Research Article

Minimalistic Approach to Coreference Resolution in Lithuanian Medical Records

Voldemaras Žitkus,1 Rita Butkienė,1 Rimantas Butleris,1 Rytis Maskeliūnas,1 Robertas Damaščius,1 and Marcin Woźniak2

1Faculty of Informatics, Kaunas University of Technology, 51386 Kaunas, Lithuania
2Institute of Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland

Correspondence should be addressed to Robertas Damaščius; robertas.damasevicius@ktu.lt

Received 18 January 2019; Accepted 26 February 2019; Published 20 March 2019

Guest Editor: Ka L. Ma

Copyright © 2019 Voldemaras Žitkus et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Coreference resolution is a challenging part of natural language processing (NLP) with applications in machine translation, semantic search and other information retrieval, and decision support systems. Coreference resolution requires linguistic preprocessing and rich language resources for automatically identifying and resolving such expressions. Many rarer and under-resourced languages (such as Lithuanian) lack the required language resources and tools. We present a method for coreference resolution in Lithuanian language and its application for processing e-health records from a hospital reception. Our novelty is the ability to process coreferences with minimal linguistic resources, which is important in linguistic applications for rare and endangered languages. The experimental results show that coreference resolution is applicable to the development of NLP-powered online healthcare services in Lithuania.

1. Introduction

Digital means of medical informatics, especially when applying natural language processing (NLP), are indispensable in the application of e-health and digitalization of medical records and processes [1]. The use of NLP has proved as a lower-cost alternative to traditional medical methods in many cases such as to forecast stress symptoms and suicide risk in free-text responses sent via a mobile phone [2], or to detect seasonal disease outbreaks by monitoring search engine queries [3], and discovery of healthcare knowledge from social media [4, 5].

With the development of Semantic Web technology, web information retrieval (IR) is changing towards meaning-based IR. The quality of retrieved documents relevant to the user also highly depends on the information extraction (IE) methods applied. In general, IE focuses on automatic extraction of structured information from the unstructured source. Standard document text preprocessing steps used in IE are lexical analysis, morphological analysis, and named entity recognition (NER), which can be complemented by coreference resolution and semantic annotation. The main issue here is the ambiguity and complexity of the natural language, thus making the progress in IE dependent on the evolution of the NLP techniques. While for widely used languages (such as English), the IE-related NLP research has already reached the levels of maturity and practical application on a massive scale (e.g., IBM Watson project) [6], but the resource-poor languages, such as Lithuanian [7], remain an open NLP research field. The baseline application is often steered towards automated concept extraction [8, 9], often in combination with text mining [10, 11].

NER when applied to biomedical texts is a critical step for developing decision support tools for smart healthcare. Examples for it are as follows: drug name recognition (DNR), which recognizes pharmacological substances from biomedical texts and classifies them for discovering drug-drug interactions [12, 13]; biomedical named entity recognition (BNER), which extracts biomedical concepts of interest such as genes and proteins [14]; and medical entity recognition, which is information extraction from unstructured electronic health records [15–17]. Such studies include mapping clinical
Machine-learning and rule-based approaches are efficient methods in semantic processing, especially when enhanced with external knowledge and coreference clues derived from the structured document, while often still performing better (in comparison with classic implementations) in coreference resolution when provided with ground truth mentions [24], while further expanded with scaffolding approaches [25]. Unsupervised methods can be applied to large-scale scenarios [26, 27]. Alternatively, a hybrid strategy may be used based on a set of statistical measures and syntactical and semantic information [28]. "Off the shelf" type of IR algorithms can be utilized quite successfully in some of the scenarios, especially with limited focus areas (in a medical sense) [29]. Accuracy can be further improved by analysing trigram frequencies [30] and applying graph-style algorithms [31] in context-sensitive corpus fragments.

In general, the coreference resolution methods can be classified into knowledge rich and knowledge poor. Both methods require large resources such as semantic information, syntactic annotations, or preannotated corpora of hospital transcripts from a hospital reception. Under resourced, rarer languages, like Lithuanian, usually do not have such resources available.

While up-to-date, no research has been performed to solve coreferences in Lithuanian, but many solutions have been proposed for other languages, mostly for English (Table 1). Note that the evaluation results are not directly comparable, as the authors used different corpora.

Considering languages that are more-or-less grammatically similar to Lithuanian (which is one of the Baltic languages), we summarize the related work on Latvian (only other Baltic language) and Slavic languages such as Polish, Russian, and Czech in Table 2.

(i) For Latvian, the only solution is LVCoref [45]. It is a rule-based system that uses an entity-centric model. It focuses on named entity matches (exact matches, acronyms) and uses Hobbs’ algorithm for pronouns.

(ii) For Polish, rule-based Ruler [46] for scoring of candidates uses coreferences gender/number and including (removal of nested groups) rules, lemma, and Wordnet rules for nominal expressions and pronoun rule specifically targeting pronouns. BARTEK [47] is an adaptation of BART, which was designed for English, to Polish. Mixed Polish coreferences resolution approach combines neural networks architecture with the sieve-based approach [48].

(iii) For Russian, RU-EVAL-2014 [49] was an evaluation campaign of anaphora and coreferences resolution tools that employed a wide variety of approaches. The evaluation was performed on Russian Coreference Corpus (RuCur). Machine learning approaches [50] were also used.

(iv) For Czech, coreferences are annotated in the tectogrammatical layer of Prague Dependency Treebank (PDT) and their first coreference resolution approach was rule based [51]. At first, all possible candidates are collected and then their list is narrowed down using 8 filters, and then from remaining ones closest to corefering object is selected as antecedent. Nguy et al. [52] adapted two older English language approaches to Czech language and used Decision Tree C5 for the classifier-based approach, while the ranker-based approach employed the averaged perceptron algorithm. Both approaches were trained and evaluated on PDT data with ranker-based approach providing.
better results. Treex CR [53] was developed for the Czech language and adapted to English, Russian, and German, although for Russian and German, English coreferences labels were projected, which produced notably lower results [54].

In summary, the rule-based solutions have the advantages of easier adaptability and provide comparable results when good training data are not available as is the case for Lithuania. Many of more advanced solutions cannot be fully adapted for rarer and under-resourced languages due to the lack of available linguistic resources, as is the case with Lithuanian language. For example, BART at the time supported 64 feature extractors, but due to lack of language-specific resources for the Polish language, only 13 could be utilized. The solutions that are not heavy on linguistic resources can be very useful for resource-poor languages in general.

