KAUNAS UNIVERSITY OF TECHNOLOGY

VOLDEMARAS ŽITKUS

COREFERENCE RESOLUTION FOR LITHUANIAN LANGUAGE

Doctoral dissertation
Technological Sciences, Informatics Engineering (T 007)

Kaunas, 2024

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Informatics, Department of Information Systems during the period of 2013–2019 and 2023–2024.
The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

The dissertation defended externally

Scientific Consultant:
Prof. Dr. Rita BUTKIENĖ (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007).

Edited by: English language editor Dovilė Blaudžiūnienė (Publishing House *Technologija*), Lithuanian language editor Aurelija Gražina Rukšaitė (Publishing House *Technologija*).

Dissertation Defence Board of Informatics Engineering Science Field:
Prof. Dr. Tomas SKERSYS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007) – **chairperson**;
Prof. Dr. Nikolaj GORANIN (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);
Prof. Dr. Diana KALIBATIENĖ (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);
Prof. Dr. Tomas KRILAVIČIUS (Vytautas Magnus University, Natural Sciences, Informatics, N 009);
Dr. Zbigniew MARSZAŁEK (Silesian University of Technology, Poland, Natural Sciences, Informatics, N 009).

The public defense of the dissertation will be held at 10:00 a.m on 17 June, 2024 at the public meeting of Dissertation Defense Board of Informatics Engineering Science Field in Rectorate Hall at Kaunas University of Technology.

Address: K. Donelaičio 73-402, LT-44249 Kaunas, Lithuania.
Phone: (+370) 608 28 527; e-mail doktorantura@ktu.lt

The doctoral dissertation was sent out on 17 May, 2024.
The doctoral dissertation is available on the internet http://ktu.edu and at the libraries of Kaunas University of Technology (Gedimino 50, LT-44239 Kaunas, Lithuania) and Vilnius Gediminas Technical University (Saulėtekio 14, LT-10223 Vilnius, Lithuania).

KAUNO TECHNOLOGIJOS UNIVERSITETAS

VOLDEMARAS ŽITKUS

KOREFERENCIJŲ SPRENDIMAI LIETUVIŲ KALBAI

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

Kaunas, 2024

Disertacija rengta 2013–2019 ir 2023–2024 metais Kauno technologijos universiteto Informatikos fakultete, Informacijos sistemų katedroje.
Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu.

Disertacija ginama eksternu

Mokslinis konsultantas:
Prof. Dr. Rita BUTKIENĖ (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007).

Redagavo: anglų kalbos redaktorė Dovilė Blaudžiūnienė (leidykla „Technologija“), lietuvių kalbos redaktorė Aurelija Gražina Rukšaitė (leidykla „Technologija“)

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:
prof. dr. Tomas SKERSYS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007) – **pirmininkas**;
prof. dr. Nikolaj GORANIN (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Diana KALIBATIENĖ (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Tomas KRILAVIČIUS (Vytauto Didžiojo universitetas, gamtos mokslai, informatika, N 009);
dr. Zbigniew MARSZAŁEK (Silezijos technologijos universitetas, Lenkija, gamtos mokslai, informatika, N 009).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2024 m. birželio 17 d. 10.00 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, LT-44249 Kaunas, Lietuva
Tel. (+370) 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2024 m. gegužės 17 d.
Su disertacija galima susipažinti interneto svetainėje http://ktu.edu, Kauno technologijos universiteto bibliotekoje (Gedimino g. 50, Kaunas, LT-44239, Lietuva) ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva).

TABLE OF CONTENTS

TERMS AND ABBREVIATIONS

AI (**A**rtificial **I**ntelligence) – Intelligence demonstrated by machines that could be compared to human intelligence.
Adverbial coreference – Type of coreference where referent is expressed by an adverb.
Anaphora – An expression, the interpretation of which depends on another word or phrase presented earlier in the text. A subtype of coreference.
Annotation – Metadata about certain data, in this case text.
Antecedent – Mention to which a referent refers to. Technically it covers only those situations where a mention is present in the text before the referent, but in literature distinction is usually not made.
Candidate antecedent – Possible antecedent for a referent.
Cataphora – An expression, the interpretation of which depends on another word or phrase presented later in the text. A subtype of coreference.
Coreference – When two, or more, expressions refer to same discourse-world entity, it can be said that they corefer. The relationship between these expressions is coreference.
CR (**C**oreference **R**esolution) – Proccess of resolving coreference expressions.
Corpus – A collection of writings or recorded remarks used for linguistic analysis.
Definitive nominal coreference – Type of coreference where a referent is expressed by a definitive noun.
Deixis – A type of Exophora usually expressed by first or second person pronouns in direct speech.
DRT (**D**iscource **R**epresentation **T**heory) – A formal framework for exploring meaning behind discourse.
Ellipsis coreference – A type of coreference where a referent is expressed by a gap.
Endophora – Text reference that refers to something present in the same text.
Exophora – Text reference that refers to something present outside of the text.
FOL (**F**irst-**O**rder **L**ogic) – Also known as predicate logic, a collection of formal systems where each statement or sentence is broken down into a subject and a predicate.
Generic nominal coreference – A type of coreference where a referent is expressed by a generic noun.
IR (**I**nformation **R**etrieval) – The process of retrieving relevant information that a user requested through query or other means.
IE (**I**nformation **E**xtraction) – The process of extracting information from various data sources.
JSON (**J**ava**S**cript **O**bject **N**otation) – Human readable data format for data storage and transmition.
LCC (**L**ithuanian **C**oreference **C**orpus) – A corpus created for Lithuanian language that focuses on coreference expressions.
Lemma – Canonical, or dictionary, form of a word.
Mention – A text fragment representing certain discource-world entity.
NE (**N**amed **E**ntity) – Text fragments, usually consisting of definitive nouns, which

refer to some discourse-world entity.
NER (**N**amed-**E**ntity **R**ecognition) – The process of identifying named entities in the text.
NLP (Natural **L**anguage **P**rocessing) – A branch of computer science, information engineering and artificial intelligence that deals with analyzing and understanding the natural human languages.
NP (**N**oun **P**hrase) – A phrase that has a noun as its head. Also known as nominal.
MUC (Message **U**nderstanding **C**onference**)** – Conferences that focused on information extraction from text.
Ontology – a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities, their relations, individuals, and constraints.
OWL (**W**eb **O**ntology **L**anguage) – A semantic markup language for publishing and sharing ontologies on the World Wide Web.
POS (**P**art-**o**f-**S**peech) **tagging** – Process of determining (and assigning) part-of-speech to words present in the text.
Postecedent – Mention which is present later in the text than the referent that refers to it.
Pronominal coreference – A type of coreference where referent is expressed by a pronoun.
RDF (**R**esource **D**escription **F**ramework) – Resource Description Framework is a standard model for data interchange on the World Wide Web.
Referent – Text fragment that refers to another text fragment present in the same text.
Salience – Prominence of certain phrase in the text.
SRL (**S**emantic **R**ole **L**abelling) – A technique used for deriving a structured semantic meaning behind sentences.
Semantic search – Type of search when results are retrieved due to underlying meaning behind the text.
Semantic Web – A "web of data" that enables machines to understand the semantics or meaning of information on the World Wide Web.
SSFLL (**S**emantic **S**earch **F**ramework for **L**ithuanian **L**anguage) – NLP and semantic search framework that targets Lithuanian language.
SPARQL – An RDF query language of the Semantic Web.
Stand-off annotation – Annotation that resides in different location than data for which this annotation was created.
Tokenization – The process of demarcating and classifying sections of a text.
VP (**V**erb **P**hrase) – A syntactic unit that has at least one verb and its dependents.
Web crawler – Internet bot that systematically browses the World Wide Web for various purposes like indexing or retrieving certain imformation.

FIGURES

TABLES

INTRODUCTION

With the emerging growth of Semantic Web technology, the way Web information retrieval (IR) has been seen is changing towards meaning-based IR, which will be referred to as semantic search. The quality of retrieved documents relevant to user information needs highly depends not only on IR methods applied but on Information Extraction (IE) methods used as well. In general, IE is known as an activity of automatically extracting structured information from the unstructured information source. Standard document text pre-processing steps used in classical IE models are lexical analysis, morphological analysis, named-entity recognition (NER). Some IE solutions are getting complimented by more advanced IE methods such as coreference resolution (CR), semantic annotation, and ontology population. The main challenge here is the complexity and ambiguity of natural language, hence making IE dependent on advances in Natural Language Processing (NLP) techniques. While state-of-the-art in IE-related NLP research for well-known languages (e.g., English) has already reached levels of successful practical application on a massive scale (e.g., IBM's Watson project) [1], less popular and resource-scarce languages such as Lithuanian, remain an open NLP research field.

In NLP context, coreference occurs when two different linguistic structures refer to the same entity. Resolving a relationship between these structures is an important part of NLP and can greatly improve semantic search, automatic translation, question answering systems, and various similar solutions [2].

For example, "*Tom* skipped school today. *He* was sick." 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 skipped school nor who was sick. In such cases, semantic information would be lost.

The purpose of this work is to create methods and required resources for CR in the Lithuanian language. At the time, to our knowledge, there are no suitable techniques proposed for coreference resolution in the Lithuanian language.

Motivation

An advance in the development of NLP tools for the Lithuanian language in 2014 allowed us to create a Semantic Search Framework[1] for Lithuanian Language (SSFLL) Internet corpus extracted from public news portals [3]. This framework is oriented towards answering questions presented in Structured Lithuanian (based on Semantics of Business Vocabulary and Business Rules [4] [5] [6] [7] [8]) language. The framework transforms these questions into SPARQL queries and executes them in ontology populated by individuals discovered by semantic annotation tool. The quality of query results (precision and recall of answer) highly depends on the quality of NLP pipeline used for IE and the quality of the components themselves.

The key component of semantic search is a semantic annotator that extracts

[1] „Syntactic and Semantic Analysis and Search System for Lithuanian Internet, Corpus and Public Sector Applications in Lithuanian Language" (No. VP2-3.1-IVPK-12-K-01-007) Project financed by EU Structural Funds. Partners: Vytautas Magnus University (coordinator), Kaunas University of Technology. (2012-2015)",

semantic information from a text which is stored in a database and can be later queried against. But the problem arises when the same discourse-world entity is referenced in the text by different linguistic structures like pronouns, synonyms, features of the entity, and the like. This, in turn, can cause two problems:

1) Some semantic information might be lost entirely if the discourse-world entity is referred to by pronoun or other ambiguous linguistic structure.
2) Even if the discourse-world entity is not referred to by ambiguous linguistic structure it can still be difficult to determine if that structure refers to the same entity as an earlier structure or they both refer to different discourse-world entities.

Due to these problems, the quality of semantic annotator and, in turn semantic search, can decrease. Therefore, it was decided that a CR component is required to solve these problems. It can significantly enrich the ontology population by identifying additional occurrences and links of entities, already identified after linguistic processing, and using various existing knowledge bases.

Unfortunately, the Lithuanian language does not have many linguistic resources and is, in general, under-researched when it comes to the NLP field. This makes it difficult to adapt to the Lithuanian language state-of-the-art CR approaches that have been developed for English and other well-researched languages. Therefore, it was decided that a new CR approach focusing on the Lithuanian language, which takes in mind available linguistic resources, has to be developed.

Object and scope of research

The object of the research is the CR process for Lithuanian language.

The scope of the research encompasses CR approaches, their evaluation, coreference corpora and related NLP resources focusing on information extraction rather than linguistic analysis.

Problem statement and research questions

While work on various NLP parts for the Lithuanian language has been done or started, CR remains an unexplored venue therefore the overall quality of semantic search is lower than it could be.

Research questions:

1) Can CR approach be developed with limited linguistic resources (lexical, morphological and NER annotations) that would provide useful results for higher-level application needs?
2) What impact can CR have on the results of semantic search?
3) Is it worth it to invest resources into developing rule-based CR approaches when compared to the solutions based on the machine learning?
4) To what degree is it possible to adapt CR approaches from one language to another that is linguistic resource-scarce?

Goals and tasks

The goal of the research is to improve the capabilities of CR in the Lithuanian language by developing methods and required resources for it. To reach the goal, the

following tasks were stated:

1. Analyse current methods and resources used for CR in English and other languages;
2. Develop models, resources, and algorithms for CR in Lithuanian language;
3. Implement developed models and algorithms that would be used for annotating Lithuanian text corpora;
4. Conduct an experiment that would evaluate the suitability of implemented models and algorithms.

Tasks were further specified in section 1.4 after analysing the related literature was completed.

Research methodology

This research is based on the Information System Research Framework adapted by Hevner et al. (2004) [9]. In the first stages of the research problem definition a potential loss of semantic information due to lack of CR approach suitable for the Lithuanian language was established. A comparative analysis of the existing CR approaches and their adaptability to Lithuanian language was then performed. Due to the complexity of the CR task related subjects, such as coreference corpora, were also analysed.

Focus of the research was a new CR approach suitable for the Lithuanian language, but due to the mentioned complexity of the CR task it was decided that the solution has to also encompass a coreference corpus and the process of evaluating the result of the CR approach. As a result, three artefacts – a CR approach, a coreference corpus and the evaluation model, were created.

Artefacts created during the implementation and evaluation stages were evaluated in the context of SSFLL project and later integrated into it. Results of this work have been published in peer-reviewed publications.

Defended statements

1) Rule-based coreference resolution that uses only lexical, morphological, and named entity annotations can resolve a subset of coreference relationships and achieve reliable precision.
2) Quality of semantic search depends on the capability of the semantic annotator to identify objects and facts related to them. Coreference resolution results allow us to aggregate dispersed semantic information. Due to that, coreference annotation can significantly enrich semantic annotations and in turn improve the results of semantic search.
3) Evaluation strategy of coreference resolution approaches can provide more detailed and valuable information if it uses linguistic information present in the coreference relationships.

Major contributions and novelty

Major contributions of this work:

- The Lithuanian Coreference Corpus (LCC) was created. It targets

specifically the Lithuanian language.

- A coreference resolution method for the Lithuanian language was created.
- An annotation scheme with which the coreference corpus was annotated was created. The proposed CR approach uses the same annotation scheme.
- An evaluation model for evaluating CR approaches was developed.

The novelty of this work:

- To our knowledge, the LCC corpus and coreferences resolution method presented in this work are the first such resources targeting the Lithuanian language.
- The created annotation scheme is very flexible, leaving most of the implementation questions open so that it could be easily adapted for other languages and integrated into pre-existing (or newly developed) solutions. Even if the classification of coreference expressions would have to be changed due to differences in languages or the focus of the research, the main principles of the proposed scheme would still be useful and relevant.
- The developed evaluation model is based on the dominant mentions instead of enforced transitivity. It gives equal weight to each coreference type regardless of their number in the text and allows to differentiate between errors based on their severity.
- CR method uses a small number of linguistic resources, which makes this approach useful for other under-researched languages when it comes to NLP resources.
- Rule-based CR approaches usually are not properly formalized, which often makes it difficult to adapt these rules for other languages or contexts. The created method is rule-based, but all developed and tested rules have been formalized using first-order logic. This makes it easier to port to other, grammatically similar languages.

Practical significance

The presented solution allows to solve coreferences in the Lithuanian language, hence the results of semantic annotators can be improved.

The results of this work have been integrated into SSFLL and have been used to annotate Lithuanian Internet corpora for Politics, Business and Economy, and Public Administration domains.

Scientific approval

The results of this work have been presented in three international and one Lithuanian conference. Two articles have been published in scientific journals. Four articles were published in other scientific publications – proceedings of the conference. The detailed list of publications is presented in section eight.

Thesis structure

The first section is divided into three parts. Firstly coreferences, their types and required resources for their resolution are overviewed. This section also establishes the boundaries of this research and clarifies which types of coreference expressions

are not covered in this work. In the second part, coreference corpora and their annotation schemes are analysed and compared against each other. Lastly, existing CR approaches for other languages are analysed and are compared against each other as well.

