

A Case Study on Pros and Cons of Regular Expression Detection and Dependency Parsing for Negation Extraction from German Medical Documents. Technical Report

Hans-Jürgen Profitlich, Daniel Sonntag

*German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
and Oldenburg University*

Abstract

We describe our work on information extraction in medical documents written in German, especially detecting negations using an architecture based on the UIMA pipeline. Based on our previous work on software modules to cover medical concepts like diagnoses, examinations, etc. we employ a version of the NegEx regular expression algorithm with a large set of triggers as a baseline. We show how a significantly smaller trigger set is sufficient to achieve similar results, in order to reduce adaptation times to new text types. We elaborate on the question whether dependency parsing (based on the Stanford CoreNLP model) is a good alternative and describe the potentials and shortcomings of both approaches.

Keywords: information extraction (IE); negation detection, regular expression detection, natural language processing; dependency parsing; electronic health record (EHR)

1. Introduction

As medical records may cover a very long history of diseases (up to several decades) and include a vast number of diagnoses, symptoms, results, medications, and laboratory values, we could highly benefit from advanced search capabilities in clinical information systems to allow for the retrieval of relevant data. The first of several steps to process such data is information extraction. The fact that medical texts often contain descriptions of examinations and therapies leads to frequently occurring negated expressions which play a central role

Email addresses: profitlich@dfki.de (Hans-Jürgen Profitlich), sonntag@dfki.de (Daniel Sonntag)

in understanding a patient’s medical history. For example, a biopsy may include various specific tests and the results state whether a test was positive or negative (“edema, no metastases”).

In comparison to many other text types, electronic health reports, discharge or transfer letters and other kinds of medical reports are often written in a rather telegraphic style with compacted information. In addition, the detection of negation scope in German language is more difficult than in other languages, such as English (cf. for example [1]). Similar to English, German negations can be directly bound to a word as prefix like in “unauffällig” (unremarkable) or as suffix like in “schmerzlos” (painless). However, German texts differ in various aspects from English texts. For example, German is a richly inflected language (e.g., “no” can be translated as “kein”, “keine”, “keiner” etc.). Furthermore, German includes discontinuous or surrounding triggers, such as “weder ... noch ...” (neither ... nor ...). Another problem is that in most cases these triggers can be reduced and use a different word stem or preposition, like in case of “wies ... zurück” (rejected) and “wies ... nach” (verify), which can easily lead to too general trigger expressions when using for example regular expressions. Triggers may also both precede and follow the negated expression.

This work describes a software system realized within a large medical project about testing AI components (pAItient—developing, testing and evidence based evaluation of clinical value). The task is to extract medical information from unstructured German texts like findings or dismissal letters. We start by implementing an information extraction module to segment the sentences and cover mentioned medical concepts like diagnoses, disorders, or examinations based on some dictionary annotators.

To recognize negations we integrate and test an implementation of the NegEx algorithm [2] which leads to a good baseline, but also reveals some shortcomings described in this paper. We tested and compared our system to the system described in [3]¹. After studying the errors we were able to establish some improvements by optimizing the code and adjusting the configuration. As the configuration of NegEx comprises mainly the trigger set and the size of the window, we conducted experiments regarding an easier set of triggers and different scopes. As a result, we describe how the trigger set can be reduced to a small number of patterns (56 instead of 506 triggers) without decreasing quality, to make such approaches more practical and applicable in changing text extraction contexts.

In the rest of this study we show, first, that the trigger set can be reduced to a very small number of patterns without decreasing the classification quality (on our real world data set). Regular expressions as used in NegEX rely solely on surface text, and thus are limited when attempting to capture complex syntactic constructions such as long noun phrases. Second, to overcome the inherent shortcomings of this methods processing the surface text only, we add several natural language processing (NLP) modules including a dependency

¹We want to thank the authors for providing the data for this study.

parser, which is often mentioned in the literature (e.g., in [3]) when dealing with these problems. We evaluate several NLP modules including the Stanford CoreNLP dependency parser. In this case study we describe details of our implementation and the pros and cons of both approaches (regular expression detection vs. dependency parsing). For example, we show why dependency parsing is neither a good alternative nor a satisfactory complement for our type of medical documents. Deep syntactic processing turned out to be helpful in some cases, but implementing it is very time-consuming and entails other risks and problems. We then share some insights about the structure of trigger sets and the scope of triggers based on regular expressions.

2. Related Work and Background

Clinical information extraction from patient records is still underrepresented and underdeveloped in clinical settings. The identification of semantic relations, such as substance A treats disease B, remains a non-trivial task [4]. A special trend becomes apparent, the need for ontology modeling of medical terminology and corresponding information extraction results [5]. Because of enormous annotation costs, mainly unsupervised methods are being used [6]. Only recently, new text mining approaches on medical literature have been proposed, for extracting adverse drug events from text [7] or automatic symptom extraction on rare diseases [8] for example.

Other research has been dedicated to clinical negation detection together with the detection of pathological entities in German texts. Bretschneider et al. [9] classify sentences containing pathological and non-pathological findings in German radiology reports. Their approach uses syntacto-semantic parsing methods. Gros and Stede [10] present Negtopus, a system that identifies negations and their scope in medical diagnoses written in German and in English.

Even though many NLP applications have been developed recently that successfully extract implicit facts mentioned in medical reports, discriminating between positive, negative, and uncertain findings remains challenging [11, 12, 13, 14].

Rule-based systems rely on negation keywords and rules to determine the negation scope [15]. There are two main approaches: 1) directly on surface text or 2) based on examining the structures resulting from natural language parsing. Research on processing negation has been carried out mostly on clinical reports, and has focused on detecting whether a medical term is negated or not. Various approaches have been published for detecting negations and speculations in English medical reports. A widely used and popular tool is NegEx [2, 16, 17]. The approach uses a simple algorithm based on a list of triggers (or simple regular expressions) that indicate negation and speculation. A hard-coded window of words preceding or following a trigger is used to determine the scope of a negation or a speculation (uncertainties or equivocal findings like “cannot be excluded”). The original NegEx algorithm with hard-coded scopes was extended to ConText by Harkema et al. [18] by extending the scope of a negating term up to the end of a sentence or a termination term (called “conjunction”)

like “but”. Additionally, there are left-looking trigger terms that scope to the beginning of a sentence. Obviously, the results still depend very much on the used window size and the type of the texts.

NegEx has also been adapted to French, Swedish, Spanish, German and other languages with good results [19, 20, 21, 22, 23]. Approaches regarding the negation scope delimitation has been done by [24] for texts in Swedish. Beside NegEx a wide range of other methods exist, for example based on syntactic techniques [25, 26, 21] or machine learning techniques [27]. The i2b2 challenges also include contests for assertion classification which show very good results ([28, 29]).

However, these systems have two major drawbacks: 1) they rely on the lexical structure of sentences and 2) the scope of a negation is somehow arbitrarily fixed, i.e., they cannot cope with the problem of long distance information between negation keywords and the target. For example, a pattern covering an expression like “no signs of an infection” does not match the phrase “no detectable signs of a new infection”. A number of systems tried to enhance the results of these approaches based on surface text and to overcome their shortcomings by additionally analyzing the syntactic structure of sentences, most notably dependency graph, the associated set of directed binary grammatical relations that hold among the words. Relations among the words are illustrated in our figures and medical example sentences with directed, labeled arcs from heads to dependents. We call this a typed dependency structure in computational linguistics.²

Sohn et al [30] used regular expressions on the dependency path and Mehrabi et al [31, 32] used dependency patterns as a post-processing step after NegEx to remove false positives. A similar approach can be found by Savova et al [33] who realized the text analysis and knowledge extraction system cTAKES. Ballesteros et al [34] also tried to infer the scope of negations by using a dependency parser. Huang and Lowe [26] proposed a classification scheme of negations based on syntactic categories and patterns in order to locate negated concepts, regardless of their distance from the negation cue. Weng et al [35] are using NegEx and the parsing tree, but also concept identification using metathesaurus, and parsing tree pruning to capture important parts for negation. [36] and [37] also use the dependency parsing for negation detection and propose a relatively comprehensive rule set (but only for English).