### 3. A Rule-Based Coreference Resolution: A Lithuanian Case

#### 3.1. Definition and Framework.

Coreference resolution (or anaphora) is an expression, the interpretation of which depends on another word or phrase presented earlier in the text (antecedent). For example, "Tom has a backache. He was injured." Here the words "Tom" and "He" refer to the same entity. Without resolving the relationship between these two structures, it would not be possible to determine why Tom has the backache, nor who was injured. In such cases, semantic information would be lost.

Anaphoric objects are expressed with pronouns and cannot be independently interpreted without going back to its antecedent. In this work, such expressions are called coreferences, unless it is required to make a distinction.

Usage of such expressions can vary depending on the type and the style of the text. Here we focus on texts from medical-related domains.

#### 3.2. Conceptual Model of Coreference Resolution.

In this chapter, a conceptualization of coreference resolution is presented. A given model, which is expressed as UML class diagram (Figure 2), specifies the concepts playing a certain role in the semantic search framework (Figure 1).

| Table 1: Comparison of coreference resolution approaches. |
|----------------------------------|----------------|----------------|----------------|----------------|
| **Method**                       | **Foundation**                  | **Precision** | **Recall**     | **F1**         |
| Hobbs [32]                       | Syntactic tree with labeled nodes, syntactic rules, selection constraint rules | 0.81–0.91     | na             | na             |
| BFP [33]                         | Centering theory                | 0.49–0.90     | na             | na             |
| Left-right centering [34]        | Modified centering theory       | 0.72–0.81     | na             | na             |
| Mitkov [35]                      | POS tagger, antecedent indicators | 0.897         | na             | na             |
| RAP [36]                         | Salience factors                | 0.85–0.89     | na             | na             |
| Xrenner [37]                     | Syntactic and semantic rules    | 0.51–0.55     | 0.49–0.57      | 0.49–0.56      |
| Probabilistic [38]               | Bayesian rule                   | 0.82–0.84     | na             | na             |
| MARS [39]                        | Genetic algorithms              | 0.53–0.84     | na             | na             |
| Soon et al. [40]                 | Machine learning (decision tree C5) | 0.65–0.69     | 0.53–0.56      | 0.62           |
| ILP [41]                         | Machine learning (logistic classifier) | 0.78–0.89     | 0.47–0.58      | 0.61–0.68      |
| Wiseman et al. [42]              | Deep learning                   | 0.77          | 0.70           | 0.73           |
| Lee et al. [43]                  | Deep learning                   | 0.81          | 0.73           | 0.77           |
| Žitnik et al. [44]               | Conditional random fields       | 0.68–0.94     | 0.30–0.87      | 0.41–0.87      |

| Table 2: Comparison of coreference resolution methods for Balto-Slavic languages. |
|----------------------------------|----------------|----------------|----------------|----------------|
| **Method**                       | **Foundation**                  | **Precision** | **Recall**     | **F1**         |
| LVCoref [45]                     | Rule based, Hobbs’ algorithm    | 0.69–0.88     | 0.66–0.80      | 0.68–0.84      |
| Ruler [46]                       | Rule based                    | 0.59–0.65     | 0.50–0.75      | 0.55–0.69      |
| BARTEK [47]                      | Machine learning              | 0.58          | 0.65           | 0.61           |
| Mixed [48]                       | Deep learning, sieve based     | 0.70          | 0.68           | 0.69           |
| RU-sys1 [49]                     | Rule based, ontology           | 0.82          | 0.70           | 0.76           |
| RU-sys2 [49]                     | Rule based                    | 0.71          | 0.58           | 0.64           |
| RU-sys3 [49]                     | Rule based                    | 0.63          | 0.50           | 0.55           |
| RU-sys4 [49]                     | Statistical, ontology          | 0.54          | 0.51           | 0.53           |
| RU-sys5 [49]                     | Machine learning, semantics    | 0.58          | 0.42           | 0.49           |
| RU-sys6 [49]                     | Decision tree                 | 0.36          | 0.15           | 0.21           |
| Khadzhikaiakia and Sysoev [50]   | Machine learning              | 0.84          | 0.77           | 0.80           |
| Kučová and Žabokrtský [51]       | Rule-based filters             | 0.60          | na             | na             |
| CZ classifier [52]               | Classifier-based machine learning | 0.70–0.76   | 0.70–0.76      | 0.70–0.76      |
| CZ ranker [52]                   | Ranker-based machine learning  | 0.79          | 0.79           | 0.79           |
| Treex CR (Czech, English) [53]   | Machine learning               | na           | na             | 0.61–0.68      |
| Treex CR (Russian, German) [54]  | Machine learning, projection   | 0.50–0.64     | 0.15–0.24      | 0.25–0.34      |


**Figure 1:** An NLP pipeline of semantic search framework with coreference resolution component.

**Figure 2:** A conceptual model of a coreference resolution domain.
role in the extraction of coreferences of a certain type. The model gives us an understanding of the following:

(i) What features of text, sentence, and word help us recognize the existence of coreference (they are specified in the package Concepts of Input Flow)

(ii) What kind of text preprocessing is required

(iii) What additional resources are required for resolution of certain type coreferences (they are specified in the package Database of Public Persons and Classification of Professions)

For example, from the model provided, it is clear that, before coreference resolution starts, it is important to preprocess text and obtain the following:

(i) A text segmented into sentences and lexemes

(ii) Morphological features of lexemes identified

(iii) Named entities recognized

Text preprocessing itself is not a task of coreference resolution, so it is out of the scope of this paper.

It is worthy to mention that the model is quite abstract, language independent, and technology independent. Therefore, it is applicable not only for Lithuanian but for grammatically similar languages as well. Concepts of this model are used for the formalization of coreference resolution rules in the next section. The concepts are explained in more detail below.