In the second section, the coreference corpus created for the Lithuanian language and the annotation scheme for it are presented. Their implementation is also overviewed there. The third section contains the proposed CR approach and its implementation. Finally, the fourth section is dedicated to the experimental evaluations of the proposed solution and coreference corpus.

The fifth section presents the conclusions of the research. In the sixth section, a summary in the Lithuanian language is provided, followed by a list of references and the author's scientific publications.

1 ANALYSIS OF THE STATE-OF-THE-ART IN THE COREFERENCE RESOLUTION FIELD

Coreference resolution is a complex problem that is important to various NLP and linguistic tasks. Therefore, the scope of this research is rather wide and scientific literature from various fields is relevant. In Section 1.1, coreference expressions are analysed: their types, dependencies, formalization, and NLP resources required for their resolution. In section 1.2, literature relating to coreference corpora, including their annotation schemes and evaluation strategies, is analysed. In Section 1.3, the existing coreference resolution approaches in English and Balto-Slavic languages are covered. In Section 1.4, research tasks have been specified and Section 1.5 provides the summary of the analysis.

1.1 Coreferences

Coreference resolution (CR) is the process of linking entities to expressions that refer to them [10]. Let us move back to the previously mentioned example:

- Tom skipped school today. He was sick.

Usually, such expressions as above are called anaphoric expressions which can be considered a subtype of coreferences, but the terminology used in the literature varies [11]. Therefore, a definition of what is consider a coreference in the context of this research is provided:

- Anaphora is an expression, the interpretation of which depends on another word or phrase presented earlier in the text (antecedent [10] [12]). In the aforementioned example, the word "Tom" would be considered an antecedent in such a case. Usually, 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.
- Cataphora is identical to anaphora with the only difference being that it refers to a phrase that will be present later in the text (postcedent [10] [12]).
- Every other type of reference that can be independently interpreted (for example, if instead of "He" there would be "The boy") is considered a general case of coreference.

Coreferences can also be divided into two groups:

- Endophoric references – referring to something present in the same text [13].
- Exophoric references – referring to something outside of the text and usually requiring some additional information (like the context in which text was written or author's other works) to make a correct interpretation. Deixis, or deictic expression, is one of such references. To interpret the phrase, we need to know, for example, who is speaking or writing the text [13].

In this research, exophoric references are not addressed and as such deixis phenomenon is ignored unless it overlaps with anaphoric expressions in certain situations.

In this work, referring expression (in this case “he”) will be called **referent** and expression to which it is pointing (in this case “Tom”) will be called **antecedent**. When talking about discourse-world entity in general, the term **mention** will be used.

Two or more referents can refer to the same antecedent, for example:

- *Tom* didn’t want to stay any longer because *he* was tired. So, *the man* went home to sleep.

Both “he” and “the man” are referring to “Tom”. Such expressions are said to corefer to the same entity. Often there are multiple possible solutions:

- *Tom* visited *Jane* yesterday. *She* looked very tired.

The referent “she” might be referring to either “Tom” or “Jane”. In such a case, both of them would be called **candidate antecedents**. After performing gender agreement between referent and candidate antecedents, “Tom” would be removed, and “Jan” would be considered correct antecedent.

The usage of coreference expressions can vary depending on the type and style of the text. For example, technical manuals tend not to have many such expressions and avoid complex constructions in general, while literary works often employ them for stylistic and other purposes. This work focuses on texts from news sites that cover political and economic domains.

1.1.1 Linguistic dependencies

Coreferences, like any other linguistic expression, have various dependencies with other words present in the text. Such dependencies can be classified into four major types [14]:

- Semantic, focusing on predicates and their arguments.
- Morphological, focusing on the dependencies between the words or parts of the words.
- Syntactic, focusing on the sentence structure.
- Phonological, focusing on separate sounds in languages.

Phonological dependencies will not be further detailed since the scope of this research is limited to written text and as such phonological sounds are not relevant.

It is important to note that different types of dependencies can overlap or even contradict each other therefore they should be treated as separate layers of the text fragment and not as separate parts of the same layer.

Semantic dependency

In sentences when one word depends on another for its meaning, they are called predicates and arguments. For example:

(a) I borrowed [W1] father’s [W3] car [W2].

(b) I suggested [w1] him [w3] to run [w2].

In sentence (a), two dependencies, W1→W2, and W2→W3 can be seen. In the first case, the word “borrowed” is predicate and “car” is its argument, in the second case “car” is predicate and “father’s” is its argument.

Semantic dependency has the following properties:

1. It is anti-symmetrical. One word cannot be an argument for the meaning of another word and then have that word as its argument.

2. It is anti-reflexive. Word cannot have itself as its argument.
3. It is neither transitive nor anti-transitive. As seen in example (a), it cannot be said that W1→W3 is true despite W1→W2 and W2→W3 being valid dependencies. But as seen in example (b), W1→W3 would be valid.
4. One word can have multiple dependencies as seen in example (b) with W2→W3 and W1→W3 being valid dependencies.
5. Semantic dependency encompasses all words in the sentence. This means that a fully connected semantic tree can be constructed for every sentence.

Morphological dependency

Generally speaking, morphological dependency happens when one word influences the morphological form of another word. For example:

(c) I [W1] am [W2] well.

(d) You [W1] are [W2] well.

Here the morphological form of the word "be" depends on the pronoun used, W1→W2. Types of influence can be further divided into various inflectional categories such as tense, gender, number, case, etc.

While semantic dependency and its consistent properties can be found in all languages, the same cannot be said about morphological dependencies. Since different languages do not share all properties of the morphological dependencies, only those applicable to the Lithuanian language will be detailed. A few examples in the Lithuanian language with English translations:

(e) Vaikas turėjo dvi [W1] monetas [W2].
The kid had two [W1] coins [W2].

(f) Aš pažinojau [W1] juos [W2] jaunus [W3].
I knew [W1] them [W2] young [W3].

(g) Aš mačiau [W1] raudoną [W3] mašiną [W2].
I saw [W1] red [W3] car [W2].

Lithuanian has the following properties of morphological dependency:

1. It is anti-symmetrical in one inflectional category. As seen in example (e), the number of noun "coins" (*"monetas"*) is influenced by the numeral "two" ("*dvi*"), but the reverse is not true.
2. It can be symmetrical between different inflectional categories. While in the previous example the numeral determined the number of the noun, the same noun determines the gender of the numeral.
3. It is anti-reflexive. A word cannot influence its morphological form.
4. Similarly to semantic dependency, it is neither transitive nor anti-transitive as can be seen in examples (f) and (g). In the first case, we have W1→W2, W2→W3, and W1→W3. In the second case, we have W1→W2 and W2→W3, but not W1→W3.
5. The agreement is a separate type of morphological dependency. In such a case, neither of the words depends on another but they agree in a certain inflectional category. For example, two words might not have any dependencies between each other but have the same gender. In such a case, we would say that these two words agree in gender.

6. One word can have only one dependency in one inflectional category but might have multiple dependencies in different categories.
7. Morphological dependency does not encompass all words in a sentence. Due to this, fully connected morphological trees cannot be constructed.

Syntactic dependency

This type of dependency does not focus on the meaning or form of the words in the sentence. Strictly speaking, syntactic dependencies allow building syntactic structure which depicts structure of the sentence. Usually, syntactic structure is represented using dependency trees.

It can be said that one word (W1) has a syntactic dependency with another (W2) if the following criteria are met:

1. W1 and W2 are linked in linear order. For example, noun following preposition.
2. W1 and W2 form a phrase, or they form a phrase with an additional set of words.
3. One of the words has to govern (dominate) another. This can be determined by valency, a number of arguments controlled by a predicate. The word with higher valency is considered a governing word in such a case. If valency cannot determine the governing word, then morphological dependency is used, i.e., the word that influences the morphological form of another word is considered the governing word in the syntactic dependency tree. If this also does not establish the governing word, then the governing word of semantic dependency is selected.

An example of a syntactic dependency tree for “My cousin saw a large ship” can be seen below in Figure 1.1.

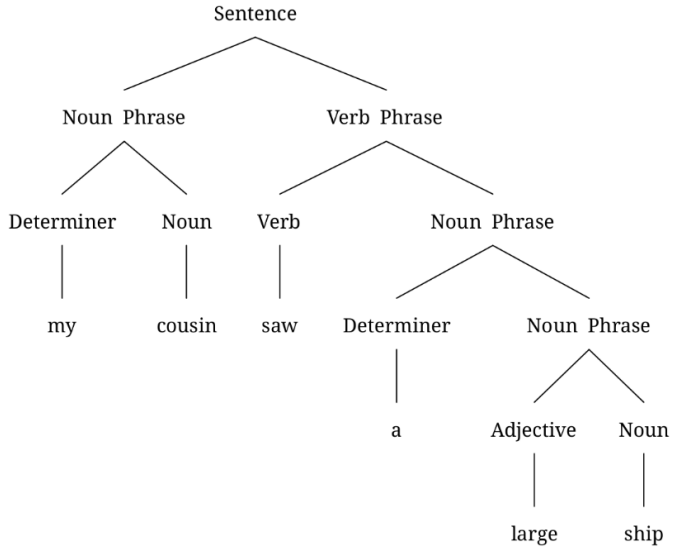


Figure 1.1 A sentence parsed into a syntactic tree

Syntactic dependencies have the following properties:

1. It is anti-symmetrical. One word cannot govern another word and, in turn, be governed by it.
2. It is anti-reflexive. A word cannot govern itself; this also follows from the first property since anti-symmetrical objects cannot be reflexive.
3. One word can be governed only by one other word.
4. It is anti-transitive. Otherwise, the uniquity of the governing word would be violated. Though it can be said that there is indirect dependency W1→W3 if there are W1→W2 and W2→W3 dependencies.
5. Like semantic dependency, syntactic dependency also encompasses all words in the sentence. This allows the construction of connected dependency trees.

1.1.2 Coreference types

There are many different types and subtypes of coreferences. Primarily all of them indicate that two or more text fragments refer to the same concept, but deeper classification is necessary since it can provide additional semantic information. In this section, coreference types are explained, examples are provided in the English language unless something specific to Lithuanian language needs to be highlighted. Not every coreference type is covered here since some of them are not relevant to this research due to differences between languages.

Pronominal coreferences

Coreference expressions when the referent is expressed by a pronoun are usually called pronominal, or pro-form, coreferences [15]. There are many different subtypes of pronominal coreference, usually depending on the type of pronoun used.

Personal pronouns like *he, she, they* are very commonly used for constructing coreferences. In CR, first- and second-person pronouns are often excluded due to being deictic [16], but this is not always the case. For example, if someone has his speech cited:

- "*I* forgot about your book, sorry" – apologised *Tom*.

In such a case it can be determined that pronoun "I" refers to "Tom" and therefore is not deictic. But the resolution of such cases should be carried out carefully since first- and second-person pronouns are not gender specific and can easily lead to false positives.

Reflexive (himself, herself) and possessive (his, hers) pronouns are fairly straightforward. They always refer to some discourse-world entity in the text.

The usage of relative pronouns (which, who) is usually syntactically defined [17] and follows rather strict rules even in free word order language like Lithuanian:

- *Žmonės*, *kurie* rūko, išleidžia daug pinigų tam.

Pronoun "kurie" refers to noun "Žmonės". If a strong syntax tagger is available, then most of the coreferences that use relative pronouns can be automatically solved by using syntax parse tree. Unfortunately, at the time, a suitable syntax tagger for the Lithuanian language was not available, hence relative pronouns are covered in this research.

Demonstrative pronouns (this, that) are also often used in coreference expressions:

- He *failed the exam*. *That* was unexpected.

In this case "That" refers to "failed the exam". But, similarly to first- and second-person personal pronouns, demonstrative pronouns can often be deictic and refer to something that is not present in the text. Moreover, in the Lithuanian language, demonstrative pronouns are often pleonastic.

Pleonastic pronouns are pronouns that are not referential and due to that do not refer to any discourse-world entity present in the text or outside of it [18]. If not identified and removed from the list of possible referents, they are bound to produce false positives during CR.

Due to these problems, it was decided to leave out coreference with demonstrative pronouns out of this research at the time. But at the same time, demonstrative pronouns in the Lithuanian language are also often used similarly to how definitive article "the" is used in the English language: *šis* namas; *tas* pastatas; *anas* vaikas. They allow identifying that the following word is referring to some specific entity and not mentioning it in general. Usually, generic nouns are not referential.

Pronouns like "each other" and "one another" are called reciprocal pronouns [19]. For example:

- *Tom* and *Jane* know *each other*.

In this case, "each other" refers to Tom and Jane. But it is questionable if this is really a coreference or is "know each other" phrase a statement of fact and therefore should be resolved by semantic annotator and not by CR. Due to this ambiguity, such expressions are not covered in this research.

Nominal coreferences

When a referent is a noun or noun phrase (NP), then such coreference is called nominal. Most common cases of this are:

- When the full (or partial) name is repeated multiple times in the text.
- When one noun is replaced by another noun that is related by some linguistic relationship (synonym, metonym, hypernym).
- When the discourse-world entity is referenced by its certain feature. For example, at first, the full name of the person might be stated, but later they might be referenced by their profession, age, political leanings, or some other feature.

Nouns are also used in a couple of corner cases like associatives and appositional coreferences.

Associative coreferences, sometimes called bridging anaphora, are expressions when two objects are related to each other in some explicitly not stated way. For example:

- *The house* is great, but *the kitchen* needs work.

In this case "The house" and "the kitchen" form associative expression. These are two different objects, but they are related to each other – the particular kitchen is in the previously mentioned house. Outside of the coreference context, such

relationships are called meronymy. There are some disagreements if such expressions should be tackled by CR tasks or not [20] [21] [22]. We do not consider such expressions as a part of CR task, and they will not be covered in this research.

Appositional coreferences occur when apposition establishes an alternate name (or feature) that can be used to identify the previously mentioned entity. For example:

- Tom, the third-year student, was late for the lecture. The lecturer was not kind to the student for missing the first 10 minutes.

In this case "third-year student" is apposition. It establishes that Tom is a student so when the student is mentioned in the second sentence, we can understand that the lecturer was not kind to Tom and not some other student in the class. But as with associative expressions, it is questionable if apposition itself is coreferential with the discourse-world entity and, if so, how should it be marked [23] [24]. While extracting information from appositions is undoubtedly an important task, we considering it as being a statement of fact rather than coreference expressions and due to that, it will not be further covered in this research.

A similar situation is with predicative [24] [25] expressions:

- Tom is a student.

This is useful information for CR as well, but we do not consider that in such case "student" refers to "Tom", but that it is a statement of a fact.

Adverbial coreferences

Adverbs can be used as well to refer to certain reason, location, or time [10]. Adverbs differ in the Lithuanian language, so an example is provided in English and in Lithuanian languages:

- EN: Yesterday it *snowed all day*. *Due to that* some roads were closed.
  LT: Vakar *visą dieną snigo*. *Dėl to* buvo uždaryti kai kurie keliai.

In this case "Dėl to" is an adverb and it refers back to "visą dieną snigo". It is a reason adverb and explains why certain roads were closed.

Ellipsis

Often called zero anaphora [26], this term describes the use of a gap in a phrase or clause that refers back to the previously mentioned phrase. For example:

- Tom saw the burglar. Identified him as Jim from the school.

In this case gap before "Identified" refers back to "Tom" and the sentence could be rewritten as "Tom identified him as Jim from the school".

Similar to zero anaphora is a kind-level expression [27] or one-anaphora. The difference is that in this case it is established that a different entity of the same type (as previously introduced) is referred to:

- John gave a *presentation*. Sarah gave *one* too.

Both John and Sarah gave their presentations, but they were different presentations. But the Lithuanian language does not use "one" construction in such cases:

- Jonas pristatė *prezentaciją*. Sara irgi pristatė.

