.8, pp. 1499–1507.

Kim, J-D, Tomoko Ohta, Yuka Tateisi, and Jun’ichi Tsujii (2003). “GENIA corpus—a semantically annotated corpus for bio-textmining”. In: *Bioinformatics* 19.suppl_1, pp. i180–i182.

Kraljevic, Zeljko, Thomas Searle, Anthony Shek, Lukasz Roguski, Kawsar Noor, Daniel Bean, Aurelie Mascio, Leilei Zhu, Amos A Folarin, Angus Roberts, et al. (2021). “Multi-domain Clinical Natural Language Processing with MedCAT: the Medical Concept Annotation Toolkit”. In: *Artificial Intelligence in Medicine*, p. 102083.

Lange Di Cesare, Kevin, Amal Zouaq, Michel Gagnon, and Ludovic Jean-Louis (2018). “A machine learning filter for the slot filling task”. In: *Information* 9.6, p. 133.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2020). “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinformatics* 36.4, pp. 1234–1240.

Liddy, Elizabeth D (2001). “Natural language processing”. In: *Encyclopedia of Library and Information Science* 2nd edition.

Liu, Haibin, Tom Christiansen, William A Baumgartner, and Karin Verspoor (2012). “BioLemmatizer: a lemmatization tool for morphological processing of biomedical text”. In: *Journal of biomedical semantics* 3.1, pp. 1–29.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). “RoBERTa: A robustly optimized BERT pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.

Lo, Kyle, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld (2019). “S2ORC: The semantic scholar open research corpus”. In: *arXiv preprint arXiv:1911.02782*.

Lowe, Henry J and G Octo Barnett (1994). “Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches”. In: *Jama* 271.14, pp. 1103–1108.

Luo, Yuan, William K Thompson, Timothy M Herr, Zexian Zeng, Mark A Berendsen, Siddhartha R Jonnalagadda, Matthew B Carson, and Justin Starren (2017). “Natural language processing for EHR-based pharmacovigilance: a structured review”. In: *Drug safety* 40.11, pp. 1075–1089.

Marques, Francisco Batel, Ana Penedones, Diogo Mendes, and Carlos Alves (2016). “A systematic review of observational studies evaluating costs of adverse drug reactions”. In: *ClinicoEconomics and outcomes research: CEOR* 8, p. 413.

Martin, Louis, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot (2019). “CamemBERT: a tasty french language model”. In: *arXiv preprint arXiv:1911.03894*.

McDonald, Clement J, Stanley M Huff, Jeffrey G Suico, Gilbert Hill, Dennis Leavelle, Raymond Aller, Arden Forrey, Kathy Mercer, Georges DeMoor, John Hook, et al. (2003). “LOINC, a universal standard for identifying laboratory observations: a 5-year update”. In: *Clinical chemistry* 49.4, pp. 624–633.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.

Mohit, Behrang (2014). “Named entity recognition”. In: *Natural language processing of semitic languages*. Springer, pp. 221–245.

Mosbach, Marius, Maksym Andriushchenko, and Dietrich Klakow (2020). “On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines”. In: *arXiv preprint arXiv:2006.04884*.

Mozzicato, Patricia (2009). “MedDRA”. In: *Pharmaceutical Medicine* 23.2, pp. 65–75.

National Library of Medicine (2020a). *UMLS Metathesaurus Vocabulary Documentation*. URL: https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html.

— (2020b). *UMLS Metathesaurus Vocabulary Documentation*. URL: https://www.nlm.nih.gov/research/umls/new_users/online_learning/Meta_005.html.

— (2021a). *National Library of Medicine - National Institutes of Health*. URL: https://www.nlm.nih.gov/.

— (June 2021b). *SPECIALIST Lexicon*. URL: https://data.lhncbc.nlm.nih.gov/lsg/lexicon/2021/release/LEX_DOC/DOCS/techrpt.pdf.

Neumann, Mark, Daniel King, Iz Beltagy, and Waleed Ammar (2019). “ScispaCy: Fast and robust models for biomedical natural language processing”. In: *Association for Computational Linguistics*.

Névéol, Aurélie, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum (2018). “Clinical natural language processing in languages other than English: opportunities and challenges”. In: *Journal of biomedical semantics* 9.1, pp. 1–13.

Ningthoujam, Dhanachandra, Shweta Yadav, Pushpak Bhattacharyya, and Asif Ekbal (2019). “Relation extraction between the clinical entities based on the shortest dependency path based LSTM”. In: *arXiv preprint arXiv:1903.09941*.

Noorda, Nikki MF, Bastiaan TGM Sallevelt, Wivien L Langendijk, Toine CG Egberts, Eugene P van Puijenbroek, Ingeborg Wilting, and Wilma Knol (2022). In: *Detection Of Adverse Drug Reactions In Older People With Polypharmacy Admitted To The Geriatric Ward Through The Emergency Department. Manuscript under review.*

Norvig, Peter (Feb. 2007). *Spelling Corrector*. URL: http://www.norvig.com/spell-correct.html.