The main concepts of coreference resolution are Text, Lexical_Unit, and Named_Entity. The concept Text assumes a textual document whose content should be analysed. Each text has an associated publication date, which is important for solving coreferences. Each text consists of at least one Lexical_Unit, which includes paragraphs, sentences, words, and punctuations, classified into the Sentence and Lexeme categories. Lexeme assumes lexical units such as words, punctuations, and numbers. Each lexeme is characterized by a lemma and a part of speech, and some of them (nouns and pronouns) by grammatical gender and number. The lexeme could be specialized by POS category: Noun, Pronoun, and Other_Part_Of_Speech. Special cases of Other_Part_Of_Speech are Comma and Conjunction, which are required for the description of conditions of some coreference resolution rules.

A Named_Entity concept defines an object to whom pronouns or certain nouns can refer. NER algorithms usually recognize three types of entities: a person (Person_NE), an organization (Organization_NE), and a location (Location_NE). The named entities of a person type require special attention a person can be mentioned not only using pronouns but also using a position he/she holds (Position_Held) and a professional name (Profession). Additional information about a person could help resolving such coreferences more precisely. As an example, source of such information could be a Database of Public Person, which includes Known_Person—a well-known person mentioned as Person_NE in the text. The output of a coreference resolution algorithm is a Coreference—a relationship between coreferents. For each coreference, its type (nominal and pronominal), subtype (relative pronoun and noun repetition), position (points backward, forward, or irrelevant in case of repetitions), and group (is singular, refers to the coreference group or is ambiguous) are specified. Each referent refers at least to one coreferent (a concept Mention). Each Mention starts at a certain position in the text, is of a certain length, and fits at least one Lexeme. Some of them can fit a certain Named_Entity.

3.3. Coreference Resolution Algorithms. The decision table with guidelines for the application of the certain resolution algorithm is shown as Figure 3. The conditions are checked consecutively on every lexeme in the text, and, if the condition is satisfied, a corresponding algorithm is activated. For example, if C2 condition is met then immediately A1 algorithm is activated.

For resolution of a specific type of references, we propose the following algorithms:

(i) A1: specific rules resolution algorithm for resolution of certain usage of pronouns

(ii) A2: general pronoun resolution algorithm which focuses on the cases where pronouns refer to nouns (or noun phrases) that are recognized as named entities of “person” class

(iii) A3: PRA (partial, repetition, and acronym) resolution algorithm for resolution of nouns recognized as named entities and their repeated usage in the same text

(iv) A4: HHS (hypernym, hyponym, synonymous) resolution algorithm for resolution of nouns recognized as profession names including their synonyms and hypernyms/hyponyms

(v) A5: feature resolution algorithm for resolution of nouns that represent certain feature (at the moment only public position being held) of the named entity of a person

The coreference resolution starts from the sequential analysis of each lexeme looking for a certain type of pronoun and noun. Depending on identified features of lexeme, a decision about further analysis is taken. The decision table (Figure 3) summarizes conditions for the application of the certain resolution algorithm. The conditions are listed in the upper left quadrant; the decision alternatives are listed in the lower left quadrant. The upper right quadrant shows the possible alternatives for the conditions of the corresponding row. In the upper right quadrant, the answer “✓” stands for “not relevant.” In the lower right quadrant, “✓” means that the algorithm should be applied and “✗” means that it should not be applied.

The idea is that the pronoun-related coreferences should be solved first sequentially by checking the conditions C1, C2, and C3. Then a noun-related coreference resolution should start by sequentially checking the conditions C4, C6, and C7.

3.4. Formal Description of Coreference Resolution Algorithms. First-order logic (FOL) formulas are employed to define the main conditions the algorithms should check when resolving coreferences. The concepts of the coreference
The resolution model (Figure 2) became the predicates or constants in the FOL formulas: the classes became the unary predicates of the same name as class; the associations between classes—the binary predicates of the same name as association; the attributes of classes—the binary predicates of the same name as attribute plus verb “has” at the beginning; and the literals of enumerations—constants.

The algorithms follow the grammar rules of the Lithuanian language which are based on the analysis of morphological features of lexemes and their order in the sentence and text. Examples of Lithuanian language sentences were translated into English as closely as possible. All proper names were changed to generic abbreviations to comply with GDPR.

3.4.1. A1: Specific Rules Resolution. In some cases, there exists a rather rigid structure for pronoun usage and it can be easily defined by using specific rules, for example,

(i) [LT] Šiandien buvo atėjęs vyras [noun], kuris [pronoun] skundesi nugaros skausmu.

(ii) [LT] Šiandien buvo atėjęs vyras [noun], su [preposition] kuriuo [pronoun] aptarėme nugaros skausmą.

Both examples are similar in their construction: [noun] [comma] [optional preposition] [specific pronoun]. In both cases, pronoun “kuriuo” refers to the noun “vyras.” In the first example, we do not have an optional preposition “su,” while we have it in the second one.

A condition for the existence of such reference formally is defined as follows:

For every sentence $s$ of text $t$ and for every “Relative” type pronoun $p$, which is contained in the sentence $s$ and has a start position $sp_1$, is of length $ln_1$, follows comma $c$ or follows prepositional lexeme $l_1$, which follows comma $c$, and for every noun $l_2$, which has a start position $sp_2$, is of length $ln_2$, precedes comma $c$, is of the same gender $g$ and of the same number $n$ as the pronoun $p$, the only one coreference relation $r$, which is resolved in text $t$, is of “Pronominal” type, “Relative” subtype, “Backward” position and “Single” group between the pronoun $p$ and the noun $n$, its referent starts at position $sp_1$ and has length $ln_1$, and which fits only one lexeme $p$ and refers to only one mention $m$, which starts at position $sp_2$, has length $ln_2$, and fits only one lexeme $l_2$, exists (Rule 1).