Gap after "pristatė", or before since Lithuanian language has free word order, refers back to "prezentaciją". Due to this zero anaphora and one-anaphora are very

similar in the Lithuanian language.

Verb phrase (VP) ellipsis is another similar case with VP, instead of NP, being referred back to by a gap. Rewriting the same example from above:

- Jonas *pristatė prezentaciją*. Sara irgi.

Gap after, or before, "irgi" refers back to VP "pristatė prezentaciją".

Presuppositions

While writing and reading a text, many presuppositions are made, for example:

- Joe broke his leg in July 1933, and Jack *also* broke his leg at the age of 15.

In this case due to the word "also" we make an assumption that when Joe broke his leg in 1933, he was 15 years old, like Jack [26]. The argument is made that since coreference and presuppositions have similar configurational triggers then they should receive uniform treatment [27] [28]. But even if they have similar triggers, it does not mean that CR should also cover presupposition resolution. Therefore, the presupposition problem will not be addressed in this research.

1.1.3 Formalization in coreference resolution

Formalization of the methods and algorithms is an important step as it allows to validate them and makes them easier to adapt to other environments. In CR, context formalisation can be of two kinds:

- Linguistic focused, where coreference expressions themselves are formalized and their links detailed.
- Information extraction focused, where algorithms and rules themselves that solve coreference expressions are formalized.

There are multiple different formalisms that have been used for one, or both, of these goals.

Predicate logic

While predicate logic, also known as First-order logic (FOL), is popular formalism in other domains, it is not particularly popular in CR. One of the reasons for that is problems with linguistically expressing certain sentences. In order to solve this problem, multiple modifications of predicate logic have been suggested [29]. But neither of these modifications have been widely adopted.

On the other hand, predicate logic is very suitable in formalising algorithms and rules. Formally defined algorithms are easier to adapt and implement. Unfortunately, in the CR context, it is rarely used for this purpose.

Discourse Representation Theory

Alternative to predicate logic for linguistic formalisation is Discourse Representation Theory (DRT) [30]. Unlike predicate logic, it is specifically aimed at interpretation of the discourse. It also has better readability than predicate or default logics, as seen in this example:

- If Pedro owns a donkey, he beats it.
- [x: Pedro(x), [y: donkey(y), owns(x,y)] ⇒ [v, w: beats(v,w)]]
- [x, v: v = x, Pedro(x), [y, w: w = y, donkey(y), owns(x,y)] ⇒ [: beats(v,w)]]

- [x: Pedro(x), [y: donkey(y), owns(x,y)] ⇒ [: beats(x,y)]]

Since it specifically targets interpretation of the discourse, it cannot be used for formalization of rules and algorithms.

Pseudocode

Pseudocode is technically not formal, but it is often used for the same purpose, to describe computer programs and algorithms. It is more readable than formal alternatives and is used more often in the CR context. But since it is not formal language and does not have any standards for its syntax, it cannot be validated and is not as easy to adapt to other environments as formal rules and algorithms. It also is not suitable for linguistic descriptions due to primarily using programming language conventions.

Since the focus of this work is on information extraction and not on linguistic research it was decided to only use predicate logic for rule and algorithm formalisation.

1.1.4 NLP pipeline

CR requires various linguistic annotations and knowledge bases (ontologies and vocabularies). Due to this reason, it is hard to imagine CR working independently outside of other natural language processing tools and processes. The abstract model of such a process, the NLP pipeline, is presented in Figure 1.2. It covers the most frequently used NLP components whose results are later used by CR.

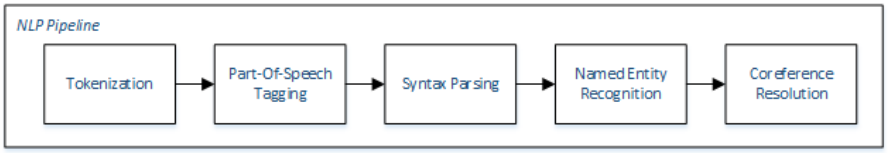


Figure 1.2 Abstract coreference resolution NLP pipeline

This NLP pipeline does not cover every possible pipeline but provides a general idea of what is usually required for CR based on some popular and widely adopted NLP frameworks [31] [32] [33] and related literature [34]. Below, each NLP component is briefly overviewed and its relevance to CR is explained.

Tokenization

Tokenization is a process of breaking up natural language text into distinct and meaningful tokens [35]. This is a vital NLP task that must be done before any other NLP tasks can be performed. For example:

- Tom missed the school bus.

This sentence has 6 tokens: Tom; missed; the; school; bus; period (.). At first, this might look like a trivial task, but the difficulty of this task varies by language depending on how well that language is punctuated. But even if the text is well punctuated there can be ambiguities. For example, does the period at the end of the word signify the end of the sentence or that previous word is an abbreviation? There is also an obvious question of which text fragments are and are not meaningful. The

answer to that question might differ in different types of research.

Sentence splitting is sometimes considered as part of the tokenization process since sentences can be argued to be meaningful tokens as well. In other cases, sentence splitting might be done by another component in the NLP pipeline.

In CR context tokens are also useful for calculating the distance between referent and candidate antecedent. Most solutions give priority to antecedents that are closer to the referent. There can also be a set limit for how far an algorithm should search since at some point the algorithm can start encountering false positives.

Part-Of-Speech (POS) Tagging

The process of assigning a POS tag to each token is called POS Tagging [36]. A list of possible tags varies by language, but the most common ones are noun, verb, adverb, pronoun, punctuation, etc.

Tokens often can have multiple viable interpretations. For example, “kick” can be both a verb and a noun depending on the context that it is being used in. Due to that POS taggers usually, also provide a disambiguation feature that selects the most likely interpretation.

This task is very important for CR because most of the time referents are either nouns or pronouns while antecedents usually are nouns.

A POS tagger might be extended to also provide the lemma (a canonical form of the word) and additional morphological information. In such a case, it is then usually called Morphological annotator. Depending on the language, if it is morphologically rich or not, it might provide such information as gender, number, case, and other relevant information.

Syntax Parsing

The main goal of syntax parsing is to identify the structure of the sentence [37]. Syntax structure is hierarchical therefore it is usually displayed in a dependency tree. An example of such a tree was provided in Section 1.1.1, Figure 1.1.

Syntax analysis helps to determine subjects, objects, and predicate-argument dependencies in sentences. Such information is useful for CR, few common usages of syntax parsing output:

- It allows establishing syntactic parallelism between a possible referent and antecedent pair.
- Often the object of the sentence becomes the subject of the following sentence.
- With the syntactic hierarchy established, certain candidate antecedents can be removed from consideration due to their place in said hierarchy.

Named-Entity Recognition

Named entities (NE) are a specific class of information units, like names of persons, organizations or locations, numeric expressions defining time, date, and money [38]. The process that identifies such information in natural language is called Named-Entity Recognition (NER). For example:

- In 2010 Tim bought Apple shares.

In this sentence we have three named entities: “2010” is a date entity, “Tim” is a person entity, and “Apple” is an organization entity.

Recognition of these entities is very important for CR for two reasons:

- Named entities are one of the most common expressions used for coreference, therefore their identification is important for having a full list of possible candidates.
- Being able to tell what type each entity is, allows constructing selection constraints when there is more than one possible solution. For example, a specific location cannot be moved from one country to another while an organization can relocate to another country. In such a case, if it can be classified that a specific entity is a location, it then can be removed from the list of candidate antecedents and leave only those entities, which can be moved (or move by themselves) from one country to another.

1.2 Coreference corpora analysis

In order to solve CR problems, a corpus has to be created simultaneously [10], which could be used for developing solutions, testing and evaluating them. In fact, the creation of pre-annotated corpus can be considered as a part of CR task. As such, there have been many attempts to create annotation schemes and the corpus that could be used for these tasks. Annotation schemes are not language independent [39], and some older schemes have noticeable flaws that have been criticized [40].

This section is divided into three parts. The first part covers the existing annotation schemes and corpora for other languages, while the second part analyses the process of creating these resources. The final section discusses the evaluation strategies of coreference annotation.

1.2.1 Analysis of existing resources

This section overviews some popular schemes and corpora made for English and other languages. Solutions done for Slavic languages are more relevant for our research due to having similarities to the Lithuanian language, being morphologically rich and having a free word-order.

Message Understanding Conferences (MUCs) [41] were one of the earliest coreference evaluation attempts done for the English language. It has not been updated since MUC-7 was released in 1998 but is still the most known scheme and closest to standard despite limited coverage and questionable coreference definition [40] [42] [43]. It is also often used as a reference point for new annotation schemes and corpora.

From newer approaches, the closest to MUC in terms of the spread of use is MATE/GNOME [42]. When compared to MUC, it has expanded coverage and clearer coreference definition. At the same time, the authors state that it is impossible to create a general-purpose coreference annotation scheme. In an attempt to solve this problem, MATE/GNOME is divided into two parts: MATE being the general-purpose mark-up scheme while GNOME is a specific implementation for required domain or language. Few additional schemes have been built upon the MATE/GNOME approach: AnCora-CO [44] for Spanish and Catalan languages, PoCoS [43] for the English, German, and Russian languages [45] [46]. The approach proposed in this paper is

fairly similar but the attempt is made to abstract the MATE equivalent even further while leaving the GNOME equivalent entirely up to specific implementation without any demanding guidelines.

One of the biggest and most developed corpora is Czech The Prague Dependency Treebank [47] [48] [49]. Unfortunately, it is based on tectogrammatical annotations and requires syntactic analysis. Therefore, their solution is difficult to adapt to other languages or corpora, especially for Lithuanian language which lacks many of the resources (and knowledge) required to adopt this approach.

Another big Slavic language resource is Polish Coreference Corpus [50] [51] with close to 2000 documents covering 14 different genres. Their annotation scheme is unique with having a quasi-identity relation, semantic-head, and dominant expression mark-ups, and grouping all mentions into nominal groups (NGs). In this research, a modified version of dominant expressions is used, but a detailed classification of mentions is provided instead of grouping them into nominal groups. Their analysis of the Polish language [51] was also useful due to similarities to the Lithuanian language.

The Automatic Content Extraction (ACE) program [52] is a comparable approach to MUC, but it expands into covering coreferences that are presented not necessarily in written text, but also by speech recognition or optic character recognition input. Additionally, it restricts coreferences relation between seven specific entity types. Due to exophoric expression being outside of this research's scope, not much was taken from this approach.

Another similar approach is OntoNotes [53], which is developing three large corpora for the English, Chinese, and Arabic languages. While being fairly straightforward, it is questioned if it can fully capture coreference expressions [44].

Despite the English language having many large corpora for coreference data, new corpora for the English language are still being created. They often focus either on a specific domain [54], a data source [55], or specific problems [56].

A summary of the analysed corpora is provided in Table 1.1.

Table 1.1 Summary of the analysed corpora

Corpus	Focus	Language
The Prague Dependency Treebank [47]	Tectogrammatical annotations	Czech
Polish Coreference Corpus [50]	General nominal coreference	Polish
The Automatic Content Extraction (ACE) [52]	Within-document Event Coreference	English, Chinese, and Arabic
OntoNotes [53]	Shared task	English, Mandarin Chinese, Arabic, and Chinese
AnCora-CO [44]	Pronouns, full noun phrases, and discourse segments	Spanish and Catalan
RuCor [45]	General nominal coreference	Russian
LitBank [54]	Long distance	English

	coreferences	
WEC [55]	Cross-document Event Coreference	English
Winogender [56]	Gender bias	English

Definitions of what is and what is not coreference vary among different corpora [57]. Each corpus also tends to focus on a few certain domains for their texts. This can lead to issues when CR approaches are trained (or evaluated) on different corpora. Due to that there have been recent efforts to create generalized and unified annotation guidelines that could be used between different types of corpora [58].

1.2.2 The process of creating a coreference corpus

The most obvious characteristic of a created corpus is its size. Depending on what kind of analysis is being done the size of the corpus can be defined differently: number of words, number of sentences, number of documents, number of specific expressions. In most cases the larger the corpus the better, but size is usually limited by practical considerations [59], such as copyright issues, research focus, and limited human or computational resources.

Next step is the collection of text documents that would be included in the created corpus. Documents can be selected either by external or internal criteria [59]. Internal criteria usually are the distribution of various linguistic expressions that are relevant to the research questions. But if a corpus is designed in such way, then it will be distorted and will not be representative of the specific domain.

Selecting by external criteria (such as domain or publication date) usually leads to more representative corpus, but there are no guarantees that such corpus will have all relevant linguistic expressions. Therefore, it is advisable to work in cyclical fashion – after texts are selected by external criteria they should be analysed by internal criteria and then further updated based on the analysis results [59].

Documents should be selected using random sampling, usually they end up being more representative even in rather specific domains [60]. The issue with this approach is that it might not capture linguistic expressions that are less frequent. To address this shortcoming, stratified random sampling might be used. It divides all available data into smaller strata based on some characteristics and randomly selects from those. Unfortunately, sometimes it is difficult to link the usage of specific linguistic expressions to specific characteristics of the document. Therefore, balance between what is optional and what is practicable has to be achieved when designing a corpus [61].

The creation of an annotation scheme depends entirely on the research questions that the created corpus should help in answering. But some practical concerns also have to be taken in mind. The more detailed the annotation scheme, the more time-consuming it will be for human annotators to use. The software used to help annotate the text for human annotators is also likely to be more complex with a more detailed annotation scheme. Lastly, it is important to note that CR models trained on one annotation scheme are likely to underperform on another [62]. Therefore, before creating a new annotation scheme, an already existing one has to be considered.

1.2.3 Evaluation strategies

Over the years, there have been multiple evaluation strategies suggested, but neither has been adopted as the standard of this field. In this section, the most popular evaluation strategies are covered.

MUC metric

Being one of the earliest corpora for CR, it naturally also suggested one of the earliest evaluation strategies. Core metrics of it are precision (*P*), recall (*R*) and F-measure (*F*).

Precision, (1.1), shows the percentage of correctly resolved (*C*) coreference expressions against the actual number of provided coreferences (*A*) by the annotator.

$$P = \frac{C}{A} \tag{1.1}$$

Recall, (1.2), shows the percentage of correctly resolved (*C*) coreference expressions against the total amount (*T*) of expressions pre-annotated in the text.

$$R = \frac{C}{T} \tag{1.2}$$

While the precision metric is rather straightforward, the recall has certain problems. First of all, not all coreferences might be in the scope of the research. As a result, the total amount of pre-annotated expressions might vary even if the evaluation is run against the same texts. The difference would be even bigger if texts were different and had higher, or lower, percentage of "out-of-scope" expressions.

There have been suggestions made by Byron [63] for result reporting guidelines proposing additional metrics called resolution rate (*RR*) that would replace recall and take into account excluded coreferences (*E*) next to the total amount of pre-annotated expressions, (1.3).

$$\mathrm{RR} = \frac{C}{T + E} \tag{1.3}$$

The obvious question here is, who defines what is excluded and what should not be covered by CR in general? As seen in Section 1.1.2, there are multiple expressions that are treated as coreferences by some researchers and not treated as such by others. It also does not address the issue of texts from different domains possibly having, on average, different numbers of certain expressions than texts from other domains.

Lastly, F-measure (1.4) is a harmonic mean of precision and recall:

$$\mathrm{F} = \frac{2PR}{P + R} \tag{1.4}$$

While no guidelines are provided, there is also a possibility (1.5) to assign an additional weight to either precision or recall.

$$F_\beta = \frac{(\beta^2 + 1)PR}{(\beta^2)P + R} \tag{1.5}$$

An alternative strategy was proposed for MUC-6 evaluation by Vilain et al. in 1995 [64]. The proposed model is link-based and assumes that each reference links two mentions and attempts to solve the transitivity problem. For example:

- In the text, there are three coreferences marked: A→B, B→C, and D→E.
- Annotator that is being evaluated marks: A→C and D→E.