3. Pipeline Implementation

In our system, information extraction is performed using UIMA (Unstructured information management architecture) [38], a framework of software systems for analyzing large volumes of unstructured information. UIMA is a widely

²Please have a look at Daniel Jurafsky’s and James H. Martin’s informative chapter in *Speech and Language Processing* (forthcoming), see <https://web.stanford.edu/~jurafsky/sl/p3/14.pdf>

used architecture in NLP systems because of its support for modular combination of components, reuse of code, and interoperability. We employ the pipeline to extract medical data about diagnoses, symptoms, medications, and laboratory values from the documents. The central data structure is a so-called CAS object (common analysis structure), which is guided through an analysis pipeline. The modules of this pipeline, called analysis engines (AE), can access annotations of previous modules and enrich the CAS with additional annotations.

We use implementations of open source libraries where applicable and appropriate, mainly the DKPro Core [39]³, a large collection of open source software components for NLP based on the Apache UIMA framework. Several of these DKPro analysis engines are based on the Stanford NLP package⁴[40]. In some cases we adapt and extend the DKPro code due to our requirements.

Our UIMA pipeline starts with some basic preprocessing steps to annotate sentences, tokens, and stop words (own implementations if not marked otherwise), as below:

- CoreNlpSegmenter: (DKPro) the Stanford NLP Segmenter to split the input text into sentences and tokens;
- PatternBasedSentenceSegmenter, PatternBasedTokenSegmenter: (DKPro) used to split sentences and tokens further by regex-patterns;
- StopwordMarker: annotates stopwords like “is”, or “and”, i.e., words that can be ignored for our purposes. It is based on a manually assembled list of words.

Additionally there are further modules to decompose German compounds and to annotate medical concepts like diagnosis, disorders, etc.

- CompoundSplitter: splits German compounds into its components (based on the code of D. Naber⁵);
- DictionaryAnnotator(s): adaption of a DKPro implementation of a dictionary annotator; for diagnoses, disorders, symptoms, examinations, etc.

Modules needed for the additional dependency parsing:

- CoreNlpPosTagger: the Stanford NLP POS (part-of-speech) tagger;
- IxaLemmatizer: (DKPro) IXA Lemmatizer (the Stanford NLP Lemmatizer does not support German);
- CoreNlpParser: Stanford NLP parser;

³<https://dkpro.github.io/dkpro-core/>

⁴<https://stanfordnlp.github.io/CoreNLP/>

⁵<https://github.com/danielnaber/jwordsplitter>

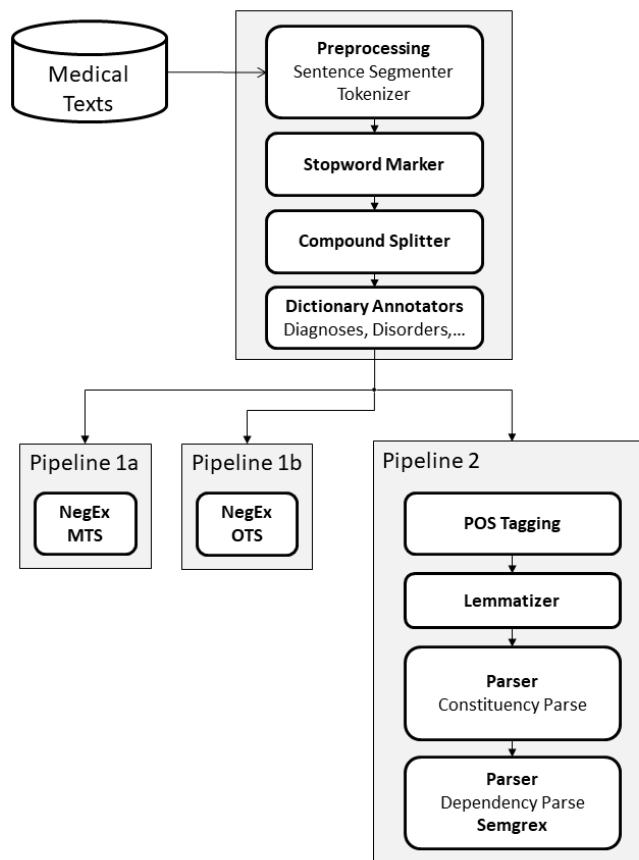


Figure 1: UIMA Pipeline showing the common starting block of analysis engines up to the dictionary annotators and the following negation detection variants (Pipeline 1a, 1b and 2).

- CoreNlpDependencyParser: adaptation of the Stanford NLP dependency tagger, integrates a module to execute Semgrex patterns on the semantic graph.

3.1. Pipelines 1a and 1b: Negation Extraction

NegEx is a widely used algorithm that utilizes regular expressions (called triggers) to detect negations. Triggers come in different versions depending on their position relative to the negated term: the main types are pre-negations, post-negations, and pseudo-negations. Pseudo-negations are used to avoid annotating a negation when a pre-negation trigger refers to a word that is not a medical finding, like in “keine Änderung ...” (no change) or “kein Anstieg ...” (no increase). Conjunctions are used to terminate the scope of a trigger. We aimed at comparing our results to the ones described in Cotik et al. [3] (referred

as the MACSS project) which used a Python implementation of NegEx. Their implementation does not use regular expression triggers for German texts but plain strings. They translated the original trigger set of Chapman et al. and expanded it which results in a set of 506 “MACSS” triggers⁶. These MACSS triggers use long expressions covering variations of wordings like “does not apply to the patient”, “did not apply to him”, “did not apply to her”, which makes them hard to handle and maintain. We tested whether it’s sufficient to use the actual negation parts only (“not” in these cases) without losing precision by the resulting greater distance between trigger and negated term. This results in a trigger set of only 56 items, a reduction by 88% (see also section 5.1).

Our own implementation of NegEx is based on the Java software of the Google code archive⁷. It had to be extended, as it originally just returned the negated text part but not the negating expression which is necessary to validate the results.

In contrast to the MACSS implementation we did not include the annotation of uncertainties (like “not sure” for example), as medical statements and wordings mostly avoid explicit unique expressions anyway. Instead, often phrases like “cannot be ruled out”, “no signs of” are used, which always leave some remaining uncertainty. Furthermore the MACSS trigger set contains only 13 triggers for uncertain negations and the only one that occurred in the example texts was the question mark.

The analysis engine that wraps the NegEx algorithm can be applied as a trailing engine to the main pipeline described above (see pipelines 1a/b in figure 1) and can be called with both the MACSS trigger set called “MTS” (pipeline 1a) or our own set (“OTS”, pipeline 1b).

3.2. Pipeline 2: Dependency Parsing

As a second extension of the pipeline (see pipeline 2 in figure 1) we integrated some natural language modules from the Stanford NLP project (or rather their DKPro implementation): part-of-speech tagger, lemmatizer, parser, and dependency parser.

The Stanford NLP framework includes a mechanism to match patterns against the dependency trees of sentences, called Semgrep⁸. We implemented a module based on the Semgrep mechanism [41]⁹ to match negation triggers and the corresponding negated tokens from the dependency graphs. The Semgrep mechanism works directly on the Stanford type system which made it necessary to integrate our module into the DKPro implementation of the dependency parser. A separate analysis engine would have been preferred but the trees are not stored in the data structure of the pipeline and there is no implementation

⁶This trigger set can be found here: http://macss.dfki.de/negex/negex_trigger_german_biotxtm_2016.txt

⁷<https://code.google.com/archive/p/negex/>

⁸There are similar modules to match sequences of tokens (TokenRegex) or partial trees (Tregex)

⁹<https://nlp.stanford.edu/software/tregex.html>

available that re-constructs a semantic graph out of the annotations in German. The module uses a set of patterns defining various negations phrases.

A simple Semgrep pattern defines two nodes and some relation between them¹⁰. A node is represented by a set of attributes and their values contained by curly braces: `{attr1:value1;attr2:value2;...}`, for example `{pos:/NN/}` refers to a node whose part-of-speech annotation is “NN”. Attributes must be plain strings, common used attributes are “pos” (part-of-speech), “lemma”, or “word” (the word itself in lowercase characters). Values can be strings or regular expressions blocked off by “/”. `{}` represents any node in the graph. A node can be given a name by appending “=*name*” to the node specification which allows us to access specific nodes after a match. Relations are defined by a symbol representing the direction of the relationship and a string or regular expression representing its value, e.g. “> /nmod/”. The symbols “<” and “>” indicate the direction of the relation, and “>>” stands for a relation chain. We defined two types of fixed names: “gov” depicts the governor of a negation (the negating expression) and names starting with “dep” link to all dependents (the negated expressions). These names are used by the analysis engine to annotate the named tokens when a pattern matches.