Rule 1: $\forall t, s, p, l_1, l_2, g, n, sp_1, sp_2, ln_2. \text{[Text(t)} \wedge \text{Sentence(s)} \wedge \text{consists of(t, s)} \wedge \text{Pronoun(p)} \wedge \text{contains(s, p)} \wedge \text{has type(p, Relative)} \wedge \text{has start position(p, sp_1)} \wedge \text{has length(p, ln_1)} \wedge \text{Comma(c)} \wedge \text{follows(p, c)} \vee \text{Lexeme(l_1)} \wedge \text{has pos(l_1, Preposition)} \wedge \text{follows(l_1, c)} \wedge \text{follows(p, l_1)} \wedge \text{Noun(l_2)} \wedge \text{follows(l_2, c)} \wedge \text{has gender}(p, g) \wedge \text{has gender}(l_2, g) \wedge \text{has number}(p, n) \wedge \text{has number}(l_2, n) \wedge \text{has start position}(l_2, sp_2) \wedge \text{has length}(l_2, ln_2) \rightarrow \exists r \exists m. \text{[Coreference(r)} \wedge \text{resolved in(r, t)} \wedge \text{has type(r, Pronominal)} \wedge \text{has subtype(r, Relative)} \wedge \text{has position(r, Backward)} \wedge \text{has group(r, Single)} \wedge \text{has start position(r, sp_1)} \wedge \text{has length}(r, ln_1) \wedge \text{fits(r, p)} \wedge \text{Mention(m)} \wedge \text{refers to(r, m)} \wedge \text{has start position(m, sp_2)} \wedge \text{has length}(m, ln_2) \wedge \text{fits(m, l_2)]}$

The relative pronoun might be plural and refer to multiple singular (or multiple plural) nouns.
In this case, a plural pronoun “kurių” is referring to four singular nouns that have different genders. The previous rule would not be able to solve such coreference. For this case, the construction would be: [noun] (comma) [noun] (comma) [noun] [comma] [pronoun] shared similar injuries.

For such case, a special condition must be defined:

For every sentence \( s \) in text \( t \) and for every “Relative” type pronoun \( p \) of “Plural” number, which is contained in the sentence \( s \) and has a start position \( s1 \), of length \( l1 \), follows comma \( c1 \) or follows prepositional lexeme \( l \), which follows comma \( c1 \), and for every noun \( n1 \), which precedes comma \( c1 \), has a start position \( s2 \), is of length \( l2 \), follows conjunction \( j \), and for every noun \( n2 \), which precedes conjunction \( j \), has a start position \( s3 \), is of length \( l3 \), and has length \( l1 \), fits only one lexeme \( p \), refers to only one mention \( m1 \), which starts at position \( s2 \), has length \( l2 \), and fits noun \( n1 \), refers to only one mention \( m2 \), which starts at position \( s3 \), has length \( l3 \), and fits only one noun \( n2 \), and refers at least to one mention \( m3 \), which starts at position \( s4 \), has length \( l4 \), and fits noun \( n4 \), exists (Rule 2).

Rule 2: \( \forall t, s, p, l, c1, n1, s1, p1, l1, s2, j, n2, s3, l3, j, n3, s4, \) \( \exists m1, m2, m3, m4 \)

Rule 3: \( \forall t, s, n, s1, p1, l1, s2, j, n2, s3, l3, s4, l4 \)
contains\((s_1, p) \wedge \neg \text{has\_type}(p, \text{Demonstrative}) \wedge \text{has\_gender}(p, g) \wedge \text{has\_number}(p, n) \wedge \text{has\_start\_position}(p, sp1) \wedge \text{length}(p, ln1) \wedge \text{Person\_NE}(e1) \wedge \text{includes}(s_1, e1) \wedge \text{Lexeme}(l) \wedge \text{expressed\_by}(e1, l) \wedge \text{has\_gender}(e1, g) \wedge \text{has\_number}(e1, n) \wedge \text{has\_start\_position}(e1, sp2) \wedge \text{length}(e1, ln2) \wedge \text{sp2} < \text{sp2} \wedge \neg (\exists e2, e3, sp3, sp4, (e1 \neq e2 \wedge e1 \neq e3 \wedge e2 \neq e3 \wedge \text{Person\_NE}(e2) \wedge \text{includes}(s_1, e2) \wedge \text{has\_gender}(e2, g) \wedge \text{has\_number}(e2, n) \wedge \text{has\_start\_position}(e2, sp3) \wedge \text{Person\_NE}(e3) \wedge \text{includes}(s_1, e3) \wedge \text{has\_gender}(e3, g) \wedge \text{has\_number}(e3, n) \wedge \text{has\_start\_position}(e3, sp4) \wedge \text{sp2} > \text{sp3} \wedge \text{sp4} > \text{sp2})) \rightarrow \exists r \exists m. (\text{Coreference}(r) \wedge \text{resolved\_in}(r, t) \wedge \text{has\_type}(r, \text{Pronominal}) \wedge \text{has\_subtype}(r, \text{General}) \wedge \text{has\_position}(r, \text{Backward}) \wedge \text{has\_group}(r, \text{Single}) \wedge \text{has\_start\_position}(r, sp1) \wedge \text{length}(r, ln1) \wedge \text{fits}(r, p) \wedge \text{Mention}(m) \wedge \text{refers\_to}(r, t) \wedge \text{has\_start\_position}(m, sp2) \wedge \text{length}(m, ln2) \wedge \text{fits}(m, e1) \wedge \text{fits}(m, l))]

The second alternative describes a case when a pronoun \(p\) refers to the named entity in the previous sentence \(s_2\):

For each text's \(t\) sentence \(s_1, s_2\), where \(s_1\) follows \(s_2\), and pronoun \(p\) not of Demonstrative type that is contained in sentence \(s_1\) and has gender \(g\), number \(n\), start position \(sp1\) and length of \(ln1\), and named entity \(e1\) that is contained in sentence \(s_2\), is expressed by lexeme \(l\), and has gender \(g\), number \(n\), start position \(sp2\) and is of length \(ln2\), and is closer to pronoun \(p\) than possible named entities \(e2\) and \(e3\) (\(sp2\) higher than \(sp3\) and \(sp4\)), the only one coreference relation \(r\), which is resolved in text \(t\), is of “Pronominal” type, “Relative” subtype, “Backward” position and “Single” group between the pronoun \(p\) and the named entity \(e1\), its referent starts at position \(sp1\) and has length \(ln1\), and which fits only one pronoun \(p\) and refers to only one mention \(m\), which starts at position \(sp2\), has length \(ln2\), and fits only one named entity \(e1\), exists (Rule 5).