Going by the original MUC metric, the precision would be 1/2 and recall would be 1/3. But if we assume that A, B, and C have a transitive relationship, then it is reasonable to claim that A→C marking is correct as well.

The proposed model constructs two groups of equivalence classes: key, and response. Key sets represent what was marked in the text for evaluation:

- {A B C} and {D E}

Response sets represent what was marked by the annotator:

- {A C} and {D E}

For recall calculation, the size of the key class is taken as *S* and it is subtracted by the number of partitions required from response class *p(S)* to match key class. Then each equivalence class is added up for total recall. The author proposes to calculate recall (1.6) by using the size of *S* and *p(S)* sets.

$$R_T = \frac{\sum|S| - |p(S)|}{\sum|S| - 1} \quad \textbf{(1.6)}$$

For the precision, the process is inverse (1.7), the size of response class is taken as S' and it is subtracted by the number of partitions required from the key class $p(S')$ to match the response class. Then each equivalence class is added up for total precision.

$$P_T = \frac{\sum|S'| - |p(S')|}{\sum|S'| - 1} \quad \textbf{(1.7)}$$

As the author notes, such an approach works only with equivalence classes and when the transition is enforced. It also favours results that over-merge entities. For example, if all mentions of different discourse-world entities in the text were merged into one coreference chain then it would result in a 100% recall and very high precision score.

B-cubed (B^3) metric

This mention-based metric was proposed as a response to MUC-6's evaluation model. It raises the important question of not all precision errors being equal when looking at the task from the information extraction viewpoint [65]. For example, we have three different equivalence classes: {A B}, {C E} {D F G}. If an annotator made a mistake by marking B→C relationship and as such merging the first two classes, then it should be considered as a smaller error than if it marked E→D relationship and merged the last two classes. The weight of the error is based on the size of the newly created class. This is relevant to information extraction tasks since the size of the error can greatly impact the end result. On the other hand, from a linguistic standpoint both errors should be treated equally.

Precision (P_i) for each entity is calculated (1.8) by taking the number of correctly

annotated entities in the equivalence class (*C*) and the total number of entities in the equivalence class (*A*).

$$P_i = \frac{C}{A} \tag{1.8}$$

Recall (R_i) for each entity is calculated (1.9) by taking the number of correctly annotated entities in the equivalence class (*C*) and the total number of entities in the pre-annotated equivalence class (*T*).

$$R_i = \frac{C}{T} \tag{1.9}$$

Then, final precision (1.10) and recall (1.11) are calculated. All previously calculated precision (P_i) and recall (R_i) values are added up and multiplied by their assigned weight (w_i). By default, the author suggests dividing 1 by the number of total entities present in the text. But, if required, weights can be altered for each specific entity.

$$Precision = \sum_{i=1}^{N} w_i * P_i \tag{1.10}$$

$$Recall = \sum_{i=1}^{N} w_i * R_i \tag{1.11}$$

One of the arguments in favour of B-cubed against MUC link-based approach was that MUC evaluation did not deal with singleton mentions. Yet, if the annotator marked all mentions as singletons, B-cubed evaluation would result in 100% precision.

CEAF metric

CEAF is an entity-based metric [66] which attempts to evaluate similarities between entities. Entities, in this case, are similar to coreference chains – all mentions of one object in the text, form one entity. It provides two ways of scoring, mention-based and entity-based.

Entity-based approach (CEAFE) measures (1.12) how many of the same mentions two entities (*R* and *S*) share. It can also function as F-measure, (1.13).

$$\phi(\mathrm{R}, \mathrm{S}) = |R \cap S| \tag{1.12}$$

$$F(\mathrm{R}, \mathrm{S}) = \frac{2|R \cap S|}{|R| + |S|} \tag{1.13}$$

Mention-based approach (CEAFM) calculates recall (1.14) and precision (1.15) separately.

$$Recall = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \tag{1.14}$$

$$Precision = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)} \quad \textbf{(1.15)}$$

Here, g^* represents the Kuhn-Munkers algorithm (1.16) that is used to find the best mapping of the two entities.

$$\Phi(g^*) = \sum_{R \epsilon R_m^*} \phi(R, g^*(R)) \quad \textbf{(1.16)}$$

One of the flaws of this approach is that it does not take into consideration unaligned entities in the response set. The annotator might make a mistake and create two entities instead of one. CEAF would ignore the second coreference chain even if it had multiple right mentions linked.

CoNLL evaluation

During CoNLL-2012 shared task on CR it was decided that all three previously mentioned metrics have their benefits and drawbacks [67]. Therefore, instead of introducing a completely new metric, it was decided to take an average of their F-measures (1.17) as the evaluation score.

$$CoNLL = \frac{F_{MUC} + F_{B^3} + F_{CEAF}}{3} \quad \textbf{(1.17)}$$

This approach was originally proposed by Denis and Baldridge in 2009 [68]. But it is not clear why the average of three flawed numbers would not result in a fourth flawed number [69].

BLANC

BLANC is a link-based approach that adapts the Rand index [39]. It was later extended to cover not only gold but predicted mentions as well [70] [71]. It constructs 4 sets. Two key sets, one representing all coreference links (C_k) in the text and another representing all non-coreference links (N_k). Same for response sets, C_r and N_r.

Recall, Formulas (1.18) and (1.19), and precision, Formulas (1.20) and (1.21), are calculated for both coreference and non-coreference links.

$$R_c = \frac{|C_k \cap C_r|}{|C_k|} \quad \textbf{(1.18)}$$

$$R_n = \frac{|N_k \cap N_r|}{|N_k|} \quad \textbf{(1.19)}$$

$$P_c = \frac{|C_k \cap C_r|}{|C_r|} \quad \textbf{(1.20)}$$

$$P_n = \frac{|N_k \cap N_r|}{|N_r|} \quad \textbf{(1.21)}$$

After this, the F-measure is calculated for both co-referring and non-co-referring links. Their average is used as BLANC's final score. The problem with this approach

is that if the text has a high number of co-referring links, then naturally it also will have a very high number of non-co-referring links. This might result in higher precision and recall values than if the same annotator marked a less populated text.

LEA metric

LEA attempts to combine link and entity-based approaches for CR [69]. It is one of the newest evaluation methods and attempts to tackle various issues with previously mentioned metrics. It has a weighting mechanism called importance, but it functions similarly to the weights of B-cubed. It is based on the size of the entity ($|e|$) but can be adjusted according to domain needs. Additionally, the number of links (1.22) for each entity (e) with the number of mentions (n) is calculated.

$$links(e) = \mathrm{n} \times \frac{n-1}{2} \tag{1.22}$$

As in other approaches, key (k) and response (r) sets are used for recall (1.23) and precision (1.24) calculations. The role of key and response sets are reversed for the calculation of precision.

$$R = \frac{\sum_{k_i \in K}(|k_i| \times \sum_{r_j \in R} \frac{links|k_i \cap r_j|}{links|k_i|})}{\sum_{k_z \in K}(|k_z|)} \tag{1.23}$$

$$P = \frac{\sum_{r_i \in R}(|r_i| \times \sum_{k_j \in K} \frac{links|r_i \cap k_j|}{links|r_i|})}{\sum_{r_z \in R}(|r_z|)} \tag{1.24}$$

The previously covered evaluation strategies can be described as being linguistically agnostic – evaluation is treated as a clustering problem. What kind of coreferences the CR approach missed or solved are not taken into consideration, only the fact that they were missed or solved is important. And this might be good enough for linguistic research, but looking from the IE perspective, not all coreference links are equally valuable. For example, discourse-world entity being referred to by full name (*A*) is more valuable than the same entity being referred to by a pronoun (*B*). With linguistic agnostic evaluation strategies, CR approaches that resolved only *A* or *B* coreference link would be valued equally, but from IE perspective, an approach that solves coreference *A* is more valuable since otherwise we could lose important semantic information [72] [73]. Hence new, linguistically aware evaluation strategies have been proposed.

ARCS

The ARCS evaluation strategy assumes that different types of higher-level applications require different types of coreference annotation [74]. It defines three different types of higher-level applications and proposes a slightly different strategy for each type:

- Application that investigates distributions and patterns of entity occurrences in discourse. In such a case, immediate antecedent should

be selected for referent.

- Summarization and machine translation applications. In such a case, closest nominal antecedent should be selected for referent.
- For query driven applications, an anchor mention should be selected for referent. An anchor mention is the first nominal antecedent in a coreference chain. This assumes that the first nominal antecedent in the text best describes the underlying discourse-world entity.

Four scores are aggregated over key and response sets:

- *TP*, true positive, where the referent is in the gold and response sets and the suggested link is correct.
- *WL*, wrong linkage, where the referent is in the gold and response sets, but the suggested link is incorrect.
- *FP*, false positive, where the referent is in the response set, but not in the gold one.
- *FN*, false negative, where the referent is in the gold set, but not in the suggested one.

F-measure is the standard harmonic mean of precision (1.25) and recall (1.26).

$$P = \frac{TP}{TP + FP + WL} \quad \textbf{(1.25)}$$

$$R = \frac{TP}{TP + FN + WL} \quad \textbf{(1.26)}$$

Prague Anapahora Score

The Prague Anaphora Score resembles the ARCS evaluation strategy [75]. It adds language-dependent variable spurious zero positive (*SZP*) that covers ellipses that should not be resolved by CR approach. This variable is used for precision (1.27) calculation, while recall and F-measure calculations are the same as in ARCS.

$$P = \frac{TP + SZP}{TP + FP + WL + SZP} \quad \textbf{(1.27)}$$

Additional difference is that different strategies for different high-level applications are not used. The coreference chain usually contains one element that does not refer to any other mention present in the text. The author suggests that such mention should be always treated as an anchor mention. The problem in this case is that such mentions might end up being a pronoun and less valuable than a noun that appears later in the text.

PARENT

PARENT divides all mentions present in the text into two disjoint subsets: *defining* and *non-defining* [76]. Defining mentions are those that carry enough semantic information that allows them to identify as discourse-world entities. A *non-defining* subset can be further divided into *referring* and *ignored*, not relevant for the evaluation process, subsets. This provides certain flexibility for the evaluation process. For example, if we want to evaluate pronoun linkage to definitive nouns then

all other types of mentions would be contained in an *ignored* subset. It focuses on finding relations between referring and defining mentions since they are more valuable than relations between two different referring mentions.

All mentions of one entity constitute one key set cluster (C_i^{key}) and response set cluster (C_i^{sys}). Relations for gold (G) set are defined in (1.28).

$$G = \{(m_{rl}^i, C_i^{key}) | \forall C_i^{key} \in C^{key} \forall m_{rl}^i, \in C_i^{key} \quad \textbf{(1.28)}$$

Here, m^i_{rl} is referring to mention that belongs to the gold set cluster. Relations for response (S) set are defined in (1.29).

$$S = \{(m_{rl}^i, [[m_{dk}^i]]\text{key}) | \forall C_i^{sys} \in C^{sys} \forall m_{rl}^i \in C_i^{sys} \forall m_{dk}^i \in C_i^{sys}\} \quad \textbf{(1.29)}$$

Here, m^i_{rl} is referring to a mention that belongs to the response set cluster and m^i_{dk} is defining the mention that it links to. Precision (1.30) and recall (1.31) are calculated using G and S. The F-measure is calculated in a standard way.

$$P = \frac{|G \cap S|}{|S|} \quad \textbf{(1.30)}$$

$$R = \frac{|G \cap S|}{G} \quad \textbf{(1.31)}$$

One of the issues with *defining* mentions is that it does not make a distinction between, for example, a full name and a partial name. Both describe discourse-world entity rather well, but a full name is clearly more informative and therefore more valuable for IE purposes.

These linguistically aware evaluation strategies, to some extent, solve the problem of all mentions being treated equally despite carrying different amount of semantic information. Another issue is the lack of coreference type representation in these evaluation strategies. As was covered in Section 1.1.2, there are many different coreference types and they can carry additional semantic information. But in all covered evaluation strategies it is evaluated only if the coreference link between an antecedent and a referent has been established, what kind of coreference link is established is not addressed. For example, the same person can be referred to by a full name and by their occupation. We can safely attribute semantic information linked to a full name as being relevant only for that person, but semantic information linked to occupation might be relevant not only for that person, but also for occupation itself. Being able to differentiate between such cases can lead to higher quality IE results. Consequently, correct identification of a coreference type should also be relevant to CR evaluation.

Furthermore, the use of coreference chains can cause confusion. Coreference chains assume that all mentions in the chain co-refer with each other, so transitivity between different mentions of the same chain is established. But in the case of group references, one-anaphora, ambiguities, and hypernyms transitivity should not be enforced. For example, two different persons can be politicians (group reference) and as such share certain traits that are linked to being a politician, but at the same time

they might have very different political views that are detailed in the text. Enforcing transitivity in such cases might lead to semantic information of these two politicians being linked to each other.

One of the goals of the evaluation process is to identify weak spots of the resolution approach and improve on them. There are some error categorization tools [77] that can help in this task, but they are limited to part-of-speech and span errors. Reason for that is that the existing evaluation metrics require to provide only resolved coreference without any additional details of what kind of coreference was resolved. If evaluation metrics required to provide a coreference type, then error identification and categorising would be more efficient. CR approaches often have different strategies for resolving different types of coreference therefore required data for such evaluation is already available internally but is not used.

Another issue is over-representation of certain coreference types in the corpus that is used for the evaluation purposes. Some coreference types, like hypernyms, are not used as often as, for example, pronouns. This can lead to misleading evaluation scores if CR approach solves very well those expressions that are over-represented and struggles with under-represented expressions present in the corpus. Such issues can be diminished by the usage of macro averages instead of micro averages [78]. While some papers presenting CR approaches provide scores with micro and macro averages in evaluation strategies themselves it is not specified which approach should be used. This sometimes leads to inconsistent score reports.

Lastly, while one final number (F score in most cases) is useful to describe which approach is overall better, it does have limited use. Assuming that we have two resolution approaches, where the first one focuses on pronouns and solves them very well but struggles with other expressions, and the second one does well with all types of expressions. Naturally, the second approach would have a higher F score, but if research or a system being created focuses on pronoun resolution then the first approach is preferable even if it has a lower F score. In such cases, the F score is not informative enough or can even be misleading.

1.3 Coreference resolution approaches

While no work to our knowledge has been done to solve coreferences in the Lithuanian language, there are many approaches available for other languages. This section is divided into three parts. In the first part, common constraints and preferences that are found in CR approaches are covered. In the second part, the overall evolution of CR approaches is overviewed focusing mostly on the English language since it has gotten most of the attention in this field. In the third part, state-of-the-art CR in grammatically similar language to Lithuanian is covered.

1.3.1 Constraints and preferences

Most of the resolution approaches employ certain constraints and preferences when resolving coreferences [79]. These usually are not hard rules that guarantee that the right candidate will be selected, or the wrong one will not be selected. Due to that, it is important to prioritize these preferences and have mechanisms capable of deciding if they should or should not select an antecedent for a referent that is

preferred in certain cases even if it is not the correct one. Most of the constraints and preferences can be roughly grouped into four categories that are further detailed.

Selection constraints

Selection constraints usually employ discourse-world knowledge, or semantic knowledge, to determine what relationships are possible between different text fragments in the document. Thus they are very useful for selecting the right antecedent for a referent, but they cannot be used as reliable rules due to a couple of factors:

- It is hard to quantify how much discourse-world knowledge we have and use when interpreting various texts. Therefore, it is very problematic to create a complete semantic model that would be able to cover most of the possible scenarios and would not contain some inconsistencies.
- Texts can intentionally have phrases that are illogical to make a certain point. In such cases, selection constraints would prevent us from selecting the correct antecedent.

There is also the case of different genres having different rules that cause additional problems. For example:

- The children ate cookies. They were delicious.
- The children ate cookies. They were happy.