As we realized two methods to detect negations (NegEx and dependency parsing), the patterns can be used for different purposes: 1) detect negations that NegEx has missed or 2) correct annotations falsely marked as negations by NegEx.

This leads to (at least) two types of patterns:

- patterns to mark negated terms: confirmation or supplement of annotations by NegEx
- patterns to mark positive terms: correction of false annotations by NegEx, e.g. “nicht ausgeschlossen” (not excluded)

Patterns are read from resource files where they are marked with the according type flags. Some examples of Semgrep patterns and their meaning are given in table 1.

3.3. User Interface

Based on the brat visualisation software¹¹ [42] we implemented a web workbench that enables the user to enter both sample sentences and Semgrep patterns, and which presents the results of predefined patterns plus the matches of the interactively entered pattern. This is of great help when specifying and tuning new Semgrep patterns. Additionally, the web frontend allows us to directly compare the results of the UIMA pipeline, especially the negations found by the NegEx modules with the matches of Semgrep patterns. Figure 2 shows the different parts of this interface.

¹⁰A more detailed description can be found here: <https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/semgraph/semgrep/SemgrepPattern.html>

¹¹<http://brat.nlplab.org>

Pattern	<code>{lemma:/frei/}=gov > /nmod:von/ {}=dep</code>
Description	A node with lemma = ‘frei’ (<i>free</i> , referenced as ‘gov’) is in relation of type ‘nmod:von’ to another node (referenced as ‘dep’)
Example	<i>Patient frei von Schmerzen</i> (patient free of pain)
Pattern	<code>!{lemma:/anzeichen hinweis/}=dep > /neg/ {}=gov</code>
Description	A node whose lemma is not ‘anzeichen’ or ‘hinweis’ (<i>signs/hints</i> , referenced as ‘dep’) is in relation of type ‘neg’ to another node (referenced as ‘gov’)
Example	<i>Keine Infektion erkennbar</i> (no infection recognizable) but not matching <i>Keine Anzeichen einer Besserung</i> (no signs of improvement)
Pattern	<code>{pos:/NN/}=dep < /. *subj.*/ ({word:/ausgeschlossen auszuschlie.*en/}=gov !>> /neg/ {})</code>
Description	A node whose ‘pos’ is not ‘NN’ (referenced as ‘dep’) is the target of a relation of type ‘subj’ (including variants like ‘nsubj’) to another node (referenced as ‘gov’) whose word (text) is ‘ausgeschlossen’ (<i>excluded</i>) and there is no relation chain containing a relation of type ‘neg’ to another node.
Example	The part ‘!>> /neg/ {}’ ensures that ‘ausgeschlossen’ is not negated. <i>Fieber ist ausgeschlossen</i> (fever is excluded) but NOT matching <i>Fieber ist nicht völlig ausgeschlossen</i> (fever is not excluded completely). This would be an example of a <i>double-negation</i> .

Table 1: Some examples of Semgrep patterns: ‘gov’ is the governor (the *negating* term) and ‘dep’ is the dependent (the *negated* term) of the negation. ‘pos’ stands for part-of-speech, ‘NN’ for a noun.

Med NLP

— Text to annotate —
 Der Patient hat kein Fieber.

— Annotations —
 named entities x NegEx MACSS triggers x dependency parse x

— Language —
 German Submit

NER + NegEx:
 1 | Der Patient hat kein Fieber .
 -Disorder

Enhanced++ Dependencies:
 1 | Der Patient hat kein Fieber .
 det nsubj neg dobj neg punct
 Der Patient hat kein Fieber .

CoreNLP Tools:
 Semgrep

Enter a Semgrep expression to run against the "enhanced dependencies" above:
 1 | (lemma:/anzeichen)=dep > /neg/ !=gov

Additional matches from predefined expressions
 1 | Der Patient hat kein Fieber .
 gov neg

label	pattern	matches (gov → dep)
1 N1	!(lemma:/anzeichen hinweis nachweis verdacht anhalt)!=dep > /neg/ !=gov	hat kein → Fieber

Visualisation provided using the brat visualisation/annotation software.

Figure 2: Web-based user interface. A: text input field plus menus to determine language and annotations, B: results of named entity detection: negated terms are marked with a “-” sign, C: result of dependency parsing, D: input field for an interactive Semgrep pattern, E: results of the interactive Semgrep pattern, F: additional results of predefined Semgrep patterns

4. Evaluation Data

The evaluation data originates from the MACSS project and is the same as described in Cotik et al. [3]. It has been provided by Charité Berlin and comprises of two types of documents from the nephrology domain: discharge summaries and clinical notes. Both types are written by medical doctors during their daily routine, but have significant differences: the clinical notes are rather short and are written shortly after a visit of a patient. Discharge summaries are written to summarize a stay at the hospital and are more structured.

It should be mentioned that the original form of the document text was not available for evaluation. Instead, all documents are provided in a format where each line contains exactly one medical concept, followed by the whole sentence and a trailing tag that can be either “Affirmed” or “Negated”. This is important as what is regarded as a “sentence” was determined by an algorithm, which splits a document into sentences according to a specific delimiter patterns like a punctuation mark (.) or a double newline for example. As the medical texts contain enumerations in a tabular like style (without punctuation marks), the resulting sentence annotations are sometimes very long (and often longer than a human reader would tell when parsing the document structure). This is important to note as the NegEx algorithm assigns the scope of negations within whole sentences which is more error-prone when (natural) sentence boundaries are not detected. In Cotik et al. the annotations have been made manually by two annotators. Some smaller inconsistencies remained as sometimes in noun phrases like “acute anemia” the complete phrase was marked and in other cases just the word “anemia”. Additionally, sometimes a correct annotation in a sentence was completely missing. We corrected these inconsistencies in the ground truth documents. Beside this, we ignored annotations originally marked as “speculated” as we do not annotate them. (This led to the effect that our results differ a bit from the results originally reported by Cotik et al., the systems however remain comparable in a quantitative evaluation with the corrected dataset.) These documents served as ground truth for our evaluation.¹²

Our ground truth data set with corrected annotations has been split (as also usual in machine learning experiments) in a learning dataset (set 1) and a testing dataset (set 2). Table 2 provides an overview of the learning set (called set 1). Table 3 shows the number of findings (annotations) that are affirmed and the number of negated terms in the documents serving as training set (after completing the annotation). Here, training means the manual iterative extraction rule generation process for a specific pipeline. For example, set 1 used to improve our NegEx algorithm and to tune the trigger sets (mainly by manually checking the false positives and false negatives and refining NegEx rules). Additional documents (also from the MACSS project) have been made available

¹²The data format does not specify the exact positions of the medical concept or the negation term. This means it is often not clear which occurrence of a concept is referred to if the same term occurs more than once in a sentence, or which negation term is related to a specific concept (sentences often contain multiple medical concepts and negation terms).

	discharge summaries	clinical notes
# documents	8	175
total words	6221	6674
total sentences	1076	1158
avg. words per doc.	777.63	38.14

Table 2: Statistics about the annotated documents (training set 1)

	discharge summaries	clinical notes
affirmed	409	284
negated	108	338
findings	517	622

Table 3: Number of affirmed and negated terms in set 1

that were used solely for evaluation (referred to as set 2). Set 2 comprises of 45 discharge summaries and 175 clinical notes (see table 4).

5. Experiments

Our original pipeline was designed to recognize different types of medical concepts like diagnoses, disorders, or examinations for example. These dictionary annotators are trained based on terminology sources such as ICD-10¹³. However, the documents used as a ground truth were annotated (manually by human annotators) based on different sources. In order to be able to compare the results we adapted the analysis engines of our pipeline: instead of searching for diagnoses, examinations etc. with different annotators we defined one special “med_concept” annotator that uses exactly the set of medical concepts collected from the training set. In future investigations, we will use ontology matching to SNOMED¹⁴, the standard used in the pAItient project over the

¹³<https://icd.who.int/browse10/2016/>

¹⁴<https://www.snomed.org>

	discharge summaries	clinical notes
affirmed	2421	261
negated	756	306
findings	3177	567

Table 4: Statistics about the annotated documents for evaluation (testing set 2)

trigger	MTS	OTS
PRE	97	17
POST	41	11
CONJ	242	7
PSEU	107	21
total	487	56

Table 5: Comparison of trigger sets

next several years (Protected Artificial Intelligence Innovation Environment for Patient Oriented Digital Health Solutions for developing, testing and evidence based evaluation of clinical value).