Rule 5: \(\forall t, s_1, s_2, s_3, p, l, e_1, g, n, sp1, ln1, sp2, ln2. [\text{Text}(t) \wedge \text{Sentence}(s_1) \wedge \text{Sentence}(s_2) \wedge \text{Sentence}(s_3) \wedge \text{consists\_off}(t, s_1) \wedge \text{consists\_off}(t, s_2) \wedge \text{consists\_off}(t, s_3) \wedge \text{contains}(s_1, p) \wedge \neg \text{has\_type}(p, \text{Demonstrative}) \wedge \text{has\_gender}(p, g) \wedge \text{has\_number}(p, n) \wedge \text{has\_start\_position}(p, sp1) \wedge \text{length}(p, ln1) \wedge \text{Person\_NE}(e1) \wedge \text{includes}(e2, e1) \wedge \text{Lexeme}(l) \wedge \text{expressed\_by}(e1, l) \wedge \text{has\_gender}(e1, g) \wedge \text{has\_number}(e1, n) \wedge \text{has\_start\_position}(e1, sp2) \wedge \text{length}(e1, ln2) \wedge \neg (\exists e2, e3, sp3, sp4, (e1 \neq e2 \wedge e1 \neq e3 \wedge e2 \neq e3 \wedge \text{Person\_NE}(e2) \wedge \text{includes}(e2, e1) \wedge \text{has\_gender}(e2, g) \wedge \text{has\_number}(e2, n) \wedge \text{has\_start\_position}(e2, sp3) \wedge \text{Person\_NE}(e3) \wedge \text{includes}(e3, e1) \wedge \text{has\_gender}(e3, g) \wedge \text{has\_number}(e3, n) \wedge \text{has\_start\_position}(e3, sp4) \wedge \text{sp2} > \text{sp3} \wedge \text{sp4} > \text{sp2})) \rightarrow \exists r \exists m. (\text{Coreference}(r) \wedge \text{resolved\_in}(r, t) \wedge \text{has\_type}(r, \text{Pronominal}) \wedge \text{has\_subtype}(r, \text{General}) \wedge \text{has\_position}(r, \text{Backward}) \wedge \text{has\_group}(r, \text{Single}) \wedge \text{has\_start\_position}(r, sp1) \wedge \text{length}(r, ln1) \wedge \text{fits}(r, p) \wedge \text{Mention}(m) \wedge \text{refers\_to}(r, t) \wedge \text{has\_start\_position}(m, sp2) \wedge \text{length}(m, ln2) \wedge \text{fits}(m, e1) \wedge \text{fits}(m, l)]\)

Another example presents a case when a coreferent of the pronoun “man” (in English, “for me”) is in the following sentence:

(i) \(\text{[LT]}\) Pastebėtina, kad ligiginėse apsilankė 10 mln. pacientų, nepaisant to, kad 2016 m. jų buvo 4% mažiau (apsilankė beveik 9,5 mln.). Tai labiausiai lėmė skaitmeninių pašalų padidinimas: “Kiek man [pronoun] teko analizuoti, padidinus skaitmenines pašlasas tik nedidele dalis Lietuvas [location noun] ligoninių sumažėjo etatų ar ateido darbuotojus, o tai lėmė nemaža papildoma indėlį i pašlasos kokybę” teige J. Jonaitis [person noun phrase].

[EN] It is noteworthy that the total hospital patient count has reached 10 million, even though in 2016 the number was less than 4% (around 9.5 million patients). This was influenced by the digitization of e-health services. “As far as I [pronoun] had analysed, only a small part of Lithuanian [location noun] hospitals have reduced their posts or dismissed employees, but digitization of services has led to a
significant additional contribution to the quality of service,” said J. Jonaitis [person noun phrase].

If the algorithm does not find any named entities moving backwards, it moves back to pronoun and proceeds forward. The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm continues moving forward until it locates “J. Jonaitis” as a postcedent of the corefering both female and male persons), only their grammatical gender of the pronoun “man” is ambiguous (it can refer to

The algorithm ignores demonstrative pronouns because they are often used to refer to entities that are not present in the written text. Such pronouns do not carry any additional semantic information and do not refer to any noun phrase. They are used mostly for syntactic reasons and due to that they are usually ignored in coreference resolution.

3.4.3. A3: PRA Resolution. This algorithm is based on exact (or partial) string matches and several rules for acronyms. Once a first named entity that can be matched with an initial named entity is found, then the algorithm stops to keep annotations simple: B — A, C — B and D — C. This allows the formation of the coreference chains linking all mentions of the same entity in a text that can be later reused for semantic analysis, for example,

(i) [LT] Tomaitis [named entity] pateko į avarija. Po pietų Tomaiti [named entity] išvežė į operacine. [EN] Tomaitis [named entity] got into a car accident. In the afternoon, Tomaitis [named entity] has been taken to a surgery room.

In this example, two mentions of the same entity are made: “Tomaitis” and “Tomaitį.” They are of different cases, but their lemmas are identical. A condition for the existence of such reference formally is defined as follows:

For each text’s t sentence s1 that includes named entity e1, that has start position sp1 and is of length ln1, which is expressed by

The algorithm ignores demonstrative pronouns because they are often used to refer to entities that are not present in the written text. Such pronouns do not carry any additional semantic information and do not refer to any noun phrase. They are used mostly for syntactic reasons and due to that they are usually ignored in coreference resolution.

3.4.3. A3: PRA Resolution. This algorithm is based on exact (or partial) string matches and several rules for acronyms. Once a first named entity that can be matched with an initial named entity is found, then the algorithm stops to keep annotations simple: B — A, C — B and D — C. This allows the formation of the coreference chains linking all mentions of the same entity in a text that can be later reused for semantic analysis, for example,


[EN] Tomaitis [named entity] got into a car accident. In the afternoon, Tomaitis [named entity] has been taken to a surgery room.

In this example, two mentions of the same entity are made: “Tomaitis” and “Tomaitį.” They are of different cases, but their lemmas are identical. A condition for the existence of such reference formally is defined as follows:

For each text’s t sentence s1 that includes named entity e1, that has start position sp1 and is of length ln1, which is expressed by

The second alternative describes the case when the pronoun p refers to the named entity in the following sentence s4:

For each text’s t sentence s4, where s4 follows s1, and pronoun p not of Demonstrative type that is contained in sentence s1 and has gender g, number n, start position sp1 and length of ln1, and named entity e1 that is contained in sentence s2, is expressed by lexeme l, and has gender g, number n, start position sp2 and is of length ln2, and is closer to pronoun p than possible named entities e2 and e3 (sp2 higher than sp3 and sp4), the only coreference relation r, which is resolved in text t, is of “Pronominal” type, “Relative” subtype, “Backward” position and “Single” group between the pronoun p and the named entity e1, its referent starts at position sp1 and has length ln1, and which fits only one pronoun p and refers to only one mention m, which starts at position sp2, has length ln2, and fits only one named entity e1, exists (Rule 7).