In the first case, it can be assumed that the pronoun "they" refers to the cookies since normally children are not considered to be delicious. In second case, it can be assumed that the pronoun "they" also refers to children since cookies are inanimate objects and do not have feelings. But these assumptions might change if these sentences were present in a fairy tale or a horror story where cookies could be animate objects or children could be described as being delicious.

Agreement in gender and number

Probably the most common preference is for an antecedent and its referent to agree in gender and number. Depending on language, this information can be categorized as either semantic or morphological, if the language is morphologically rich. The Lithuanian language is morphologically rich therefore this information can be gained from the words themselves. Unfortunately, some of the word cases have identical word forms for singular and plural forms and additional context is required to determine the correct one.

And while this preference sounds rather straightforward it does have more complex cases like the plural pronoun, or NP, referring to multiple singular and plural NPs. Similarly, a plural referent might be of male gender and refer to multiple male and female antecedents.

Nearest candidate and sentence structure preferences

Most of the approaches tend to prioritize antecedent-referent pairs that are closer to each other than other possible pairs. This is a reasonable preference but often has to be overruled, for example:

- Tom told John about leaving early. He was ill.

In this case, John←He pair is closer to each other than Tom←He, but from the

context, we can infer that Tom←He is the correct choice. Alternatively, preference can be given to the main clause of the sentence or subject preference. But as with the nearest candidate preference these preferences also have to be often overruled.

Syntactic parallelism assumes that mentions that are in the same syntactic position in different sentences are more likely to refer to the same entity:

- Tom mixed red and green colours. Last week he mixed it with black colour.

Candidate antecedent "red" is parallel with referent "it" so it should be preferred over the "green" candidate antecedent.

C-command constraints

C-command refers to syntactic trees where one node commands another node by simply being directly above it in the parse tree. One of the usual c-command constraints is that NP cannot co-refer with the pronominal that c-commands it. For example:

- He told Jason to leave.

In this case "He" c-commands "Jason", therefore they cannot refer to the same entity. Naturally, the usage of such constraints requires the existence of a syntactic parse tree.

Overall, most of these preferences and constraints are useful regardless of the technique used for CR. At the same time, their usefulness is limited due to required resources (semantic knowledge, syntactic parse tree) being expensive and these rules having numerous exceptions.

1.3.2 Evolution and state-of-the-art in coreference resolution

CR approaches usually take many expensive resources. The task itself is sometimes called AI-complete [80], meaning that a fully functioning AI might be required to fully solve these expressions. Due to that, it is not possible to simply take the state-of-the-art approach developed for another language and port it to the Lithuanian language, as that ignores the problem of different languages having different properties. The goal of this section is to analyse existing approaches and their evolution to see what could be applied for the Lithuanian language.

It is difficult to classify CR approaches since many of them have underlying similarities and use the same, or similar, knowledge sources. In this section, the focus will be on the approaches that introduced new techniques or their novel usage.

Syntax-based approaches

One of the earliest CR methods was proposed by J. R. Hobbs in 1977 [81]. It is often referenced to as Hobbs's naive algorithm. Despite its age, it is still referenced and measured against. However, being one of the first attempts at solving coreference expressions, it does not try to solve all cases of them, only a subset of pronouns (*He*, *she*, *it*, and *they*) was covered. Additionally, the evaluation and all the preprocessing were done manually.

The algorithm assumes that a fully parsed syntactic tree exists. This is common for most of the syntax-based approaches. Next algorithm performs a left-to-right breadth-first search. Hobbs presents his algorithm in the following nine steps:

1. Begin at the *NP* node immediately dominating the pronoun.
2. Go up the tree to the first *NP* or *S* node encountered. Call this node *X*, and the path passed to reach it *p*.
3. Search in the subtree of *X* to the left of *p*. Propose as the antecedent any *NP* node that is encountered which has an *NP* or *S* node between it and *X*.
4. If node *X* is the highest *S* node in the sentence, traverse the surface parse trees of the previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when *NP* node is encountered, it is proposed as antecedent. If *X* is not the highest S node in the sentence, continue to step 5.
5. From node *X*, go up the tree to the first *NP* or *S* node encountered. Call this new node *X*, and call the path traversed to reach it *p*.
6. If *X* is an *NP* node and if path *p* to *X* did not pass through the *N* node that *X* immediately dominates, propose *X* as the antecedent.
7. Traverse all branches below node *X* to the *left* of path *p* in a left-to-right, breadth-first manner. Propose any *NP* node encountered as the antecedent.
8. If *X* is an *S* node, traverse all branches of node X to the *right* of path *p* in a left-to-right, breadth-first manner, but do not go below any *NP* or *S* node encountered. Propose any *NP* node encountered as the antecedent.
9. Go to step 4.

After collecting all candidates, the algorithm checks them against gender, number, and person constraints. Hobbs also suggests additional selection constraints based on discourse-world knowledge to improve precision. Such as:

- Dates cannot move;
- Places cannot move;
- Large fixed objects cannot move.

However, the utility of such constraints is limited and depends on the context of texts that are being parsed.

Another significant use of syntax has been done for the Czech language in the form of tectogrammatical layer that combines syntactic and limited semantic information. But since the Czech language is related to Lithuanian, it will be covered in Section 1.3.3.

Overall, syntax is often used for NLP-resource rich languages. But many less researched languages, like Lithuanian, lack such resources.

Centering theory

Kibble summarizes this theory in the following points [82]:

1. For each utterance in the discourse, there is precisely one entity that is the centre of attention.
2. There is a preference, formalized as **Rule 2**, (1) for consecutive utterances within a discourse segment to keep the same entity as the centre of attention, and (2) for the entity most prominently realized in an utterance to be identified as the centre of attention.
3. The centre of attention is the entity that is most likely to be pronominalized: this preference is formalized as **Rule 1**.

Centres link one utterance with other utterances in discourse. Each utterance has one backwards-looking centre (**Cb**) and a number of possible forward-looking centres (**Cf**) that a particular utterance has evoked. Forward-looking centres are ranked by discourse salience and grammatical rules, the highest rated centre is called the preferred centre (**Cp**).

One of the best-known approaches that utilize CT was presented by Brennan, Friedman, and Pollard in 1987 [83]. They divide their algorithm into three phases:

1. Construct the proposed anchors by creating all possible *<Cb, Cf>* pairs of the utterance. Additional *Cb* called *NIL* should be added as well since it is possible that current utterance does not have a valid *Cb* present in the discourse.
2. Filter the proposed anchors by contraindices and *CT* rules.
3. Classify remaining anchors by transitions and rank them.

Highest ranking pair of *<Cb, Cf>* is considered to be the most likely anchor of the utterance. One of the benefits of this approach is that it does not require semantic information. This approach is often referred to as **BFP**.

An alternative for this approach was proposed by Tetreault in 1999 called Left-Right Centering [84]. Notably, it does not use the second rule of centering theory and takes priority in searching the same sentence before looking in other sentences. For evaluation, Tetreault provides additional modifications of his suggested approach and achieves better results than Brennan, Friedman, and Pollard.

Tetreault also makes a point about centering not being a pronoun resolution method. Possibility to resolve pronouns is just a side effect of rules and constraints present in the centering theory. He also mentions that ranking should be affected by semantic information while both of the approaches perform ranking based only on syntactic and grammatical information.

Salience factors

While salience plays a role in most of the approaches, usually it is not considered as the main criteria for CR. A notable exception is **RAP** (Resolution of Anaphora Procedure) algorithm introduced by Lappin and Leass in 1994 [85]. The assumption is made that the most prominent word is likely to be the antecedent for the referent. Prominence is based on a number of salience factors, i. e., the recency of a sentence, the emphasis of a subject, existential emphasis, accusative emphasis, etc.

Only gender, number, and person of possible antecedents are taken into consideration, no other semantic information is used. A fully parsed syntactic tree is used to determine the subject, object, and other parts of the sentence. Salience factors and their weights are presented in Table 1.2. With each new sentence, weights of salience factors are degraded by a factor of 2. Precise weights were reached after empirical experimentation and numerous adjustments. One of the benefits of this approach is that weights can be adjusted to better suit different types of text being processed.

While this approach provided encouraging results, salience factors continued being mostly used as an additional feature for CR approaches rather than the focus.

Table 1.2 Salience factors and their weights [85]

Factor Type	Proposed weight	Explanation
Sentence recency	100	Priority is given to antecedents located in the same sentence
Subject emphasis	80	Subjects are more salient than other grammatical roles
Existential emphasis	70	Existential constructions (*There are...*) are more salient
Accusative emphasis	50	Direct objects are salient, but less than subjects
Indirect object emphasis	40	Indirect objects are less salient than direct objects
Head noun emphasis	80	NPs not contained in another NP are preferred
Non-adverbial emphasis	50	NPs not contained in adverbial constructions are preferred

Minimal models and Default Logic

It can be assumed that most of the CR approaches use, either directly or indirectly, minimal models when analysing text – only parts that are relevant to CR are analysed, and all other details are ignored. Default logic can be used to formalize these minimal models [86] [87].

The suggested approach uses DRT to linguistically formalise coreference expressions and uses Default Logic to determine correct antecedent and referent pairs. The assumption is made that if it cannot be proven that antecedent and the referent are not equal then they must be equal – equality by default. It acknowledges that there might be situations where multiple candidate antecedents cannot be proven to be not equal to referent but do not provide specific details on how such cases should be tackled.

Another novel idea is that every possible referent should have an antecedent. If it does not, then it should be considered as deictic – referring to some entity that is not present in the text itself. But since deictic expressions are outside of scope for this work, this addition is not very relevant.

Unfortunately, this approach was not implemented in practice, and it is difficult to evaluate how useful it could be.

Semantic knowledge and rule-based approaches

Semantically Enhanced Domain Specific Natural Language (SE-DSNL) is targeted at NLP purposes in general but can also be used for CR [88]. When compared

to other approaches, CR is rather simplistic. It uses only two features and focuses only on pronouns while other more recent approaches tend to encompass other expressions as well.

The first one is distance measuring in the syntax tree. The assumption is that a pronoun is more likely to refer to the closest candidate antecedent. Nodes that have the same governing node are more favourable (-1 to distance score) while nodes in different sentences are penalized (+1 to distance score).

The second feature determines semantic compatibility. It uses semantic values of the candidate antecedent and verb related to the pronoun. The assumption is that the antecedent and verb should be related in meaning for the pronoun to refer to that particular antecedent. For this task OWL ontology is used.

For the Tamil language, a CR approach based on Universal Networking Language (UNL) [89] was proposed. UNL is a formal language designed for representing semantic information of natural language texts and is presented in the hypergraph. Nodes represent concepts that were mentioned in the sentence and 46 types of relationships connecting them. For automatic construction of these graphs, semantic information about language is required. This is a difficult task due to differences between languages and requires language-specific rules.

Xrenner is one of the more recent (2016) rule-based approaches [90] that focuses on adaptability and mention-border definitions when there is no training data available for such cases. It highlighted the problem of domain-adaptation for learning-based approaches. The argument is made that it is easier to add a new rule (or alter an existing one) when domain specifics change than to create new training data.

CORP is another new solution that adapts various semantics-based rules for the Portuguese language [91]. Authors note that the Portuguese language does not have many NLP-related resources and as such syntactic-semantic rules are a viable alternative to learning-based approaches. To determine semantic relationships between candidate antecedents and referents Onto.PT [92], a lexical database based on the WordNet, is used.

Statistical methods

A salience-based algorithm (RAP) was further improved by assigning statistical values next to salience factors. In the case of multiple candidate antecedents having a similar (not exceeding specified threshold) salience value statistical value can be used to determine the preferred candidate. This improvement was introduced by Dagan as **RAPSTAT** [93]. The evaluation results of RAP vary in a 85–86% range, with the addition of statistics results improved to 89%. Interestingly, when the statistical component was used without salience values, it disagreed with RAP in 45% of the cases. Out of those, it was correct only in 21% of the cases. Hence the authors reason that statistical information can provide modest improvement to CR but is not efficient as the basis of resolution methods when compared to algorithms based on syntax.

One of the earliest fully statistical approaches with the minimal amount of hand-crafting was proposed by Ge, Hale and Charniak in 1998 [94]. It uses a small corpus of Penn Wall Street Journal Tree-bank [95] annotated with coreference information.

In their probabilistic model they consider the following:

- Distance between the pronoun and candidate antecedent.
- The syntactic situation in which the pronoun is present. Based on Hobbs naive algorithm.
- Gender, number, and animacy (world knowledge) of candidate antecedents.
- Interaction between the head constituent of pronoun and its candidate antecedents.
- Mention count. Repeatedly mentioned NPs are preferred.

The evaluation process involved investigating the relative importance of these factors while adding them incrementally. The biggest increases in precision were achieved by adding syntactic constraints and world knowledge. Noticeably, information about head constituents improved precision only by 2.2%. Authors attribute this to constraints not being clear cut and some of the verbs are too general to provide any additional constraints.

Genetic methods

Mitkov's 1998 algorithm uses a part-of-speech tagger and simple NP rules to identify possible referents and their antecedents [96]. To select preferable antecedent when there is more than one candidate algorithm uses scoring indicators: definiteness, givenness, indicating verbs, lexical reiteration, a section heading preference, non-prepositional NPs, collocation pattern preference, immediate reference, referential distance, term preference. Each candidate is scored for these indicators and the candidate with the highest score is picked as the antecedent for the referent.

MARS is a reimplementation of Mitkov's original 1998 algorithm with genetic algorithms [97]. Additionally, it adds 3 new indicators and operates as an end-to-end system, while earlier work relied on pre-processed data.

Machine learning

The first learning system to achieve comparable results with other approaches was presented by Soon, Ng, and Lim [98] in 2001. An important trait of their system is that it is an end-to-end system including tokenization and segmentation, morphological processing, POS tagging, NP identification, named-entity recognition, nested NP extraction, and semantic class determination. For the CR Decision Tree, C5 and Closest First Clustering algorithms are used.

To improve the learning capabilities of the engine, authors introduced 12 feature vectors that determine if two annotated objects (annotated by previously mentioned parts of their system) co-refer or not. In the following feature list, *i* is potential antecedent and *j* is the referent:

1. Distance between *i* and *j*.
2. If *i* is a pronoun.
3. If *j* is a pronoun.
4. If *i* and *j* strings match. Before attempting match articles and the demonstrative pronouns are removed so the license matches this license.
5. If *j* is a definitive pronoun.
6. If *j* is a demonstrative pronoun.
7. If *i* and *j* agree in number.

8. If *i* and *j* agree in semantic class. The proposed system has a number of semantic classes such as a female, male, person. This feature vector checks if *i* and *j* have the same semantic class or share a subclass-parent relationship.
9. If *i* and *j* agree in gender.
10. If *i* and *j* are proper names. Value is determined based on capitalization.
11. If *i* and *j* are aliases of each other.
12. If *j* is in the apposition of *i*. In this case, usually, appositive *j* is separated by a comma from antecedent *i*. Additionally, either *i* or *j* must be a proper name.

Ng and Cardie expanded on this work [99]. They improved the linguistics of the learning framework and increased the number of feature vectors from 12 to 53. This expansion provided mixed results. Linguistic modifications increased precision. However, with a full set of additional feature vectors recall improved, but precision dropped significantly. This problem was attributed to poor common noun resolution and data fragmentation having problems with a large feature set.

Another machine learning approach ILP [100] uses logistic classifier algorithm and integer linear programming. Transitivity constraints are implemented in this approach. The antecedent-pair model is still widely used, but transitivity cannot always be enforced due to ambiguities and different types of coreferences having different relationships between its antecedent and referent.

Fernandes et al.'s proposed model [101] introduces two modeling approaches: latent coreference trees and entropy guided feature induction. It was tested in the English, Arabic, and Chinese languages. Performance drops in Arabic and Chinese, but this is attributed to these languages having smaller training corpora and feature limitations. A lack of high-quality NLP resources remains a major problem of the coreferences resolution regardless of the approach.