5.1. Comparison of pipelines 1a and 1b

We performed the evaluation on 1) the MACSS triggers used in Cotik et al¹⁵, called “MTS” in pipeline 1a, and 2) our own set of triggers “OTS” in pipeline 1b. Table 5 shows the different kinds of triggers in the two trigger sets and their counts.

Besides the set of triggers the only remaining configuration option of NegEx is the size of the window, i.e., the maximum scope of a negation (given as number of words). As a basic rule of NegEx any negation reaches up to the end/beginning of a sentence or to a conjunction. We evaluated all documents with the following window sizes: no restriction, 5, 4, and 3. The complete results are shown in the following tables (7, 8, 9, and 10). Table 6 shows the F1 scores of all variations (highest values are in bold, lowest values in italics). For comparison, the F1 values reported in Cotik et al. (which used trigger set 1) has been 0.91 for summaries and 0.96 for notes.

If the size of the scope remains unlimited, the distance between a negation and a concept can be up to 22 words (for document type notes). The reason for this lies in the telegram like writing style of notes which can lead to long ‘sentences’ which are sometimes just long listings of findings.

We also studied which triggers occurred most often. It turned out that looking at the notes the terms *keine*, *kein*, *ohne*, and *Ausschluss* (no, without, and exclusion) cover over 98% of negations. The evaluation reveals some interesting aspects:

1. The results on clinical notes are almost identical when comparing the two trigger sets on the same data. This is probably caused by the fact that within notes, negations are mostly expressed with simple negation terms.
2. When comparing the results of different window sizes, there is a noticeable difference between the two types of documents: the best F1 values on notes can be achieved with window size 3, the worst with unlimited size. Almost

¹⁵http://macss.dfki.de/negex/negex_trigger_german_biotxtm_2016.txt

F1	discharge summaries				clinical notes			
	∞	5	4	3	∞	5	4	3
MTS set 1	0.922	0.959	0.962	<i>0.878</i>	<i>0.953</i>	0.971	0.974	0.981
OTS set 1	0.909	0.954	0.933	<i>0.832</i>	<i>0.950</i>	0.970	0.974	0.981
MTS set 2	0.900	0.902	0.891	<i>0.849</i>	<i>0.929</i>	0.966	0.966	0.974
OTS set 2	0.909	0.909	0.897	<i>0.854</i>	<i>0.934</i>	0.971	0.971	0.979

Table 6: F1 values for all tests

MTS set 1	discharge summaries				clinical notes			
	∞	5	4	3	∞	5	4	3
TP	107	105	102	86	336	335	333	332
TN	392	403	407	407	253	267	271	277
FP	17	6	2	2	31	17	13	7
FN	1	3	6	22	2	3	5	6
Acc	0.965	0.983	0.985	<i>0.954</i>	<i>0.947</i>	0.968	0.971	0.979
Prec	<i>0.863</i>	0.946	0.981	0.977	<i>0.916</i>	0.952	0.962	0.979
Rec	0.991	0.972	0.944	<i>0.796</i>	0.994	0.991	0.985	<i>0.982</i>
F1	0.922	0.959	0.962	<i>0.878</i>	<i>0.953</i>	0.971	0.974	0.981

Table 7: Evaluation results for set 1 and trigger set MTS

the opposite is true for summaries: window size 3 performs worst, size 5 is best (except for MTS on set 1, where 4 is slightly better). This can be explained by the more complex phrasings in the summaries where more adjectives are used (more complex adverbial modifier constructions are used) that increase the distance between negations and negated terms.

3. The difference between using MTS and OTS triggers is small: on the first set for discharge summaries the best F1 value for MTS is 0.962, the best for OTS is 0.954, on the second set its 0.902 and 0.909.
4. When examining the false positives of summaries (evaluation set 2), the following causes were most frequent: 1) faults due to the propagation of a trigger too far, 2) wrong assignment (according to the ground truth) when a term (probably) just negates an adjective as in “kein sicherer Anhalt” (no reliable indication)¹⁶. Another interesting case is “Zeichen rechts positiv, links negativ” (signs right positive, left negative), where a

¹⁶These cases are ambiguous, as in “no bacterial infection”, for example it is not clear if there is any other kind of infection.

OTS set 1	discharge summaries				clinical notes			
	∞	5	4	3	∞	5	4	3
TP	105	103	97	79	334	334	333	332
TN	391	404	406	406	253	267	271	277
FP	18	5	3	3	31	17	13	7
FN	3	5	11	29	3	4	5	6
Acc	0.959	0.981	0.973	<i>0.938</i>	<i>0.944</i>	0.966	0.971	0.979
Prec	<i>0.854</i>	0.954	0.970	0.963	<i>0.915</i>	0.952	0.962	0.979
Rec	0.972	0.954	0.898	<i>0.731</i>	0.988	0.988	0.985	<i>0.982</i>
F1	0.909	0.954	0.933	<i>0.832</i>	<i>0.950</i>	0.970	0.974	0.981

Table 8: Evaluation results for set 1 and trigger set OTS

MTS set 2	discharge summaries				clinical notes			
	∞	5	4	3	∞	5	4	3
TP	684	655	613	555	302	302	301	300
TN	1743	1792	1811	1821	220	246	248	254
FP	111	62	43	33	41	15	13	7
FN	36	65	107	165	4	4	5	6
Acc	0.943	0.951	0.942	<i>0.923</i>	<i>0.921</i>	0.966	0.968	0.977
Prec	<i>0.860</i>	0.914	0.934	0.944	<i>0.880</i>	0.953	0.959	0.977
Rec	0.950	0.910	0.851	<i>0.771</i>	0.987	0.987	0.984	<i>0.980</i>
F1	0.900	0.902	0.891	<i>0.849</i>	<i>0.929</i>	0.966	0.966	0.974

Table 9: Evaluation results for set 2 and trigger set MTS

OTS set 2	discharge summaries				clinical notes			
	∞	5	4	3	∞	5	4	3
TP	692	654	629	565	302	302	301	300
TN	1741	1789	1800	1816	222	247	248	254
FP	113	65	54	38	39	14	13	7
FN	28	66	91	155	4	4	5	6
Acc	0.945	0.949	0.944	<i>0.925</i>	<i>0.924</i>	0.968	0.968	0.977
Prec	<i>0.860</i>	0.910	0.921	0.937	<i>0.886</i>	0.956	0.959	0.977
Rec	0.961	0.908	0.874	<i>0.785</i>	0.987	0.987	0.984	<i>0.980</i>
F1	0.909	0.909	0.897	<i>0.854</i>	<i>0.934</i>	0.971	0.971	0.979

Table 10: Evaluation results for set 2 and trigger set OTS

term seems both positive and negative and has to be discriminated or a complex term to be constructed first.

Furthermore we observed that NegEx does not handle interferences between pre- and post-negation triggers: let’s assume “not” is a pre-negation trigger and “not visible” a post-negation. Let’s further regard the sentence “lesions are not visible”. The implementations of NegEx and ConText both iterate over the words of a sentence and replace parts matching a trigger by special tags. As it uses pre-negations prior to post-negations in our example “not” will be replaced by “<NEG_PRE>”. In the resulting sentence “lesions are <NEG_PRE> visible” the post-negation trigger “not visible” is no longer present and the correct negation will therefore not be annotated. We adopted our own implementation to consider this special case by doing an extra post-check if a pre-negation did not cover any concept.

5.2. Pipeline 2: Dependency parsing

We originally planned to compare the results of two methods to detect negations: NegEx and dependency parsing. During our implementations and after some experiments we had to drop the idea of a quantitative evaluation of the dependency parser based on the following reasons: 1) phrasings and negation patterns in our example documents turned out to be mostly pretty simple (as shown above), therefore the NegEx algorithm produces very good results and there is pragmatically not much room for improvement, 2) applying the Stanford dependency parser to German medical texts turned out to be much more difficult and resource-intensive than expected. We describe the qualitative results in the following.