Rule 7: ∀t, s1, s2, p, l, e1, g, n, sp1, ln1, sp2, ln2. [Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(s1, s2) ∧ follows(s2, s1) ∧ Pronoun(p) ∧ contains(s1, p) ∧ ¬ has_type(p, Demonstrative) ∧ has_gender(p, g) ∧ has_number(p, n) ∧ has_start_position(p, sp1) ∧ has_length(p, ln1) ∧ Person_NE(e1) ∧ includes(s2, e1) ∧ Lexeme(l) ∧ expressed_by(e1, l) ∧ has_gender(e1, g) ∧ has_number(e1, n) ∧ has_start_position(e1, sp2) ∧ has_length(e1, ln2) ∧ ¬(∃e2, e3, sp3, sp4. (e1≠e2 ∧ e1≠e3 ∧ e2≠e3 ∧ Person_NE(e2) ∧ includes(s2, e2) ∧ has_start_position(e2, sp3) ∧ Person_NE(e3) ∧ includes(s1, e3) ∧ has_start_position(e3, sp4) ∧ sp2<sp3 ∧ sp4<sp2)) → ∃r. ∃m. [Coreference(r) ∧ resolved_in(r, t) ∧ has_type(r, Pronominal) ∧ has_subtype(r, General) ∧ has_position(r, Forward) ∧ has_group(r, Single) ∧ has_start_position(r, sp1) ∧ has_length(r, ln1) ∧ fits(r, p) ∧ Mention(m) ∧ refers_to(r, t) ∧ has_start_position(m, sp2) ∧ has_length(m, ln2) ∧ fits(m, e1) ∧ fits(m, l)]]

The algorithm ignores demonstrative pronouns because they are often used to refer to entities that are not present in the written text. Such pronouns do not carry any additional semantic information and do not refer to any noun phrase. They are used mostly for syntactic reasons and due to that they are usually ignored in coreference resolution.

3.4.3. A3: PRA Resolution. This algorithm is based on exact (or partial) string matches and several rules for acronyms. Once a first named entity that can be matched with an initial named entity is found, then the algorithm stops to keep annotations simple: B — A, C — B and D — C. This allows the formation of the coreference chains linking all mentions of the same entity in a text that can be later reused for semantic analysis, for example,


[EN] Tomaitis [named entity] got into a car accident. In the afternoon, Tomaitis [named entity] has been taken to a surgery room.
lexeme \( l_2 \) that has lemma \( l_1 \), the only one coreference relation \( r \), which is resolved in text \( t \), is of “Nominal” type, “Repetition” position and “Single” group between the noun \( n_1 \) and the noun \( n_2 \), its referent starts at position \( sp_1 \) and has length \( ln_1 \), and which fits only one noun \( n_1 \) and refers to only one mention \( m \), which starts at position \( sp_2 \), has length \( ln_2 \), and fits only one noun \( n_2 \), exists (Rule 8).

Rule 8: \( \forall t, s_1, s_2, e_1, e_2, s_1, n_1, s_2, n_2, \text{[Text}(t) \wedge \text{Sentence}(s_1) \wedge \text{Sentence}(s_2) \wedge \text{consists_of}(t, s_1) \wedge \text{consists_of}(t, s_2) \wedge \text{Named_Entity}(e_1) \wedge \text{includes}(s_1, e_1) \wedge \text{has_start_position}(e_1, sp_1) \wedge \text{has_length}(e_1, ln_1) \wedge \text{Named_Entity}(e_2) \wedge \text{includes}(s_2, e_2) \wedge \text{has_start_position}(e_2, sp_2) \wedge \text{has_length}(e_2, ln_2) \wedge \text{\#}(l_1, l_2) \wedge \text{Lexeme}(l_1) \wedge \text{Lexeme}(l_2) \wedge \text{expressed_by}(e_1, l_1) \wedge \text{expressed_by}(e_2, l_2) \wedge \text{has_lemma}(l_1, l) \wedge \text{has_lemma}(l_2, l) \implies \exists ! r \exists m. \text{[Coreference}(r) \wedge \text{resolved_in_in}(r, t) \wedge \text{has_type}(r, \text{Nominal}) \wedge \text{has_subtype}(r, \text{Repetition}) \wedge \text{has_position}(r, \text{Irrelevant}) \wedge \text{has_group}(r, \text{Single}) \wedge \text{has_start_position}(r, sp_1) \wedge \text{has_length}(r, ln_1) \wedge \text{fits}(r, l_1) \wedge \text{fits}(r, e_1) \wedge \text{Mention}(m) \wedge \text{refers_to}(r, t) \wedge \text{has_start_position}(m, sp_2) \wedge \text{has_length}(m, ln_2) \wedge \text{fits}(m, l_2) \wedge \text{fits}(m, e_2)]\]

Acronym rules vary depending on the type of named entity (currently persons, locations, and organizations are covered).

3.4.4. A4: HHS Resolution. This algorithm is based on profession classification. It attempts to resolve the use of synonyms and hypernyms/hyponyms.


[EN] Doctors [noun referring to profession] complain about heavy workload. Surgeons’ [noun referring to profession] workload is the greatest.

The algorithm determines that “Doctor” in professions classification is a hyponym of “Surgeon,” they also agree in gender and number; therefore, the algorithm adds their pair to annotations. Conditions for the existence of such reference formally are defined as follows:

For each text’s sentence \( s_1 \) that has a profession’s \( p \) name \( v_1 \), which is expressed by noun \( n_1 \) that has gender \( g \), number \( m \), start position \( sp_1 \) and is of length \( ln_1 \), and for each same text’s sentence \( s_2 \) that has same profession’s \( p \) name \( v_2 \) expressed by noun \( n_2 \) that has gender \( g \), number \( m \), start position \( sp_2 \) and is of length \( ln_2 \), the only one coreference relation \( r \), which is resolved in text \( t \), is of “Nominal” type, “Synonym” subtype, “Irrelevant” position and “Single” group between the noun \( n_1 \) and the noun \( n_2 \), its referent starts at position \( sp_1 \) and has length \( ln_1 \), and which fits only one noun \( n_1 \) and refers to only one mention \( m \), which starts at position \( sp_2 \), has length \( ln_2 \), and fits only one noun \( n_2 \), exists (Rule 9).