Semantic role labelling

Semantic role labelling (SRL) is a technique used for deriving a semantic meaning behind sentences in a structural manner.

Consider the following examples:

(a) Tom sold a car to John.

(b) John bought a car from Tom.

For the human reader, it is obvious that in both cases the same situation is described, Tom is a seller of a car and John is a buyer. But for natural language processors, it is not as obvious since the mentioned sentences have different grammatical and syntactical structures.

PropBank [102], one of the available SRL resources, suggests the particular framework for sentences describing an act of selling:

- Arg0: seller
- Arg1: the thing sold
- Arg2: buyer
- Arg3: the price paid
- Arg4: beneficiary

Using this framework, both sentences can be annotated in this manner:

(a) Tom [Arg0] sold a car [Arg1] to John [Arg2].

(b) John [Arg2] bought a car [Arg1] from Tom [Arg0].

With such annotations, automated algorithms can determine that both sentences are describing the same situation despite obvious differences in their structure.

PropBank, VerbNet [103], and FrameNet [104] are three of the most widely used resources for these tasks. They provide corpora with semantic role annotations and frameworks for specific types of sentences (like the aforementioned act of selling). But, as with most of NLP resources, their support for Lithuanian language is missing. Moreover, these resources are criticized [105] due to roles lacking strict definitions, having inconsistencies and one syntactic argument having only one semantic role. These inconsistencies might cause problems for machine learners and, in general, cause difficulties for applications that depend on these resources.

In the context of CR, SRL is usually used as input data for machine learning algorithms. One of such solutions was proposed by Ponzetto and Strube [106]. As a base, it takes the previously mentioned machine learning approach by Soon et al. and slightly alters it. The main difference is the addition of two new features based on argument-predicate pairs. These pairs are created with ASSERT parser [107] that identifies all verb predicates in the sentences together with their semantic arguments as PropBank arguments. When evaluated, this approach achieved slightly higher results than the selected base approach with recall showing most of the gains – 1.9%.

Another machine learning approach [108] introduced four new features that classify if the referent and antecedent are agent or patient. During the evaluation, a 1-2% increase, depending on the size of the evaluation data, was observed when the results were compared to the baseline.

SRL has also been used in end-to-end semantic search systems [109], but not only for CR but for search query formation as well. For SRL construction, the ASSERT parser was used, while coreferences were resolved with the Gate tool [110]. Neither improvement in CR results due to SRL usage nor overall results of CR were provided, but improvement in relevant document selection based on the provided query was noted.

Deep learning

Due to the growing popularity of deep learning methods, they were also applied in CR. One of the earliest deep learning approaches for CR was developed by Wiseman et al. [111]. It introduces global features to CR. Global features allow scoring candidate antecedents on a global level, which helps to better determine their compatibility.

The state-of-the-art in deep learning CR could be considered another deep learning-based approach introduced by Lee et al. [112]. It is the end-to-end system that does not use any external tools.

After success with these and similar deep learning-based approaches there have been a number of attempts to add external knowledge to these approaches. One of such approaches by Zhang et al. [113] uses various real-world knowledge that is stored in triplets. But each object can have many triplets related to it and the majority of them

would be irrelevant when solving a single coreference expression. To solve this problem, a knowledge attention model is introduced that calculates individual scores for each candidate noun and pronoun pair. Based on this score, only the relevant information is taken from the knowledge base.

Another approach by Lai et al. [114] uses symbolic features extracted from the text that help in solving event coreferences. A novel addition in this approach is that during the training phase noise is added to the training data. This helps the model to identify more reliable features when solving event coreferences.

Both of these approaches are similar in that they use external knowledge but add filtering methods. This is done under the assumption that not all external data is relevant or even correct in every case.

Large language models

These models are based on deep learning and a very large number of parameters. They have received lots of attention recently due to their usage in artificial intelligence projects like ChatGPT and are able to resolve coreferences. Large language models are living and are updated constantly therefore it is difficult to evaluate their performance since by the time the results are published, they are often already outdated with the model developing new advantages and disadvantages. Nevertheless, some studies have been done [115] [116] and results have been encouraging. On the other hand, it is questioned how useful they can really be for coreference resolution considering that most of the systems are closed and very resource intensive [117].

Comparison of different approaches

A side-by-side comparison of resolution methods is provided in Table 1.3, while keeping in mind what kind of expressions they try to solve. MUC's metric is used since it is the oldest metric and was used by most of these approaches, while some older ones had only precision listed. This was done so that the comparison would be more accurate than listing various metrics. Some of the approaches have been evaluated numerous times with different results, but only metrics that were reported in the original research are listed. Approaches are ordered chronologically.

Approaches that were not implemented in practice, or provided small modifications to the already covered approaches, were not listed in the comparison table.

It is important to note that the listed evaluations were performed against different corpora, some against different languages, therefore, the evaluation results are not directly comparable. Moreover, it could be argued that recall is not a very accurate measurement as was covered, alongside other problems of coreference resolution evaluation, in Section 1.2.3. The main goal of such a comparison is to provide a general understanding about the achievable results.

Additionally, methods that are based on machine learning usually are end-to-end systems. They might have problems in other parts of the system that could cause inaccuracies in CR, while the method responsible for it would be performing well. Rule-based approaches usually avoid this problem since most of their data are handpicked and inaccuracies are solved before CR is done.

Table 1.3 A comparison of coreference resolution approaches

Method	Year	Foundation	Solved expressions	Precision	Recall	F1
Hobbs [81]	1986	Syntactic, selection constraint rules	Main pronouns: he, she, they, it	81–91%	-	-
BFP [83]	1987	Centring Theory	Pronominal	49–90%	-	-
RAP [85]	1994	Salience factors	Third person, reflexive and reciprocal pronouns	85–89%	-	-
Ge et al. [94]	1998	Bayesian rule	He, she, it and their forms	82–84%	-	-
L-R Centering [84]	1999	Modified Centring Theory	Pronominal	72–81%	-	-
Soon et al. [98]	2001	Machine learning	Nominal and pronominal	65–69%	53–56%	60–63%
MARS [97]	2002	Genetic algorithms	Pronominal	53–84%	-	-
ILP [100]	2008	Machine learning	Nominal and pronominal	78–89%	47–58%	61–68%
UNL appraoch [89]	2011	UNL semantics	Pronominal	67%	-	-
Fernandes et al. [101]	2012	Machine learning	Not specified	77–91%	65–71%	71–80%
SE-DSNL [88]	2013	Pattern based, semantic knowledge	Pronominal	60%	80%	70%
Wiseman et al. [111]	2015	Deep learning	Not specified	77%	70%	73%
Xrenner [90]	2016	Syntactic and semantic rules	Nominal and pronominal	51–55%	49–57%	49–56%
Veena et al. [108]	2017	Semantic role labelling	Not specified	67–85%	60–86%	63–85%
Lee et al. [112]	2017	Deep learning	Not specified	81%	73%	77%
CORP [91]	2018	Lexical-semantic rules	Nominal	45–64%	44–52%	48–55%
Zhang et al. [113]	2019	Deep learning, knowledge aware	Pronominal	75–95%	75–94%	75–95%
Lai et al [114]	2021	Deep learning, symbolic features	Event coreferences	-	-	56–58%
GPT-2 [115]	2022	Large language models	Not specified	37%	100%	54%
InstructGPT [116]	2023	Large language models	Not specified	71.1–89.6%	69.7–88.9%	70.4–89.2%

Early resolution approaches relied on various rules and algorithms for resolving coreferences. Usually, they were based on observations of these expressions and discourse-world knowledge. Eventually statistical and machine learning approaches gained popularity with increasing number and quality of corpus-based NLP tools (pre-annotated corpora, shallow parsers, etc.). Currently, deep learning is used as well to develop state-of-the-art approaches.

Overall, progression can be observed with movement from knowledge-rich and rule-based approaches to knowledge-poor and learning-based approaches in this field [118]. But as Xrenner and CORP examples show, rule-based approaches are still relevant and can be useful when expensive NLP resources are not available, the annotation scheme changes and (or) large training datasets are not available for particular tasks.

1.3.3 State-of-the-art in related languages

In this section, we cover the situation in related languages. The Lithuanian language belongs to the Balto-Slavic language family. Therefore, the Latvian, Polish, Russian and Czech languages are covered in this section.

The Latvian Language

To our knowledge, only one solution for the Latvian language (LV) has been developed, LVCoref [119]. 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. For evaluation purposes, the Latvian coreference corpus was constructed.

The Polish Language

One of the first CR approaches for the Polish language (PL) was rule-based Ruler [120]. For the scoring of candidates, it uses 5 rules: gender and number agreement, including (removal of nested groups) rules, lemma rules, Wordnet rules for nominal expressions, pronoun rules.

BARTEK is an adaptation of BART [121] for the Polish language [122]. The BART system was primarily designed for the English language, but its modular design makes it adaptable to other languages as well. At the time it supported 64 feature extractors, but due to a lack of language-specific resources for the Polish language, BARTEK-3 was able to utilize only 13.

A mixed Polish coreferences resolution approach combines neural network architecture with a sieve-based approach [123] [124] to achieve the best results for the Polish language. For training and evaluation, the Polish Coreference Corpus was used.

The Russian Language

RU-EVAL-2014 was an evaluation campaign of anaphora and coreferences resolution tools available for the Russian language (RU). An analysis paper provided data of 6 participants [125] that employed a wide variety of approaches. The evaluation was performed on Russian Coreference Corpus (RuCor). After the evaluation campaign, work on coreferences resolution for the Russian language

continued mainly with improving machine learning approaches [126] by Khadzhiiskaia and Sysoev.

There was another iteration of this evaluation campaign, RU-EVAl-2019 [127], but it did not provide detailed information on what type of systems were evaluated nor what were the common errors. It reported that best F-measure score was higher by 7.7% than from RU-EVAL-2014 but clarified that these results should not be directly compared due to evaluation being cared out in different settings.

The Czech language

One of the earliest Czech language approaches was based on activation. This theory was proposed by the linguists of the Prague group [128]. It introduces the concept of **Stock of Shared Knowledge** (SSK) that is available to speaker and hearer. It represents objects that are mentioned in the discourse, their properties, and mutual relationships. It is similar to the centring theory since it considers some of the objects closer to the attention of the hearer. Such objects are called *activated* and are similar in their function to centres. One of the big differences from centring theory is that the activated entity can be indirectly mentioned in the text. For example:

- Birds already started to migrate due to early autumn.

In such a sentence, migration is not directly mentioned but the reader can understand that one of the topics of the sentence is migration and it would be considered an activated entity for further understanding of the text. While analysis was done for the Czech language, it was argued that such an approach can be multilingual as well. But, to our knowledge, it was not implemented in practice neither for Czech nor for other languages.

Most of the work in this field for the Czech language (CZ) has been done in the context of Prague Dependency Treebank (PDT). It has three annotation layers: morphological, analytical, and tectogrammatical. Coreferences are usually annotated in the tectogrammatical layer, and their first CR approach was rule-based [129]. At first, all possible candidates are collected and then the list is narrowed down using eight filters, then from remaining candidates, the closest one to the co-referring object is selected as antecedent.

As with other languages, after initial rule-based attempts, there was a movement towards machine learning approaches [130]. Nguy et al. adapted two older English language approaches to the Czech language and used Decision Tree C5 for classifier-based approach, while the ranker-based approach employed Collins' (2002) averaged perception algorithm. Both approaches were trained and evaluated on PTD data with a ranker-based approach providing better results.

Treex CR is a part of Treex NLP framework [131]. It has been developed primarily for the Czech language, but since then has been successfully adapted to the English, Russian, and German languages. For the Czech and English (EN) languages, a parallel CzEng corpus was constructed and used while for the Russian and German languages (DE), English coreference labels were projected [132]. The projection-based approach produced notably lower results.

Further attempts at multilingual CR have been made using CorefUD corpus [133], which combines 11 smaller corpora from different languages, most of them

were covered in Section 1.2.1. This corpus attempts to harmonize coreference annotations from different corpora and languages to create a unified training dataset. Their CR approach uses deep learning to create two models, one for Slavic languages and one for all languages, then these models are joined in order to solve multilingual coreferences. Results showed that this approach benefitted the most languages that had smaller corpora. The Czech language, having the biggest corpora, performed worse than with non-joined models.

Comparison of different approaches

Like in the previous section, a side-by-side comparison is provided in Table 1.4. MUC's evaluation metric is used as well. Approaches are ordered by language and year. For language, letter codes are used to make the table more compact. Covered expressions are not detailed because newer approaches tend not to detail what kind of coreference expressions are ignored or not solved by a proposed approach, which is a common problem [63].

As can be seen from this analysis, initial coreference resolution approaches are usually rule-based that do not rely much on linguistic resources due to most of Balto-Slavic language being resource-scarce. And while machine and deep learning approaches are also used, they tend not to show significant improvement and sometimes even underperform due to lack of training data.

Table 1.4 A comparison of coreference resolution approaches for Balto-Slavic languages

Method	LN	Year	Foundation	Precision	Recall	F1
LVCoref [119]	LV	2014	Rule-based, Hobbs' algorithm	69–88%	66–80%	68–84%
Ruler [120]	PL	2011	Rule-based	59–65%	50–75%	55–69%
BARTEK [122]	PL	2012	Machine learning	58%	65%	61%
MIXED [123][124]	PL	2017–2018	Deep learning, sieve-based	70%	68%	69%
RU-sys1 [125]	RU	2014	Rule-based, ontology	82%	70%	76%
RU-sys2 [125]	RU	2014	Rule-based	71%	58%	64%
RU-sys3 [125]	RU	2014	Rule-based	63%	50%	55%
RU-sys4 [125]	RU	2014	Statistical, ontology	54%	51%	53%
RU-sys5 [125]	RU	2014	Machine learning, semantics	58%	42%	49%
RU-sys6 [125]	RU	2014	Decision tree	36%	15%	21%
Khadzhiiskaia, Sysoev [126]	RU	2017	Machine learning	84%	77%	80%
Kučová, Žabokrtský [129]	CZ	2005	Rule-based filters	60%	-	-
CZ-Classifier [130]	CZ	2009	Classifier-based machine learning	70–76%	70–76%	70–76%
CZ-Ranker framework [130]	CZ	2009	Ranker-based machine learning	79%	79%	79%
Treex CR (CZ, EN) [131]	CZ, EN	2017	Machine learning	-	-	61–68%
Treex CR (RU, DE) [132]	RU, DE	2017	Machine learning, projection	50–64%	15–24%	25–34%
CorefUD coreference resolution [133]	CZ, RU, PL, DE, ES, Catalan	2021	Deep learning	-	-	61–69%

1.4 Specified research tasks

After analysing related scientific literature, the following research tasks were specified:

1. Analyse current methods and resources used for CR in English and other languages;
2. Develop an annotation scheme and coreference corpus for the Lithuanian language that could be used for developing and evaluating CR approaches;
3. Develop a linguistically aware evaluation strategy suitable for evaluating CR approaches that takes advantage of the developed annotation scheme;
4. Develop rule-based CR models and algorithms for the Lithuanian language that use only lexical, morphological, and named entity annotations;
5. Implement CR models and algorithms suitable to use for coreference annotation of Lithuanian text corpora;
6. Conduct an experiment for evaluating the suitability of the created annotation scheme, CR models, and algorithms.