The selected examples of our dataset used to demonstrate the potentials of dependency parsing are encouraging but they just don’t occur often enough in our documents to be significant. For example, one possible source of incorrect

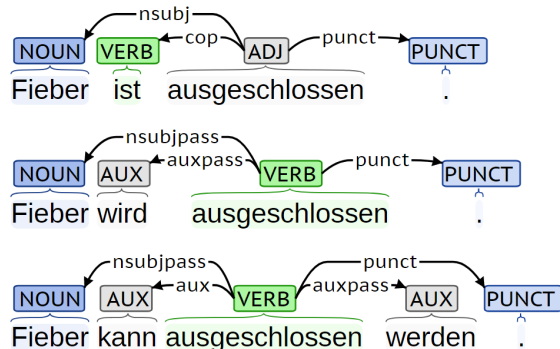


Figure 3: Simple patterns covering several use cases, extended to except negated exclusions: `{}/</nsubj.*/ ({word:/ausgeschlossen/} !>> /neg/ {})`

results of NegEx are double and pseudo-negations. Phrases like “x ist nicht vorhanden” (x is not present) or “x ist nicht ausgeschlossen” (x is not excluded) are very similar but the latter does not negate the existence of x but the *exclusion* of x which says little about the actual existence of x. Therefore it’s called a pseudo-negation. As a consequence it is not sufficient to use “nicht” or “ausgeschlossen” as triggers as both will produce a false positive result on this sentence. “not excluded” is a double negation which must be handled special.

To cover these cases we employed special rules on the dependency trees. We started by examining some simple sentences like in figure 3. The corresponding pattern matches sentences like “Fieber ist ausgeschlossen.“, “Fieber wird ausgeschlossen.“, and “Fieber kann ausgeschlossen werden.” (Fever is excluded. Fever can be excluded.). This pattern even covers sentences where governor and dependents are far separated (in German the verb often appears at the end of a sentence), like “*Fieber und Schwindel können nach aktuellem Stand der Untersuchungen bisher ausgeschlossen werden.*” (see figure 4: fever and dizziness can be excluded so far according to the current state of research). “!>> neg” is added to ensure negated verbs are *not* matched (remember ‘>>’ is a relation chain and the preceding ‘!’ excludes such a chain). These long distance phrases can be covered when regarding the (semantic) dependency structure of a sentence.

On the other hand very similar sentences may produce different dependency parse trees and therefore are difficult to be covered with only a few patterns. Consider the two sentences of figure 5 that differ only in the word “keine” / “keinerlei” (both words mean “no”, “none”, or “not any”).

When you step deeper into specifying patterns for negations you soon encounter some challenges (at least in German). Sentences without a conjugated verb are not well covered by the Stanford NLP dependency parser model. They often occur in medical short texts like findings where the auxiliary verb “be” is left out, e.g., “no edema detectable” or “no fever”.

Figure 6 shows four very common wordings of the same fact (all stating that

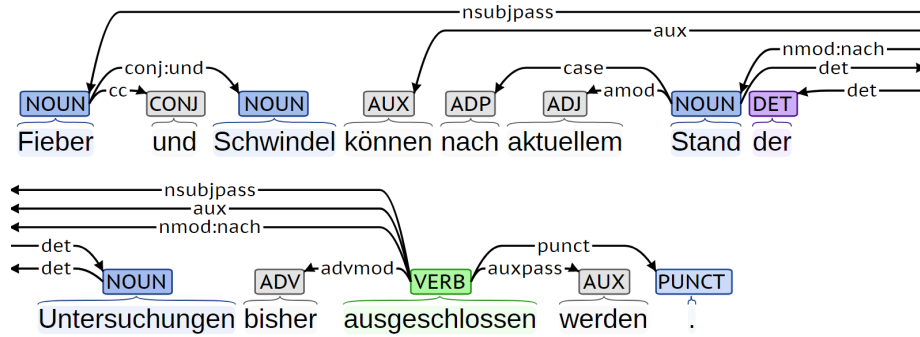


Figure 4: Sentence with long inclusion

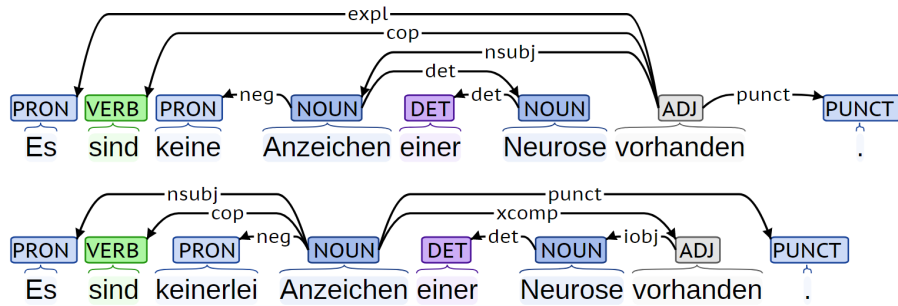


Figure 5: Similar sentences with heavily differing parse trees.

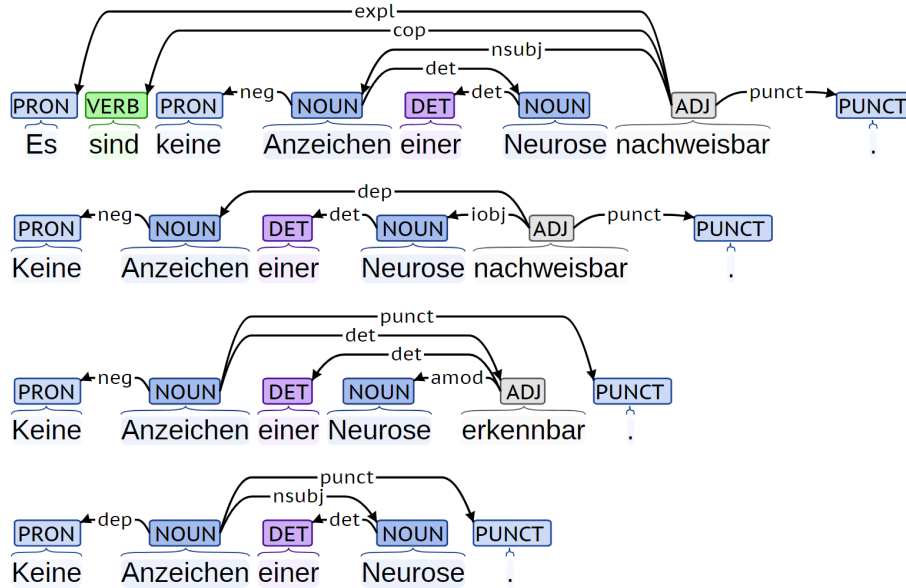


Figure 6: Different semantic structures of similar phrases.

there are no signs of neurosis recognizable/verifiable). Although they differ only slightly, their dependency parser outputs are very different and cannot be covered by the same pattern.

Incorrect or ‘sloppy’ punctuation makes this effect even worse, as can be seen in figure 7 on the sentence “Fieber as well as infections are excluded”.

On the other hand dependency parsing has the great advantage that patterns over dependency trees offer easy ways to *exclude* specific wordings which is relevant to the mentioned double negations. Consider a simple negation pattern like “kein x (no x)”, which can be covered by the pattern “{ } > /neg/ { }”. To exclude sentences like “kein Ausschluss von x” (no exclusion of x) we can extend the pattern to `!{lemma: /ausschluss/}=dep >/neg/{ }`, which specifies that the

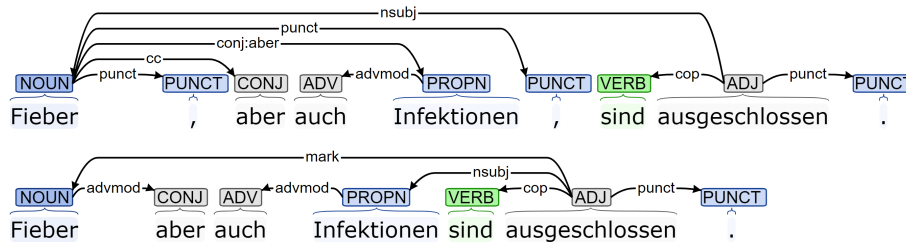


Figure 7: Different semantic structures caused by punctuation.

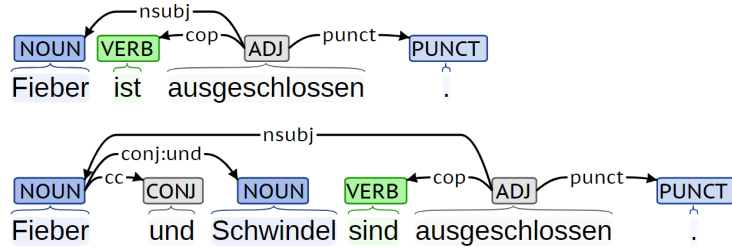


Figure 8: Node conjunctions.