Rule 9: \( \forall t, s_1, s_2, n_1, n_2, sp_1, ln_1, sp_2, ln_2, v_1, v_2, p_1, p_2. \text{[Text}(t) \wedge \text{Sentence}(s_1) \wedge \text{Sentence}(s_2) \wedge \text{consists_of}(t, s_1) \wedge \text{consists_of}(t, s_2) \wedge \text{Noun}(n_1) \wedge \text{contains}(s_1, n_1) \wedge \text{has_start_position}(n_1, sp_1) \wedge \text{has_length}(n_1, ln_1) \wedge \text{Noun}(n_2) \wedge \text{contains}(s_2, n_2) \wedge \text{has_start_position}(n_2, sp_2) \wedge \text{has_length}(n_2, ln_2) \wedge n_1 \neq n_2 \wedge \text{Profession}(p_1) \wedge \text{Profession}(p_2) \wedge p_1 \neq p_2 \wedge \text{Profession_Name}(v_1) \wedge \text{Profession_Name}(v_2) \wedge \text{express}(n_1, v_1) \wedge \text{express}(n_2, v_2) \wedge \text{describes}(v_1, p_1) \wedge \text{describes}(v_2, p_2) \wedge \text{broadens}(p_2, p_1) \wedge \text{broadens}(p_1, p_2) \wedge \text{has_gender}(n_1, g) \wedge \text{has_gender}(n_2, g) \wedge \text{has_number}(n_1, n) \wedge \text{has_number}(n_2, n) \implies \exists ! r \exists m. \text{[Coreference}(r) \wedge \text{resolved_in}(r, t) \wedge \text{has_type}(r, \text{Nominal}) \wedge \text{has_subtype}(r, \text{Irrelevant}) \wedge \text{has_group}(r, \text{Single}) \wedge \text{has_start_position}(r, sp_1) \wedge \text{has_length}(r, ln_1) \wedge \text{fits}(r, n_1) \wedge \text{Mention}(m) \wedge \text{refers_to}(r, t) \wedge \text{has_start_position}(m, sp_2) \wedge \text{has_length}(m, ln_2) \wedge \text{fits}(m, n_2)]]\]

An example of synonym is given as follows:


[EN] From now J. Jonaitis was the head surgeon [noun referring to profession] of operation. Unfortunately, the last operation of chief surgeon [noun referring to profession] was unsuccessful.

Both “head surgeon” and “chief surgeon” are synonymous; therefore, the condition for the existence of such reference formally could be defined as follows:

For each text’s sentence \( s_1 \) that has profession’s \( p \) name \( v_1 \), which is expressed by noun \( n_1 \) that has gender \( g \), number \( m \), start position \( sp_1 \) and is of length \( ln_1 \), and for each same text’s sentence \( s_2 \) that has same profession’s \( p \) name \( v_2 \) expressed by noun \( n_2 \) that has gender \( g \), number \( m \), start position \( sp_2 \) and is of length \( ln_2 \), the only one coreference relation \( r \), which is resolved in text \( t \), is of “Nominal” type, “Synonym” subtype, “Irrelevant” position and “Single” group between the noun \( n_1 \) and the noun \( n_2 \), its referent starts at position \( sp_1 \) and has length \( ln_1 \), and which fits only one noun \( n_1 \) and refers to only one mention \( m \), which starts at position \( sp_2 \), has length \( ln_2 \), and fits only one noun \( n_2 \), exists (Rule 10).
3.4.5. A5: Feature Resolution. This algorithm at the time attempts to resolve only those cases when a person is being referred to by his public post (feature) that he holds, other types of features are not currently resolved, for example.

(i) [LT] Ką rekomenduoja S. Suskelis [person noun phrase]? Aptarkime kardiologo [noun referring to held position] siūloma gydymo planą.

[EN] What does S. Suskelis [person noun phrase] recommend? Let’s discuss treatment plan proposed by the cardiologist [noun referring to held position]. Here a noun “cardiologist” is selected, the algorithm moves backwards till it reaches “S. Suskelis” and checks the knowledge base if at the time of the publication of the medical record he has held the position of the cardiologist.. Since “he holds it” the algorithm checks if “S. Suskelis” and “cardiologist” agree in gender and number. They agree, and their pair is added to annotation as a feature reference. A condition for the existence of such reference formally is defined as follows:

For each text’s t sentence s1 that has known person k, who during publication date d had certain position h (publication date d is same or later than position h start date fd and same or earlier than position h end date td), mention as named entity e, that has a start position sp1 and is of length ln1, and for each same text’s t sentence s2 mentioned noun n, that has a start position sp2 and is length ln2, which is mentioned after named entity e (noun n has a higher start position sp2 than named entity’s sp1), whose lemma l matches with position’s h lemma l, number is Singular and gender g matches known person’s k gender g, the only one coreference relation r, which is resolved in text t, is of “Nominal” type, “Feature” subtype, “Backward” position and “Single” group between the noun n and the named entity e, its referent starts at position sp2 and has length ln2, and which fits only one noun n and refers to only one mention m, which starts at position sp1, has length ln1, and fits only one named entity e, exists (Rule 11).

Rule 11: ∀t, s1, s2, n, k, h, l, d, fd, td, g, sp1, sp2, ln1, ln2.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ Person_NE(e) ∧ includes(s1, e) ∧ Noun(n) ∧ contains(s2, n) ∧ Known_Person(k) ∧ mentioned_as(k, e) ∧ Position Held(h) ∧ holds(h, k) ∧ has_lemma(h, l) ∧ has_lemma(n, l) ∧ has_publication_date(t, d) ∧ has_from_date(h, fd) ∧ has_to_date(h, td) ∧ td≤d ∧ td≥d ∧ has_gender(k, g) ∧ has_gender(n, g) ∧ has_number(n, Singular) ∧ has_start_position(e, sp1) ∧ has_start_position(n, sp2) ∧ sp1<sp2 ∧ has_length(e, ln1) ∧ has_length(n, ln2) → ∃r! ∃m. [Coreference(r) ∧ resolved_in(r, t) ∧ has_type (r, Nominal) ∧ has sub-type (r, Feature) ∧ has_position(r, Backward) ∧ has_group(r, Single) ∧ has_start_position(r, sp2) ∧ has_length(r, ln2) ∧ fits(r, e) ∧ Mention(m) ∧ refers_to(r, t) ∧ has_start_position(m, sp1) ∧ has_length(m, ln1) ∧ fits(m, n)]]

In this case, it is also relevant to track if coreference is pointing backward or forward. We can rewrite the same example and switch a known person with his positions:


[EN] Who is my family doctor [noun referring to held position]? T. Tomaitis [person noun phrase] is very experienced doctor assigned to you.