1.5 Summary of the analysis

1. Analysis of the coreference resolution field revealed that it is an important part of the NLP task and has been researched since the 1970s. Yet, despite that, it is still not a solved problem even for well-researched languages, like English.
2. Coreferences are a relevant topic for NLP task and linguistic research. It is important to make a distinction between those two fields. There is a big overlap between them, but the overall classification and priorities are different and can cause confusion. This research is NLP- and IE-oriented and as such it might not be relevant for strictly linguistic tasks.
3. An analysis of the coreference phenomenon revealed many different types of coreferences. They can further have different properties depending on the language that is being processed. Some of the coreference types found in the literature are questionable and it is not clear if they do not fall outside of the coreference resolution field.
4. The coreference corpus is a vital resource for the coreference resolution task. An integral part of it is the annotation scheme that defines what kind of expressions should be marked by the annotator, and how this should be done. It is the most language dependent resource required for coreference resolution.
5. After analysing the evaluation strategies for coreference resolution approaches, it was determined that none of the proposed metrics have been accepted as the standard. While some are more popular than others, all of them have certain drawbacks like enforced transitivity or lack of coreference type coverage. Evaluation metrics are not language dependent.
6. An analysis of the resolution approaches for major and related languages revealed a similar situation. The initial resolution approaches are usually rule-based; after that, the movement towards the machine and deep learning can be observed.
7. Adapting the coreference resolution approach from one language to another is

problematic due to differences between different languages and a lack of equivalent language-related resources. No coreference resolution approaches have been developed for the Lithuanian language.

8. Despite the prevalence of machine learning resolution approaches, rule-based approaches are still relevant due to higher adaptability and requiring less expensive language resources. This is very relevant for the Lithuanian language since it does not have many language-related resources. Due to these reasons, it was decided to develop a rule-based CR approach for the Lithuanian language.

2 COREFERENCE CORPUS FOR THE LITHUANIAN LANGUAGE

The work on CR in the Lithuanian language is done in the context of the semantic search for the Lithuanian language [3]. Due to that, the primary focus is the information extraction task and not necessarily covering issues relevant to linguistic research.

In order to improve the capabilities of CR for Lithuanian language it is not enough to only develop a CR approach that would be able to solve coreferences. This section presents the created resources and models that are important to the CR task, but do not solve coreferences themselves.

Some of the earlier analysed expressions, like bridging and presuppositions, are not considered as coreferences and are not covered by this research.

In Section 2.1, the proposed annotation scheme with examples is provided, while Section 2.2 covers dominant expressions and Section 2.3 details annotation guidelines. Section 2.4 provides an insight into the Lithuanian Coreference Corpus, its statistics and other relevant information. The results of these 4 sections have been published in [134] and [135]. Section 2.5 proposes a linguistically aware evaluation strategy for evaluating CR approaches, which has been also published in [136]. Finally, Section 2.6 summarises the created resources.

2.1 Annotation scheme

Before creating the annotation scheme, it was attempted to create a taxonomy of coreference expressions, at the time called anaphoric expressions, for the Lithuanian language [134]. Proposed taxonomy attempts to combine two different approaches: morphological and semantic classifications. Each approach is represented by a different layer with semantic classification further categorized into lexical and domain-specific semantics. This allows identifying coreference expressions from different viewpoints. It also reflects the actual situation that the same coreference expression may have a referent expressed as pronoun (morphological type), agent (lexical semantics type) and person (domain semantics type). It can be seen in Figure 2.1.

Unfortunately, to our knowledge, no suitable lexical semantics resources have been adapted for the Lithuanian language. Creating them is also outside of the scope of this research. Furthermore, the classification of domain semantics is based on the NER annotations and most of the time they have a 1:1 match. Therefore, adding such information to the coreference annotation scheme and coreference annotation layer is redundant. Due to these reasons, the created taxonomy is not very relevant to the current stage of the research.

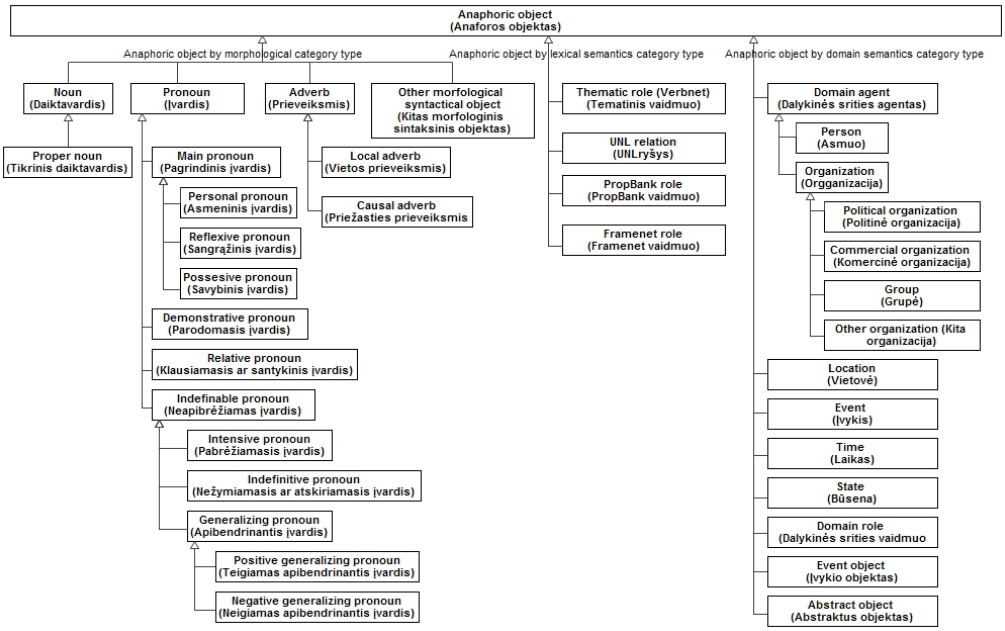


Figure 2.1 The taxonomy of anaphoric expressions [134]

Based on the created taxonomy, and taking its criticism in mind, an annotation scheme for the Lithuanian language was developed, it can be seen in Table 2.1. It is divided into four levels. On the first level, coreferences are grouped into four broad classes: pronominal, nominal (covers generic and proper nouns), ellipsis, and adverbs. Further, on the second level, more specific classes, specifying their parent classes, are defined. At the moment, the second level specifies only nominal, pronominal, and ellipsis coreferences. Third and fourth levels are global and are used by all types of coreferences. The third level determines if the referring object is pointing backwards (towards antecedent) or forwards (towards postecedent). The fourth level defines if the referring object is referring to one entity, a group of entities, or if it is ambiguous to which entity it is referring. Each class also has a letter specified in brackets. These letters are combined to define a specific code for each type of coreference. For example, if we have a pronominal (p) personal (p) anaphora (a) referring to multiple entities (g) then we would have "ppag" as the code of that specific coreference. In the case of adverbs, since they do not have the second level specified, they would form codes like this: "a-is".

Variations of the first two levels often are found in related scientific literature describing the type of referent. Certain types of coreferences, like anaphora and cataphora, differ only by their direction (a referent is either pointing backward or forward). In order to define such types and avoid duplication of the first two levels, it was decided to add a third level that would identify these and similar coreference types. A similar situation was with the fourth level as well. Group and ambiguous coreferences are also often treated as separate types of coreference but adding them as a fourth level classification also allows to identify them, specify the type of referent

and if they are anaphoric or cataphoric.

Adverbial coreferences could be defined with 3-letter codes instead of 4-letter ones since they do not have second level classification. It was decided to preserve 4-letter unified structure in case of future research that would provide meaningful second level classification. A similar situation was noted with ellipsis as well: in the initial version of the annotation scheme, ellipsis did not have second level classification [135], but after further research it was added to the scheme.

Table 2.1 Annotation scheme

First level	Second level	Third level	Fourth level
Pronominal (p)	Personal (p)	Position (a/p/i)	Group (g/a/s)
	Reflexive (r)		
	Possessive (o)		
	Relative (e)		
Nominal (g/d)	Repetition (t)		
	Partial repetition (a)		
	Abbreviations (b)		
	Feature (f)		
	Hyponymy/hypernymy (h)		
	Metonym (m)		
	Synonym (s)		
Adverbial (a)	-		
Ellipsis (e)	Same object (i)		
	Same type of object (y)		
	Verb phrase (v)		

Letter codes are unique for each level but are used repeatedly at different levels. For example, both the pronominal and personal levels have the code “p”. Some classes have multiple code options:

- Nominal coreferences can be expressed with regular nouns (g) or proper nouns (d).
- Position determines which way, backward or forward, the referent is pointing towards; this is relevant since the Lithuanian language has free word order. Options are: backward (antecedent, usually called anaphora, a), forward (postecedent, usually called cataphora, p) and in some cases direction might be irrelevant (i).
- Group determines if the referent is referring to a single entity (s), to multiple entities (g) or if it is ambiguous (a).

In the presented annotation scheme, we propose to tackle ambiguity the same as group references: “Tom hugged Jim, he was happy”. The only difference being that in group references it would be considered that “he” is referring to both boys while in ambiguous cases it would be interpreted as “he” referring to one of the two options. During the evaluation [Tom ← he, code: ppas], [Jim ← he, code: ppas] and [Tom, Jim ← he, code: ppaa] should be considered valid annotations.

The main advantage of our approach is that our annotation scheme provides the

classification of coreferences (with their type codes) and guidelines on what should and should not be annotated or how certain annotations should be evaluated. At the same time, most of the implementation has been left open and easily adaptable to the specific needs of the research.

Next, four main classes are explained and examples in the Lithuanian (LT) and English (EN) languages are provided. In the examples, only coreferences that are relevant to a given class of coreferences are marked (with [] brackets and "c" subscript), other coreferences are ignored to make examples simpler. The code of the type is provided along with the example according to the proposed annotation scheme.

2.1.1 Pronominal coreferences

Pronominal coreferences are such relationships where the pronoun is referring to the NP. In most cases, it is relevant if the pronoun is pointing backwards or forward. These coreferences are further classified into four classes: personal, reflexive, possessive, and relative.

Personal pronouns are often used in the context of deixis (I, you), but there are cases like direct speech where it is possible to determine to whom deictic personal pronoun is referring, for example:

LT: "[Aš]$_c$ einu namo", pasakė [Tomas]$_c$.
EN: "[I]$_c$ am going home", said [Tom]$_c$.
Code: ppps.

Other personal pronouns are more straightforward, for example:

LT: [Tomas]$_c$ šiandien praleido mokyklą. [Jis]$_c$ sirgo.
EN: [Tom]$_c$ skipped school today. [He]$_c$ was sick.
Code: ppas.

Reflexive pronouns usually point toward the subject:

LT: [Tomas]$_c$ namų darbus padarė [pats]$_c$.
EN: [Tom]$_c$ did the homework [himself]$_c$.
Code: pras.

Possessive pronouns usually are embedded in the NP:

LT: [Tomas]$_c$ pamiršo [savo]$_c$ knygą.
EN: [Tom]$_c$ forgot [his]$_c$ book.
Code: poas.

Relative pronouns are usually used to join two different text fragments:

LT: Tomas pasiilgo [Džimo]$_c$, [kurio]$_c$ jis nematė nuo praėjusios žiemos.
EN: Tom missed [Jim]$_c$, [whom]$_c$ he had not seen since last winter.

Code: peas.

While singular pronouns usually refer to a single entity, plurals can refer to multiple entities:

LT: [Tomas]$_c$ ir [Džimas]$_c$ yra labai geri draugai. [Jie]$_c$ pažįsta vienas kitą nuo antros klasės.
EN: [Tom]$_c$ and [Jim]$_c$ are very good friends. [They]$_c$ know each other since second grade.
Code: ppag.

If multiple pronouns refer to the same entity, then instead of forming a coreference chain they all should link to the initial entity. For example:

- [Tom]$_{c1;c2;c3}$ skipped school today. [He]$_{c1}$ was sick, but [he]$_{c2}$ will have to do [his]$_{c3}$ homework still.

All three pronouns refer to Tom, instead of the first pronoun referring to Tom, the second one referring to the first one and the third one referring to the second pronoun. This is done so that each individual element of coreference annotation would provide a maximal amount of information possible. Linking two pronouns with each other might be linguistically sound, but from the viewpoint of IE, it does not provide much new or additional information. Hence, NP-pronoun pairing is preferable to pronoun-pronoun pairing.

2.1.2 Nominal coreferences

In the case of nominal coreferences, it is usually two NPs being coreferent. As mentioned previously, we note a difference between proper nouns and regular nouns. Unlike pronominal coreferences, it is often not important if the noun is pointing backwards or forwards, those cases where it is important will be mentioned separately. These coreferences are further classified into seven classes: repetitions, partial repetitions, abbreviations, features, hyponyms and hypernyms, metonyms and synonyms.

Repetition is a type of reference where the same noun is repeated multiple times and is referencing same discourse-world entity:

LT: [Bibliotikeninkas]$_c$ į darbą atvyko anksti. Bet netvarka palikta nuo vakar jam nepatiko ir dėl to [bibliotekininkas]$_c$ ir namo išėjo anksti.
EN: [Librarian]$_c$ came to work early. But the mess that was left from yesterday did not please him and due to that [librarian]$_c$ went home early as well.
Code: gtis.

Repetition of generic mentions is not linked:

LT: Liūtai yra laukiniai gyvūnai. Liūtai paprastai gyvena šeimomis.

EN: Lions are wild animals. Lions usually live in families.

No coreference is marked between two mentions of lions since they are not referring to any specific group of lions.

Partial repetition is more frequent with named entities when a certain person might be introduced by his full name at first, but later only the first or second name is used:

LT: [Tomas Petrauskas]$_c$ praleido pamokas. [Petrauskas]$_c$ sirgo.
EN: [Tom Petrauskas]$_c$ missed the school day. [Petrauskas]$_c$ was sick.
Code: dais.

There are multiple different techniques to shorten the words: abbreviations, contractions, crasis, acronyms, and initialisms. While their distinction is relevant in linguistic research it is not the case for IE focused CR research. Therefore, no distinction between them is made in this work and all of them are considered a form of abbreviation for simplicity. An Abbreviation is usually used with named entities where at first, we might get a full name of an organization, and later we get the abbreviated name only:

LT: [Kauno technologijų universitetas]$_c$ yra Kaune. [KTU]$_c$ yra didžiausias universitetas ten.
EN: [Kaunas University of Technology]$_c$ is located in Kaunas. [KTU]$_c$ is the biggest university there.
Code: dbis.

It is important to note here that abbreviations can contain many complex variations, for example, “Tom Petrauskas” can be altered into TP, T.P., Tom P., and T. Petrauskas. Acronyms can also look unnatural when translated into other languages. In the previous example, it looks like the abbreviation of “Kaunas University of Technology” should be “KUT”, but that is not the case since in Lithuanian language it is called “Kauno technologijos universitetas” and its acronym is “KTU“. Additionally, some words and symbols (like the hyphen “-”) might be omitted in abbreviations.

Feature references are most common when specific entities are referenced by one of their attributes. For example:

LT: [Dalia Grybauskaitė]$_c$ nebuvo patenkinta naujuoju Ministru Pirmininku. [Prezidentė]$_c$ pateikė aštrios kritikos jam.
EN: [Dalia Grybauskaite]$_c$ was not happy with the new Prime Minister. [The President]$_c$ had some harsh criticism directed at him.
Code: gfas.

Often to correctly identify such relationships we might need some contextual information. In this case, we should know when the article was published (or about what time period it is talking) to make sure that Dalia Grybauskaite was the president

of Lithuania at that time. Otherwise, it might be that Grybauskaitė and another president were not happy about the new prime minister. Additionally, position class allows us to mark if the feature of the entity or the entity itself was mentioned first in the text.

A hyponym is a word or a phrase whose meaning is included within the meaning of another word, its hypernym. Due to that, we can say that hyponyms and hypernyms form a hierarchical relationship. For example, "eagle" can be considered a hyponym of "bird", or in reverse: "bird" is a hypernym of "eagle". Obviously, we can go further with the "bird" being a hyponym of "animal" and "animal" being the hypernym of "bird" and "eagle". This hierarchy does not have to be well-defined, it might be rather abstract or derive structure more from the context than some discourse-world classification. In the context of CR, a hyponym and a hypernym can be used to refer to the same entity or part of the same entity:

LT: [Rinkėjai]$_c$ nebuvo patenkinti naujais valdžios planais. Dėl to [protestuotojai]$_c$ susirinko pagrindinėje aikštėje pareikšti savo nepritarimą.
EN: [Voters]$_c$ were not happy with new government plans. Due to that [protestors]$_c$ gathered in the main square to voice their disagreement.
Code: ghas or ghps.