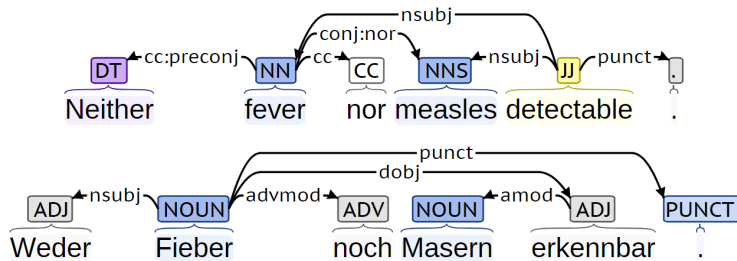


Figure 9: English: “Neither x nor y”, German: “Weder x noch y”

dependent of the negation must not be the word “ausschluss” (exclusion).

The need for a more detailed distinction of wordings becomes obvious if you look at the phrases “Test auf Masern” and “Verdacht auf Masern” (test/suspicion on measles). Although measles are mentioned in the first case this is no statement on their appearance. This could be flagged as a pseudo-positive wording. The second phrase at least increases the likelihood of measles and could be marked as a (positive) speculation. If we extend the phrase to “Test auf Masern war positiv” (test on measles was positive) measles have to be marked as affirmed.

Our example data contained an interesting sentence: “Weder enzymkinetisch noch elektrokardiografisch gab es Anhalt für eine kardiale Ischaemie” (Neither enzyme kinetics nor electrocardiography gave evidence for cardiac ischemia). This form of surrounding trigger “weder ... noch ...” (neither ... nor ...) cannot be covered by the NegEx patterns. In contrast to NegEx, Semgrex patterns are more flexible and allow for the specification of such cases. However, the Stanford NLP model for German does not seem to know about this construction as a comparison of the resulting trees in English and German shows (see figure 9).

6. General Results

For our example documents a rather short list of triggers combined with a scope window size of 5 was sufficient to achieve good results with the NegEx algorithm. This set can easily be maintained and requires no special know-how. When regarding the ground truth, we reached a F1 value of at least 0.954. Extending the set of triggers (like it was done by Cotik et al. [3]) from “kein” to “Patient zeigt kein”, for example, does not have any advantages.

Our results in dependency parsing using the Stanford NLP parser depend very much on the German model used by the parser, which—as we showed above—does not always seem to be well trained on wordings often used in medical texts.

The examples above demonstrate the inherent difficulties of dependency parsing, namely the noticeable differences of parse trees, e.g.,

- on omitting terms like ‘is’ or ‘are’ or a declined verb,
- on variants in punctuation,
- on using synonyms,
- slight changes in the sentence structure.

As these wording aspects are very common in medical texts (at least in documents of the type we tested on), handling all resulting parse trees is very cumbersome and tedious. The correct coverage of more complex phrasings is of little importance in our case as they seldom occur in our text (if at all). Therefore the disadvantages of dependency parsing predominate significantly. The rather poor Stanford model for German adds an additional negative aspect.

Additionally the negations of most of our test phrases from medical texts can be easily covered by NegEx. The idea of combining both methods (NegEx and Dependency Parsing) and exploiting an additional analysis by dependency parsing failed due to the lacking robustness to small changes in surface structure.

7. Discussion

Structural information as provided by dependency parsers is useful for a variety of text types in the medical domain. Although [43, 44, 45, 46] reported on several success stories in bioinformatics, e.g., recognizing context polarity (e.g., negations of medical terms), named entity recognition, or the extraction of medical entities in medical web documents, the (structure and) results of the type of examination reports as under investigation here is very different.

In order to evaluate and predict parser usefulness for negation detection, we reported on the accuracy when compared to simple string parsing methods on the surface form. Evaluating the parsers using task specific metrics, we have shown that dependency parsing does not lead to a better average accuracy.

Dependency parsers provide information useful for a variety of down the line applications in medical texts, [47] for example report on automatically assigning

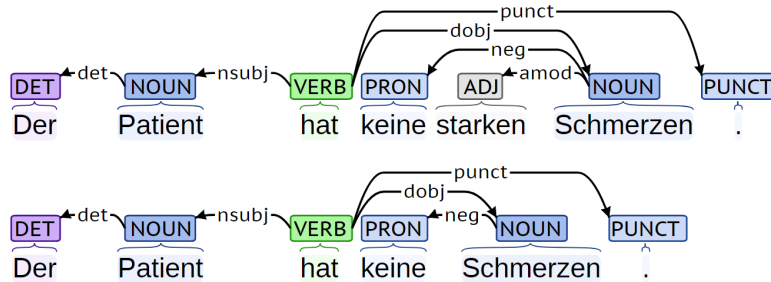


Figure 10: Unclear association of negations.

ICD-9-CM codes to patient records which is structurally similar to our extraction task. However, negation detection as found in our text repository cannot benefit from the advantages reported in previous works ([43, 44, 45, 46, 47]). It seems that more sophisticated extraction tasks only (e.g., the automatic extraction of relations between medical concepts ([48])) which exhibit free-word order constructs or other complex syntactical constructions are needed. These can be characterized as information extraction tasks best described in terms of a set of binary relations that hold between the relevant constituents in more complex relations in a sentence of a medical text, an assumption which does not hold in our corpus.

Sometimes the reference of a negation in a sentence is ambiguous. Although it is argued (e.g., in [3]) that in these cases dependency parsing would help, the following example shows that this is not always the case: It is not clear whether “Keine starken Schmerzen” (no strong pain) should be interpreted as “no pain at all” or just “no *strong* pain”. The Stanford dependency parser does not change the assertion of the negation to the noun, see figure 10, which is the same as we get from NegEx.

8. Conclusion

We presented our work on detecting negations of concepts in medical texts like discharge summaries and clinical notes. We integrated a variant of the NegEx algorithm in our information extraction framework and examined different configurations. As an alternative approach and to overcome shortcomings of NegEx, components for deeper natural language analysis were added. Both methods showed some advantages as well as drawbacks.

NegEx is an easy-to-implement, simple algorithm based on a list of triggers easily maintained and fairly clear. It allows for a simple handling of triggers, but relies very much on the surface structure of sentences and a complete covering of variants, which can be rather cumbersome in German. The vague definition of the scope of negation triggers and the associated inability to handle long distance phrases are its main disadvantages. Some wordings like “weder ... noch” (neither ... nor) that enclose a term cannot be expressed.

Applying patterns on the result of dependency parsing appears to overcome the disadvantages of triggers depending on regular expressions (like in NegEx). Semgrep patterns can match long distance phrases without any additional effort by the programmer, conjuncted terms are automatically included, enclosing patterns may cover additional cases. On the other hand the patterns are much harder to define and understand (requires expertise in linguistics) and the handling of a larger list of patterns is rather cumbersome. Every simple and small word changes in a sentence may unpredictably alter the corresponding (variant of the) dependency structure leading to the need to check every pattern when considering new phrases and every example sentence when adding new patterns. Our results show that dependency parsing is neither a good alternative nor a satisfactory complement, at least for our type of German medical documents with the given internal dependency parsing mechanism; this is counter-intuitive to the fact that head-dependent relations should provide an approximation to the negation scope and this should make them useful for medical IE tasks.

Based on our empirical evaluation, we would propose the following procedure:

- start with a simple implementation of NegEx,
- check the types of negations in your documents,
- define corresponding simple triggers reduced to a minimum of words,
- evaluate NegEx with these triggers and different scope values,
- if complex negation phrases are common to your texts, you may try to cover them with dependency parsing.

Acknowledgements

This research is part of the project “clinical data intelligence” (KDI) which was founded by the Federal Ministry for Economic Affairs and Energy (BMWi) and part of the new pAItient project founded by the Federal Ministry of Health (BMG).