As a result, sp1 is higher than sp2 and coreference has different position constant values:

For each text’s t sentence s1 that has known person k, who during publication date d had certain position h (publication date d is same or later than position h start date fd and same or earlier than position h end date td), mention as named entity e, that has a start position sp1 and is of length ln1, and for each same text’s t sentence s2 mentioned noun n, that has a start position sp2 and is length ln2, which is mentioned after named entity e (noun n has a lower start position sp2 than named entity’s sp1), whose lemma l matches with position’s h lemma l, number is Singular and gender g matches known person’s k gender g, the only one coreference relation r, which is resolved in text t, is of “Nominal” type, “Feature” subtype, “Forward” position and “Single” group between the noun n and the named entity e, its referent starts at position sp2 and has length ln2, and which fits only one noun n and refers to only one mention m, which starts at position sp1, has length ln1, and fits only one named entity e, exists (Rule 12).

Rule 12: ∀t, s1, s2, n, k, h, l, d, fd, td, g, sp1, sp2, ln1, ln2.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ Person_NE(e) ∧ includes(s1, e) ∧ Noun(n) ∧ contains(s2, n) ∧ Known_Person(k) ∧ mentioned_as(k, e) ∧ Position Held(h) ∧ holds(h, k) ∧ has_lemma(h, l) ∧ has_lemma(n, l) ∧ has_publication_date(t, d) ∧ has_from_date(h, fd) ∧ has_to_date(h, td) ∧ td≤d ∧ td≥d ∧ has_gender(k, g) ∧ has_gender(n, g) ∧ has_number(n, Singular) ∧ has_start_position(e, sp1) ∧ has_start_position(n, sp2) ∧ sp1>sp2 ∧ has_length(e, ln1) ∧ has_length(n, ln2) → ∃r! ∃m. [Coreference(r) ∧ resolved_in(r, t) ∧ has_type (r, Nominal) ∧ has sub-type (r, Feature) ∧ has_position(r, Forward) ∧ has_group(r, Single) ∧ has_start_position(r, sp2) ∧ has_length(r, ln2) ∧ fits(r, e) ∧ Mention(m) ∧ refers_to(r, t) ∧ has_start_position(m, sp1) ∧ has_length(m, ln1) ∧ fits(m, n)]]

4. Results and Evaluation

The coreference resolution algorithms and rules presented in Section 3 were implemented as a separate component and integrated into the semantic search framework NLP pipeline
(Figure 1) because it requires lexical, morphological, and NE annotations of the text should be analysed. Solutions for other languages should not follow the same NLP pipeline architecture. But a supply of coreference resolution component with lexical, morphological, and NE information of the text must be ensured.

Coreference resolution for Lithuanian was implemented using Java programming language and JSON data format for annotation storage. But the proposed approach is not technology dependent, and for other languages, it can be implemented on any other platform.

The evaluation was performed by analysing 100 articles that have been preannotated and are available in our Lithuanian Language Coreference Corpus [55], in addition to the transcribed records of medical reception, which we cannot disclose due to the privacy requirements.

For evaluation, we used precision, recall, and F1 metrics. Recall R is the ratio of correctly resolved anaphoric expressions C to the total number of anaphoric expressions T. Precision P is the ratio of correctly resolved anaphoric expressions C to the number of resolved anaphoric expressions F. F1 is a harmonic mean of P and R:

\[ R = \frac{C}{T} \]
\[ P = \frac{C}{F} \]
\[ F1 = \frac{2 \times R \times P}{R + P} \]

Table 3: Experiment results.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>T (actual number of solvable expressions)</th>
<th>F (number of solvable expressions resolved by the algorithm)</th>
<th>C (number of solvable expressions correctly resolved by the algorithm)</th>
<th>R (recall)</th>
<th>P (precision)</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>General purpose (GP)</td>
<td>648</td>
<td>248</td>
<td>202</td>
<td>0.311</td>
<td>0.814</td>
<td>0.451</td>
</tr>
<tr>
<td>GP + specific usage rules (SUR)</td>
<td>648</td>
<td>277</td>
<td>223</td>
<td>0.344</td>
<td>0.805</td>
<td>0.482</td>
</tr>
<tr>
<td>GP + SUR + PRA</td>
<td>648</td>
<td>330</td>
<td>267</td>
<td>0.412</td>
<td>0.809</td>
<td>0.546</td>
</tr>
<tr>
<td>GP + SUR + PRA + HHS</td>
<td>648</td>
<td>343</td>
<td>274</td>
<td>0.422</td>
<td>0.799</td>
<td>0.553</td>
</tr>
<tr>
<td>GP + SUR + PRA + HHS + feature</td>
<td>648</td>
<td>371</td>
<td>289</td>
<td>0.446</td>
<td>0.779</td>
<td>0.567</td>
</tr>
</tbody>
</table>

(Figure 1) 5. Conclusion

Medical entity recognition and coreferencing are difficult tasks in Lithuanian natural language processing (NLP). We proposed the coreference resolution approach for the Lithuanian language. The coreference resolution algorithm depends on morphological and named entity recognition (NER) annotations and preexisting databases. Due to the proposed approach being detached from specific implementation and rules being formalized, it would not be difficult to adapt it for grammatically similar languages. Our novelty is the ability to process coreferences with minimal linguistic resources, which are very important to consider in linguistic applications for under-resourced and endangered languages. While the proposed method provides encouraging results, when analysing transcribed medical records and other corpora, and they are comparable to the results achieved by other authors applying different resolution approaches on other languages, it has certain limitations: it is domain specific and is able to resolve only a subset of coreference types, while the relatively small dataset was used for experiments. Nevertheless, we hope that our method can contribute to the sustainable development of the NLP-powered online healthcare services in Lithuania.

Data Availability

The dataset used in this research is available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