Office of the National Coordinator for Health Information Technology (Sept. 2019). *What is an electronic health record (EHR)?* URL: https://www.healthit.gov/faq/what-electronic-health-record-ehr.

Ogren, Philip V, Guergana K Savova, Christopher G Chute, et al. (2008). "Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition." In: *LREC*. Vol. 8, pp. 3143–3150.

Ortiz Suárez, Pedro Javier, Laurent Romary, and Benoît Sagot (July 2020). "A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 1703–1714. URL: https://www.aclweb.org/anthology/2020.acl-main.156.

Paperswithcode (2021). *Papers with code - semeval-2010 task 8 benchmark (relation extraction)*. URL: https://paperswithcode.com/sota/relation-extraction-on-semeval-2010-task-8.

Patel, Tejas K and Parvati B Patel (2018). "Mortality among patients due to adverse drug reactions that lead to hospitalization: a meta-analysis". In: *European journal of clinical pharmacology* 74.6, pp. 819–832.

Patton, K and DC Borshoff (2018). "Adverse drug reactions". In: *Anaesthesia* 73, pp. 76–84.

Peek, Niels, Carlo Combi, Roque Marin, and Riccardo Bellazzi (2015). "Thirty years of artificial intelligence in medicine (AIME) conferences: A review of research themes". In: *Artificial intelligence in medicine* 65.1, pp. 61–73.

Perera, Sujan, Amit Sheth, Krishnaprasad Thirunarayan, Suhas Nair, and Neil Shah (2013). "Challenges in understanding clinical notes: Why nlp engines fall short and where background knowledge can help". In: *Proceedings of the 2013 international workshop on Data management & analytics for healthcare*, pp. 21–26.

Pires, Telmo, Eva Schlinger, and Dan Garrette (2019). "How multilingual is multilingual BERT?" In: *arXiv preprint arXiv:1906.01502*.

Reátegui, Ruth and Sylvie Ratté (2018). "Comparison of MetaMap and cTAKES for entity extraction in clinical notes". In: *BMC medical informatics and decision making* 18.3, pp. 13–19.

Richardson, Leonard (2007). "Beautiful soup documentation". In: *April*.

Routledge, Philip A, MS O'Mahony, and KW Woodhouse (2004). "Adverse drug reactions in elderly patients". In: *British journal of clinical pharmacology* 57.2, pp. 121–126.

Ruiter, Rikje, Loes E Visser, Eline M Rodenburg, Gianluca Trifiró, Gijsbertus Ziere, and Bruno H Stricker (2012). "Adverse drug reaction-related hospitalizations in persons aged 55 years and over". In: *Drugs & aging* 29.3, pp. 225–232.

Sahu, Sunil Kumar, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu (2016). "Relation extraction from clinical texts using domain invariant convolutional neural network". In: *arXiv preprint arXiv:1606.09370*.

Saluja, Sonali, Steffie Woolhandler, David U Himmelstein, David Bor, and Danny McCormick (2016). "Unsafe drugs were prescribed more than one hundred million times in the United States before being recalled". In: *International Journal of Health Services* 46.3, pp. 523–530.

Savova, Guergana K, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute (2010). "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications". In: *Journal of the American Medical Informatics Association* 17.5, pp. 507–513.

Sboev, Alexander, Sanna Sboeva, Ivan Moloshnikov, Artem Gryaznov, Roman Rybka, Alexander Naumov, Anton Selivanov, Gleb Rylkov, and Viacheslav Ilyin (2021). "An analysis of full-size Russian complexly NER labelled corpus of Internet user reviews on the drugs based on deep learning and language neural nets". In: *arXiv preprint arXiv:2105.00059*.

Sboev, Alexander, Anton Selivanov, Ivan Moloshnikov, Roman Rybka, Artem Gryaznov, Sanna Sboeva, and Gleb Rylkov (2022). "Extraction of the Relations among Significant Pharmacological Entities in Russian-Language Reviews of Internet Users on Medications". In: *Big Data and Cognitive Computing* 6.1, p. 10.

Searle, Thomas, Zeljko Kraljevic, Rebecca Bendayan, Daniel Bean, and Richard Dobson (2019). "MedCATTrainer: a biomedical free text annotation interface with active learning and research use case specific customisation". In: *arXiv preprint arXiv:1907.07322*.

Sessa, Maurizio, Abdul Rauf Khan, David Liang, Morten Andersen, and Murat Kulahci (2020). "Artificial Intelligence in Pharmacoepidemiology: A Systematic Review. Part 1—Overview of Knowledge Discovery Techniques in Artificial Intelligence". In: *Frontiers in pharmacology* 11, p. 1028.