References

- [1] M. Wiegand, A. Balahur, B. Roth, D. Klakow, A. Montoyo, A survey on the role of negation in sentiment analysis, in: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, NeSp-NLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 60–68.
URL <http://dl.acm.org/citation.cfm?id=1858959.1858970>
- [2] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, B. G. Buchanan, A simple algorithm for identifying negated findings and diseases

- in discharge summaries, *Journal of Biomedical Informatics* 34 (5) (2001) 301–310. doi:10.1006/jbin.2001.1029.
URL <https://doi.org/10.1006/jbin.2001.1029>
- [3] V. Cotik, R. Roller, F. Xu, H. Uszkoreit, K. Budde, D. Schmidt, Negation detection in clinical reports written in german, in: S. Ananiadou, R. Batista-Navarro, K. B. Cohen, D. Demner-Fushman, P. Thompson (Eds.), *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining, BioTxtM@COLING 2016, Osaka, Japan, December 12, 2016, The COLING 2016 Organizing Committee, 2016*, pp. 115–124.
URL <https://www.aclweb.org/anthology/W16-5113>
- [4] S. Vintar, L. Todorovski, D. Sonntag, P. Buitelaar, Evaluating context features for medical relation mining, in: *Proceedings of the ECML/PKDD Workshop on Data Mining and Text Mining for Bioinformatics, 2003*.
- [5] D. Sonntag, P. Wennerberg, P. Buitelaar, S. Zillner, Pillars of ontology treatment in the medical domain, *J. Cases on Inf. Techn.* 11 (4) (2009) 47–73.
- [6] A. Alicante, Unsupervised entity and relation extraction from clinical records in italian, *Computers in Biology and Medicine* 72 (1) (2016) 263–275.
- [7] P. Odom, V. Bangera, T. Khot, D. Page, S. Natarajan, Extracting adverse drug events from text using human advice, in: *Artificial Intelligence in Medicine - 15th Conference on Artificial Intelligence in Medicine, AIME 2015, Pavia, Italy, June 17-20, 2015. Proceedings, 2015*, pp. 195–204.
- [8] J. Métivier, L. Serrano, T. Charnois, B. Cuissart, A. Widlöcher, Automatic symptom extraction from texts to enhance knowledge discovery on rare diseases, in: *Artificial Intelligence in Medicine - 15th Conference on Artificial Intelligence in Medicine, AIME 2015, Pavia, Italy, June 17-20, 2015. Proceedings, 2015*, pp. 249–254.
- [9] C. Bretschneider, S. Zillner, M. Hammon, P. Gass, D. Sonntag, Automatic extraction of breast cancer information from clinical reports, in: P. D. Bamidis, S. T. Konstantinidis, P. P. Rodrigues (Eds.), *30th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2017, Thessaloniki, Greece, June 22-24, 2017, IEEE Computer Society, 2017*, pp. 213–218. doi:10.1109/CBMS.2017.138.
URL <https://doi.org/10.1109/CBMS.2017.138>
- [10] O. Gros, M. Stede, *Determining Negation Scope in German and English Medical Diagnoses*, Brill, Leiden, Niederlande, 2014, pp. 113 – 126. doi:https://doi.org/10.1163/9789004258174_006.
URL https://brill.com/view/book/edcoll/9789004258174/B9789004258174_006.xml

- [11] R. Morante, Descriptive analysis of negation cues in biomedical texts, in: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odiijk, S. Piperidis, M. Rosner, D. Tapias (Eds.), Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta, European Language Resources Association, 2010.
URL <http://www.lrec-conf.org/proceedings/lrec2010/summaries/335.html>
- [12] A. S. Wu, B. H. Do, J. Kim, D. L. Rubin, Evaluation of negation and uncertainty detection and its impact on precision and recall in search, *J. Digital Imaging* 24 (2) (2011) 234–242. doi:10.1007/s10278-009-9250-4.
URL <https://doi.org/10.1007/s10278-009-9250-4>
- [13] S. T. Wu, T. A. Miller, J. J. Masanz, M. Coarr, D. Carrell, S. R. Halgrim, D. Harris, C. Clark, Negation’s not solved: Reconsidering negation annotation and evaluation, in: AMIA 2013, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 16-20, 2013, AMIA, 2013.
URL <http://knowledge.amia.org/amia-55142-a2013e-1.580047/t-03-1.584514/f-003-1.584515/a-560-1.584527/a-581-1.584521>
- [14] G. Gkotsis, S. Velupillai, A. Oelrich, H. Dean, M. Liakata, R. Dutta, Don’t let notes be misunderstood: A negation detection method for assessing risk of suicide in mental health records, in: K. Hollingshead, L. H. Ungar (Eds.), Proceedings of the 3rd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, CLPsych@NAACL-HLT 2016, June 16, 2016, San Diego, California, USA, The Association for Computational Linguistics, 2016, pp. 95–105.
URL <http://aclweb.org/anthology/W/W16/W16-0310.pdf>
- [15] C. Friedman, G. Hripcsak, L. Shagina, H. Liu, Research paper: Representing information in patient reports using natural language processing and the extensible markup language, *JAMIA* 6 (1) (1999) 76–87. doi:10.1136/jamia.1999.0060076.
URL <https://doi.org/10.1136/jamia.1999.0060076>
- [16] W. W. Chapman, D. Hillert, S. Velupillai, M. Kvist, M. Skeppstedt, B. E. Chapman, M. Conway, M. Tharp, D. L. Mowery, L. Deléger, Extending the negex lexicon for multiple languages, in: Lehmann et al. [49], pp. 677–681. doi:10.3233/978-1-61499-289-9-677.
URL <https://doi.org/10.3233/978-1-61499-289-9-677>
- [17] K. J. Mitchell, M. J. Becich, J. J. Berman, W. W. Chapman, J. R. Gilbertson, D. Gupta, J. Harrison, E. Legowski, R. S. Crowley, Implementation and evaluation of a negation tagger in a pipeline-based system for information extraction from pathology reports, in: M. Fieschi, E. W. Coiera, J. Y. Li (Eds.), MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics, San Francisco, California, USA, September 7-11,

- 2004, Vol. 107 of Studies in Health Technology and Informatics, IOS Press, 2004, pp. 663–667. doi:10.3233/978-1-60750-949-3-663.
URL <https://doi.org/10.3233/978-1-60750-949-3-663>
- [18] H. Harkema, J. N. Dowling, T. Thornblade, W. W. Chapman, Context: An algorithm for determining negation, experiencer, and temporal status from clinical reports, *Journal of Biomedical Informatics* 42 (5) (2009) 839–851. doi:10.1016/j.jbi.2009.05.002.
URL <https://doi.org/10.1016/j.jbi.2009.05.002>
- [19] M. Skeppstedt, Negation detection in swedish clinical text: An adaption of negex to swedish, *J. Biomedical Semantics* 2 (S-3) (2011) S3.
URL <http://www.jbiomedsem.com/content/2/S3/S3>
- [20] L. Deléger, C. Grouin, Detecting negation of medical problems in french clinical notes, in: G. Luo, J. Liu, C. C. Yang (Eds.), *ACM International Health Informatics Symposium, IHI '12*, Miami, FL, USA, January 28–30, 2012, ACM, 2012, pp. 697–702. doi:10.1145/2110363.2110443.
URL <http://doi.acm.org/10.1145/2110363.2110443>
- [21] V. Cotik, V. Stricker, J. Vivaldi, H. Rodríguez, Syntactic methods for negation detection in radiology reports in spanish, in: K. B. Cohen, D. Demner-Fushman, S. Ananiadou, J. Tsujii (Eds.), *Proceedings of the 15th Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2016*, Berlin, Germany, August 12, 2016, Association for Computational Linguistics, 2016, pp. 156–165. doi:10.18653/v1/W16-2921.
URL <https://doi.org/10.18653/v1/W16-2921>
- [22] R. Costumero, F. Lopez, C. Gonzalo-Martín, M. Millán, E. Menasalvas, An approach to detect negation on medical documents in spanish, in: D. Slezak, A. Tan, J. F. Peters, L. Schwabe (Eds.), *Brain Informatics and Health - International Conference, BIH 2014*, Warsaw, Poland, August 11–14, 2014, Proceedings, Vol. 8609 of Lecture Notes in Computer Science, Springer, 2014, pp. 366–375. doi:10.1007/978-3-319-09891-3_34.
URL https://doi.org/10.1007/978-3-319-09891-3_34
- [23] Z. Afzal, E. Pons, N. Kang, M. C. J. M. Sturkenboom, M. J. Schuemie, J. A. Kors, Contextd: an algorithm to identify contextual properties of medical terms in a dutch clinical corpus, *BMC Bioinformatics* 15 (2014) 373. doi:10.1186/s12859-014-0373-3.
URL <https://doi.org/10.1186/s12859-014-0373-3>
- [24] H. Tanushi, H. Dalianis, M. Duneld, M. Kvist, M. Skeppstedt, S. Velupillai, Negation scope delimitation in clinical text using three approaches: Negex; pycontextnlp and synneg, in: S. Oepen, K. Hagen, J. B. Johannessen (Eds.), *Proceedings of the 19th Nordic Conference of Computational Linguistics, NODALIDA 2013*, May 22–24, 2013, Oslo University, Norway, Vol. 85 of Linköping Electronic Conference Proceedings, Linköping

University Electronic Press, 2013, pp. 387–397.
URL <http://www.ep.liu.se/ecp/article.asp?issue=085&article=035&volume=>

- [25] P. Mutalik, A. M. Deshpande, P. M. Nadkarni, Research paper: Use of general-purpose negation detection to augment concept indexing of medical documents: A quantitative study using the UMLS, *JAMIA* 8 (6) (2001) 598–609. doi:10.1136/jamia.2001.0080598.
URL <https://doi.org/10.1136/jamia.2001.0080598>
- [26] Y. Huang, H. J. Lowe, Research paper: A novel hybrid approach to automated negation detection in clinical radiology reports, *JAMIA* 14 (3) (2007) 304–311. doi:10.1197/jamia.M2284.
URL <https://doi.org/10.1197/jamia.M2284>
- [27] Ö. Uzuner, X. Zhang, T. C. Sibanda, Research paper: Machine learning and rule-based approaches to assertion classification, *JAMIA* 16 (1) (2009) 109–115. doi:10.1197/jamia.M2950.
URL <https://doi.org/10.1197/jamia.M2950>
- [28] B. de Bruijn, C. Cherry, S. Kiritchenko, J. D. Martin, X. Zhu, Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010, *JAMIA* 18 (5) (2011) 557–562. doi:10.1136/amiajnl-2011-000150.
URL <https://doi.org/10.1136/amiajnl-2011-000150>
- [29] C. A. Bejan, L. Vanderwende, F. Xia, M. Yetisgen-Yildiz, Assertion modeling and its role in clinical phenotype identification, *Journal of Biomedical Informatics* 46 (1) (2013) 68–74. doi:10.1016/j.jbi.2012.09.001.
URL <https://doi.org/10.1016/j.jbi.2012.09.001>
- [30] S. Sohn, S. Wu, C. G. Chute, Dependency parser-based negation detection in clinical narratives, in: *AMIA Joint Summits on Translational Science proceedings*. AMIA Joint Summits on Translational Science, 2012.
- [31] S. Mehrabi, C. M. Schmidt, J. A. Waters, C. Beesley, A. Krishnan, J. Kesterson, P. R. Dexter, M. A. Al-Haddad, W. M. Tierney, M. J. Palakal, An efficient pancreatic cyst identification methodology using natural language processing, in: *Lehmann et al. [49]*, pp. 822–826. doi:10.3233/978-1-61499-289-9-822.
URL <https://doi.org/10.3233/978-1-61499-289-9-822>
- [32] S. Mehrabi, A. Krishnan, S. Sohn, A. M. Roch, H. Schmidt, J. Kesterson, C. Beesley, P. R. Dexter, C. M. Schmidt, H. Liu, M. J. Palakal, DEEPEN: A negation detection system for clinical text incorporating dependency relation into negex, *Journal of Biomedical Informatics* 54 (2015) 213–219. doi:10.1016/j.jbi.2015.02.010.
URL <https://doi.org/10.1016/j.jbi.2015.02.010>

- [33] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. K. Schuler, C. G. Chute, Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications, *JAMIA* 17 (5) (2010) 507–513. doi:10.1136/jamia.2009.001560. URL <https://doi.org/10.1136/jamia.2009.001560>
- [34] M. Ballesteros, V. Francisco, A. Díaz, J. Herrera, P. Gervás, Inferring the scope of negation in biomedical documents, in: A. F. Gelbukh (Ed.), *Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part I*, Vol. 7181 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 363–375. doi:10.1007/978-3-642-28604-9_30. URL https://doi.org/10.1007/978-3-642-28604-9_30
- [35] W. Weng, Y. Chung, S. Tong, Clinical text summarization with syntax-based negation and semantic concept identification, *CoRR abs/2003.00353* (2020). arXiv:2003.00353. URL <https://arxiv.org/abs/2003.00353>
- [36] Y. Peng, X. Wang, L. Lu, M. Bagheri, R. M. Summers, Z. Lu, Negbio: a high-performance tool for negation and uncertainty detection in radiology reports, *CoRR abs/1712.05898* (2017). arXiv:1712.05898. URL <http://arxiv.org/abs/1712.05898>
- [37] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. L. Ball, K. S. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, A. Y. Ng, Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, in: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, AAAI Press, 2019, pp. 590–597. doi:10.1609/aaai.v33i01.3301590. URL <https://doi.org/10.1609/aaai.v33i01.3301590>
- [38] D. Ferrucci, A. Lally, K. Verspoor, E. Nyberg, Unstructured information management architecture (UIMA) version 1.0, *OASIS Standard* (mar 2009). URL <https://docs.oasis-open.org/uima/v1.0/uima-v1.0.html>
- [39] R. Eckart de Castilho, I. Gurevych, A broad-coverage collection of portable nlp components for building shareable analysis pipelines, in: *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, Association for Computational Linguistics and Dublin City University, Dublin, Ireland, 2014, pp. 1–11. URL <http://www.aclweb.org/anthology/W14-5201>

- [40] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.
URL <http://www.aclweb.org/anthology/P/P14/P14-5010>
- [41] N. Chambers, D. M. Cer, T. Grenager, D. L. W. Hall, C. Kiddon, B. MacCartney, M. de Marneffe, D. Ramage, E. Yeh, C. D. Manning, Learning alignments and leveraging natural logic, in: S. Sekine, K. Inui, I. Dagan, B. Dolan, D. Giampiccolo, B. Magnini (Eds.), Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June 28-29, 2007, Association for Computational Linguistics, 2007, pp. 165–170.
URL <https://aclanthology.info/papers/W07-1427/w07-1427>
- [42] P. Stenetorp, G. Topić, S. Pyysalo, T. Ohta, J.-D. Kim, J. Tsujii, Bionlp shared task 2011: Supporting resources, in: Proceedings of BioNLP Shared Task 2011 Workshop, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 112–120.
URL <http://www.aclweb.org/anthology/W11-1816>
- [43] K. Fundel, R. Küffner, R. Zimmer, Relex - relation extraction using dependency parse trees, *Bioinformatics* 23 (3) (2007) 365–371. doi:10.1093/bioinformatics/btl616.
URL <https://doi.org/10.1093/bioinformatics/btl616>
- [44] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada, The Association for Computational Linguistics, 2005, pp. 347–354.
URL <https://www.aclweb.org/anthology/H05-1044/>
- [45] J. R. Finkel, C. D. Manning, Joint parsing and named entity recognition, in: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA, The Association for Computational Linguistics, 2009, pp. 326–334.
URL <https://www.aclweb.org/anthology/N09-1037/>
- [46] J. R. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. D. Manning, G. Sinclair, Exploiting context for biomedical entity recognition: From syntax to the web, in: N. Collier, P. Ruch, A. Nazarenko (Eds.), Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, NLPBA/BioNLP 2004, Geneva, Switzerland, August 28-29, 2004, 2004.
URL <https://www.aclweb.org/anthology/W04-1217/>

- [47] P. Chen, A. Barrera, C. Rhodes, Semantic analysis of free text and its application on automatically assigning ICD-9-CM codes to patient records, in: F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner, L. A. Zadeh (Eds.), Proceedings of the 9th IEEE International Conference on Cognitive Informatics, ICCI 2010, July 7-9, 2010, Beijing, China, IEEE Computer Society, 2010, pp. 68–74. doi:10.1109/COGINF.2010.5599783.
URL <https://doi.org/10.1109/COGINF.2010.5599783>
- [48] B. Rink, S. M. Harabagiu, K. Roberts, Automatic extraction of relations between medical concepts in clinical texts, JAMIA 18 (5) (2011) 594–600. doi:10.1136/amiajnl-2011-000153.
URL <https://doi.org/10.1136/amiajnl-2011-000153>
- [49] C. U. Lehmann, E. Ammenwerth, C. Nøhr (Eds.), MEDINFO 2013 - Proceedings of the 14th World Congress on Medical and Health Informatics, 20-13 August 2013, Copenhagen, Denmark, Vol. 192 of Studies in Health Technology and Informatics, IOS Press, 2013.