Si, Yuqi, Jingqi Wang, Hua Xu, and Kirk Roberts (2019). "Enhancing clinical concept extraction with contextual embeddings". In: *Journal of the American Medical Informatics Association* 26.11, pp. 1297–1304.

SNOMED (1993). *The ICD-10 classification of mental and behavioural disorders*. URL: https://www.snomed.org/resources/resources.

— (2021). *Resources*. URL: https://www.snomed.org/resources/resources.

spaCy (2021a). *Dutch · spaCy Models Documentation*. URL: https://spacy.io/models/nl.

— (2021b). *spaCy Models Documentation*. URL: https://spacy.io.

Tang, Yixuan, Jisong Yang, Pei San Ang, Sreemanee Raaj Dorajoo, Belinda Foo, Sally Soh, Siew Har Tan, Mun Yee Tham, Qing Ye, Lynette Shek, et al. (2019). "Detecting adverse drug reactions in discharge summaries of electronic medical records using Readpeer". In: *International journal of medical informatics* 128, pp. 62–70.

USA Food and Drug Administration (FDA) (Apr. 2018). *Step 3: Clinical Research*. URL: https://www.fda.gov/patients/drug-development-process/step-3-clinical-research.

Uzuner, Özlem (2009). "Recognizing obesity and comorbidities in sparse data". In: *Journal of the American Medical Informatics Association* 16.4, pp. 561–570.

Uzuner, Özlem, Brett R South, Shuying Shen, and Scott L DuVall (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.

Vries, Wietse de, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019). "Bertje: A Dutch BERT model". In: *arXiv preprint arXiv:1912.09582*.

Wang, Yanshan, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu (2018). "A comparison of word embeddings for the biomedical natural language processing". In: *Journal of biomedical informatics* 87, pp. 12–20.

Wang, Zhiwei, Yao Ma, Zitao Liu, and Jiliang Tang (2019). "R-transformer: Recurrent neural network enhanced transformer". In: *arXiv preprint arXiv:1907.05572*.

Warlé-van Herwaarden, Margaretha F, Vera E Valkhoff, Ron MC Herings, Marjolein Engelkes, Jan C van Blijderveen, Eline M Rodenburg, Sandra de Bie, Jelmer Alsma, Caroline van de Steeg-Gompel, Cornelis Kramers, et al. (2015). "Quick assessment of drug-related admissions over time (QUADRAT study)". In: *Pharmacoepidemiology and drug safety* 24.5, pp. 495–503.

Warlé-van Herwaarden, MF, C Kramers, MC Sturkenboom, PM Van den Bemt, and PA De Smet (2014). *Targeting outpatient drug safety: recommendations of the Dutch Harm-Wrestling Task Force. Ministry of Health, Welfare and Sport: Den Haag, 2009*.

World Health Organization (2007). *Management sciences for health and world health organization: drug and therapeutics committee training course*. URL: https://www.who.int/workforcealliance/knowledge/technicalbrief/en/.

— (2009). *The use of The WHO-UMC system for standardised Case causality assessment*. URL: https://www.who.int/publications/m/item/WHO-causality-assessment.

— (2021). *ATC classification index with DDDs*. URL: https://www.whocc.no/atc_ddd_index_and_guidelines/atc_ddd_index/.

Wouts, Joppe, Janna de Boer, Alban Voppel, Sanne Brederoo, Sander van Splunter, and Iris Sommer (2021). "belabBERT: a Dutch RoBERTa-based language model applied to psychiatric classification". In: *arXiv preprint arXiv:2106.01091*.

Wu, Honghan, Giulia Toti, Katherine I Morley, Zina M Ibrahim, Amos Folarin, Richard Jackson, Ismail Kartoglu, Asha Agrawal, Clive Stringer, Darren Gale, et al. (2018). "SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research". In: *Journal of the American Medical Informatics Association* 25.5, pp. 530–537.

Wu, Ruidong, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun (2019). "Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 219–228.

Wu, Shanchan and Yifan He (2019). "Enriching pre-trained language model with entity information for relation classification". In: *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2361–2364.

Xu, Hua, Zhenming Fu, Anushi Shah, Yukun Chen, Neeraja B Peterson, Qingxia Chen, Subramani Mani, Mia A Levy, Qi Dai, and Josh C Denny (2011). "Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases". In: *AMIA Annual Symposium Proceedings*. Vol. 2011. American Medical Informatics Association, p. 1564.

Yang, Liu, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork (2020). "Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1725–1734.

Yates, Andrew, Nazli Goharian, and Ophir Frieder (2015). "Extracting adverse drug reactions from social media". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1.

Zorginstituut Nederland (Sept. 2021). URL: https://www.farmacotherapeutischkompas.nl/.