Ignoring the pronoun "their", the relationship between "Voters" and "protestors" can be interpreted in two ways. Voters pointing towards protestors is interpreted as hypernym relation or protestors pointing backwards to voters is interpreted as hyponym relation. In order to avoid confusion, according to this scheme, the hyponym should always point towards its hypernym and [Voters ← protestors, code: ghas] would be considered as the only valid annotation when evaluating the results.

Metonym relation is similar to synonyms, but it is more dependent on the context of the text rather than grammatical classification. For example, in a political text, Russia, Moscow, and the Kremlin can refer to the same entity: the government of Russia.

LT: [Rusija]$_c$ buvo nepatinka pastarosiomis atakomis prieš Siriją. Dėl to [Kremlius]$_c$ paskelbė pranešimą smerkianti pastaruosius įvykius.
EN: [Russia]$_c$ was not happy with the recent attack on Syria. Due to that [Kremlin]$_c$ issued a statement condemning recent events.
Code: dmis.

Synonym relation is rather straightforward and similar to repetition with the only difference: instead of the same noun being repeated it is replaced with another noun having a similar meaning:

LT: Kai atėjau į restoraną, [palydovas]$_c$ laukė prie durų. Vėliau tas pats [padavėjas]$_c$ priėmė mano užsakymą.

EN: When I walked into the restaurant, [attendant]$_c$ was waiting near the door. Later the same [waiter]$_c$ came to take my order.
Code: gsis.

2.1.3 Adverbial coreferences

This category covers adverbial coreferences. There are multiple different adverb types in the Lithuanian language but separate second level classes are not specified for those because their usage and structure are fairly similar.

Place adverbs are used in coreference expression:

LT: Jonas nesenai nusipirko naują [namą]$_c$. [Ten]$_c$ jis pradės savo naują gyvenimą.
EN: John recently bought new [house]$_c$. [There]$_c$ he will start his new life.
Code: a-is

Cause adverbs used in coreference expression:

LT: Jonas nesenai [išėjo į pensiją]$_c$ ir [išvyko iš miesto]$_c$. [Dėl to]$_c$ niekas neužbaigė seno uosto projekto.
EN: John recently [retired]$_c$ and [left town]$_c$. [Due to that]$_c$ nobody finished old harbour project.
Code: a-ig.

Some adverbs cause and overlap with VP ellipsis that will be further elaborated on in the following section.

2.1.4 Ellipsis coreferences

Ellipsis is a linguistic expression in which a part of the phrase is omitted since its meaning can be understood anyway due to the context or things already mentioned in the text. In general, it can be said that a gap in the text refers back to the earlier mentioned phrase, or antecedent. This scheme covers three types of this phenomenon, the first is when the gap refers to the same object:

LT: [Tomas]$_c$ mate plėšiką. [Identifikavo]$_c$ jį kaip Džimą iš mokyklos.
Tomas mate plėšiką. Jis identifikavo jį kaip Džimą iš mokyklos.
EN: [Tom]$_c$ saw the burglar. []$_c$ Identified him as Jim from school.
Tom saw the burglar. He identified him as Jim from school.
Code: eias.

The meaning in both sentences is identical, but in the first one, the pronoun “he” is omitted since it is clear that the speaker is talking about Tom. We can see different markings in the English and Lithuanian languages. While in the English language usually the gap is marked, it is not a suitable solution for a free-word-order language like Lithuanian. Therefore, according to this scheme, instead of the gap, the predicate of the omitted subject should be marked.

The second case is when the gap refers to a different entity, but of the same type,

as previously mentioned:

LT: Šis Žalgirio [sezonas]$_c$ geras. Tikėkimes sekantis [bus]$_c$ taip pat geras.
Šis Žalgirio sezonas geras. Tikėkimes sekantis sezonas bus taip pat geras.
EN: Zalgiris is having a good [season]$_c$. Hopefully, next [one]$_c$ will be good too.
Zalgiris is having a good season. Hopefully, next season will be good too.
Code: eyas.

In this case, the omitted word is "basketball", but every year we have a different season. Therefore, in such cases, we have a different entity of the same type as in the previous sentence. In English such expressions are usually expressed with "one", although this is not the case in Lithuanian.

The last type of ellipsis is when the VP is omitted:

LT: Jonas [pristatė prezentaciją]$_c$. Sara [irgi]$_c$.
Jonas pristatė prezentaciją. Sara irgi pristatė prezentaciją.
EN: John [gave a presentation]$_c$. Sarah did too []$_c$.
John gave a presentation. Sarah also gave a presentation.
Code: a-is.

In this case, the VP "pristatė prezentaciją" is omitted. In Lithuanian, adverbs ("irgi") are often used in such cases to imply that something is being referenced back. As can been seen from the assigned code ("a-is"), such coreferences are treated as adverbial.

This subtype covers only cases when the gap or punctuation mark "–" is used:

LT: Jonas [gavo]$_c$ tris obuolius. [Sara]$_c$ du obuolius.
Jonas gavo tris obuolius. Sara gavo du obuolius.
EN: John [got]$_c$ three apples. Sarah did two []$_c$.
John got three apples. Sarah got two apples.
Code: evas.

As with previous subtypes, the gap is not marked due to free word order. But since VP is omitted, there are no suitable predicates to be marked. In such case, the subject is marked.

2.2 Dominant expressions

A dominant expression is an expression that carries the richest semantics or describes most precisely the discourse-world entity [50]. Alongside the coreference annotation, the coreference annotator should also provide a list of dominant expressions. Expressions can be ordered by their dominance in the following order:

full named entity, abbreviated named entity, partial named entity, NP, and ellipsis referring back to the same object. Certain expressions like pronouns, adverbs and other types of ellipsis should not be listed as dominant since they do not carry any semantic information on their own.

If two or more expressions are of the same dominance level, then preference should be given to expression that appeared earlier in the text. Referent should always be less dominant than its antecedent. Hence, these expressions will be referred to as dominant mentions in the rest of this dissertation.

In order to determine the dominant expression, coreference chains have to be created first. This can be done either at the same time as individual coreference relationships are resolved or after it. It depends entirely on the specific implementation. Elements of the coreference chain have to be ordered by their appearance in text starting from the earliest to the latest. For example, we have this chain created:

- {[President]$_1$ [He]$_2$ [B. Obama]$_3$ [Obama]$_4$ [His]$_5$ [Barack Obama]$_6$ [Himself]$_7$ [Barack Obama]$_8$}

Subscript here indicates the order in which these mentions appeared in the text. Next, pronouns are filtered out since they do not carry any semantic information on their own:

- {[President]$_1$ [B. Obama]$_3$ [Obama]$_4$ [Barack Obama]$_6$ [Barack Obama]$_8$}

Then elements are ordered by their dominance:

- {[Barack Obama]$_6$ [Barack Obama]$_8$ [B. Obama]$_3$ [Obama]$_4$ [President]$_1$}

After these steps, the first element in the chain is selected as the dominant mention. The sixth overall element gets the preference over the eight element, due to it being present earlier in the text. This process is formalized using a pseudocode in (2.1).

```
Input: allEntities list containing all entities and their mentions present in the
text
Output: list of dominant mentions
List allEntities
List dominantMentions

Dictionary importanceOrder = {[full_name, 1], [abbreviated_name, 2],
[partial_name, 3], [noun_phrase, 4], [ellipsis_same, 5]}
                                                                        (2.1)
For entity in allEntities
  List allValidMentions = entity.getMentions.
          Where(importanceOrder.hasKey(x.mentionType))
  List sortedMentions =
allValidMentions.orderBy(importanceOrder.getValue(x.mentionType)).
        thenBy(x.startingPosition)
  dominantMentions.add(sortedMentions.first)
Endfor
```

Dominant mentions are similar to anchor and defining mentions that were

covered in Section 2.2. The advantage of dominant mentions is that it better selects the semantically richest mention. “President” would be selected as the anchor mention since it appears first in the text and is nominal. Defining mentions would not differentiate between different named entities (“Barack Obama”, “B. Obama” and “Obama”) and would consider them all equally important. But as we can see from the provided example, “Barack Obama” is clearly the semantically richest mention and as such the most important one.

The Stanford CoreNLP solution has a similar concept in representative mentions [137]. Unlike dominant mentions, they do not consider the order that mentions appear in the text. An additional issue is that preference is always given to the longest mention, which can lead to a less descriptive selection. For example, the last name of the person usually better describes him than his first name, yet the first name can be longer, and, in such case, a less descriptive proper noun would be selected.

A possible shortcoming of coreference chains themselves was also covered in Section 2.2. Instead of treating all elements in the chain as equal and therefore transitive with each other, dominant mentions are treated as the most important mentions. All other mentions just refer to the dominant mention and are its referents, thus transitivity between different referents is not required.

Dominant mentions are used in the proposed evaluation strategy (Section 2.5) and they would be useful for future research concerning exophoric coreferences. It would be easier to link the same entities from different texts if we had the semantically richest mentions in each text already identified.

2.3 Annotation guidelines

This section provides additional details relevant to the annotation scheme, which were not covered in the previous section. Examples are provided only in the English language unless something specific to Lithuanian has to be highlighted.

There is no standard on either maximal or minimal approach that should be taken when marking mentions. A maximal marking usually covers entire NP including grammatical modifiers while a minimal approach usually covers only head nouns. In this work, the minimal approach is taken. Various modifiers are considered as attributes of the mention and as such not part of the CR task. For example:

- President of the Republic of Lithuania Dalia Grybauskaitė is ending her term as the head of the state.

According to the proposed scheme only “Dalia Grybauskaitė” should be marked as an antecedent. The benefit of minimal marking is that it is more useful for coreference resolution between different texts. Maximal marking is likely to differ from text to text, while minimal marking should remain more consistent.

Mentions in coreference chains are often considered to be equivalent and co-refer to the same discourse-world entity, and each subsequent mention refers to the previous one rather than the first mention of the same entity. For example:

- $[\text{Tom}]_c$ missed the school. $[\text{He}]_c$ was sick.

In this situation, it can be said that “Tom” and “He” are referring to the same discourse-world entity – Tom, therefore what is said about one mention is valid for the other as well.

In the proposed scheme, such mentions in coreference chains are not considered equivalent due to multiple issues.

The first issue is that all mentions in the chain do not necessarily co-refer with each other. For example, when a generic mention is linked with a specific mention which in turn gets linked by another generic mention, we cannot say that both generic mentions will always co-refer:

- [Lions]$_{c1}$ are wild animals. So, it is not surprising that [Leo]$_{c1;c2}$ and [Savannah]$_{c1;c2}$ look rather dangerous. Despite that these [cats]$_{c2}$ always attract attention from tourists.

While both "Lions" and "cats" refer to Leo and Savannah, they are not coreferent with each other due to being generic mentions and having a hyponym/hypernym relationship with each other.

Another problematic case is group references, for example:

- [Tom]$_{c1;c2}$ and [Jim]$_{c1}$ are very good friends. [They]$_{c1}$ know each other since second grade. Unfortunately, [Tom]$_{c2}$ has been sick and hasn't seen his friend for a while.

In this case, the pronoun "They" refers to two entities, Tom and Jim. Here what is said about "them" is true for both Tom and Jim, yet what is later said about Tim is not true for both of them and Jim. A similar problem is created in a case of ambiguity. Therefore, a valid chain linking all mentions of the same entity cannot be created in these cases.

While relationships between an entity and its feature (if it is used in the place of said entity) are covered as coreferences, appositions themselves are not covered. For example:

- "Tom is a librarian."

We do not mark "librarian" as coreferent with "Tom", but, as seen in our earlier example with features, if later in the text "librarian" (as a reference to "Tom") were mentioned then we would link that later mention as coreferent with "Tom". Same applies to time functions, for example, if stock value changes are listed in the text. Each value represents an attribute of the specific stock in a given moment, but the value and the stock itself is not coreferent unless deliberately used in such a way, but that is also covered by our nominal feature type.

Unlike in many other annotation schemes, mentions that do not have any coreferences in the text are not marked. It makes the annotation process faster while at the same time no important data, in the context of IE, is lost.

Usually, closest antecedent-referent pairs are marked, for example:

- [Dalia Grybauskaitė]$_{c1}$ returned from the overseas trip tonight. [D. Grybauskaitė]$_{c1;c2}$ looked tired, but [she]$_{c2}$ immediately addressed the news concerning political changes in the country.

In this case, the following chain would usually be formed: [Dalia Grybauskaitė ← D. Grybauskaitė, D. Grybauskaitė ← she]. But what would be the difference if the pronoun linked to the first NE and not to the second one? Let us alter this chain with this question in mind: [Dalia Grybauskaitė ← D. Grybauskaitė, Dalia Grybauskaitė ← she]. Technically these are two different annotations and often only one of them would be considered correct. But looking from the IE angle, what actually changed?

From both annotations, we would be able to extract the same semantic information. Due to this reason, we believe that the annotation scheme should not enforce preferred annotation order in such cases and the evaluation model should be able to determine that both annotations are equivalent.

Nevertheless, order should be enforced in some cases. A referent should always be less dominant than its antecedent. If both of them are equally dominant, then the referent should be an expression that appears later in the text. In the case of feature mixed with a hypernym-hyponym relationship, the most recent antecedent should be marked. For example:

- [John]$_{c1}$ released a new book. [Author]$_{c1;c2}$ in this book talks a lot about climate change. This [novelist]$_{c2}$ is known for his ecology views.

In this case, the correct order is [John ←Author, Author ← novelist], with the second element having a hypernymy-hyponymy subtype. If the novelist was instead linked to John with a feature subtype, then it would count as a mistake; more on the classification of mistakes in Section 2.5. The same preference applies to a synonym relationship as well.

2.4 Lithuanian Coreference Corpus

The source of texts for the developed corpus were articles from news sites that focus on domains of politics and economy. Articles from these domains are heavy on named entity mentions and quotations. One hundred (100) articles that would cover all cases that are relevant to the current stage of this research in CR were preselected.

The selection process involved random sampling from the articles available to the Semantic Search Framework. After that, they were overviewed to see what type of coreferences they had and an additional round of sampling was done if less common coreference expressions, in our experience, it was various forms of ellipses, were not present in those articles. These new articles would replace articles from previous sampling that were redundant – had same type of coreferences as many other articles.

Grammatical errors found in the articles were corrected so that they would not negatively impact morphological annotations. Original versions were archived and stored separately in case of future research where grammatical errors would be taken in mind.

Coreference annotations are stored in JSON format with the following structure of coreference chain element:

- [{"Mentions": [["0, 3"], ["8, 3"]], “Referent": ["35, 4"], "Type": "ppag"}] for following fragment: “Tom and Jim are very good friends. They know each other since second grade.”
- In the “mentions” object all antecedents that were referred by “referent” phrase are listed. Usually “mentions” would have only one object, but it can contain multiple ones, as in the current example, if there was a group or ambiguous reference made.
- “Type” specifies what type of coreference has been identified by following the proposed annotation scheme.
- The annotation process uses a lexical segmentator, which annotates a starting position and the length of each lexeme. It allows us to identify the

starting position and the length of antecedent and referent phrases. Therefore, the first number is the starting position of the phrase in the text and the second number determines its length.

- All such chain elements are stored in a separate file and attached to a particular text that has been annotated.

This specific implementation has been done due to practical needs and is not tied to the proposed annotation scheme. If required, the implementation could be changed to a different format and structure.

In total, there now are 100 articles covering these cases and the following numbers of expressions:

- 1,217 repetition, partial repetition and abbreviations of nominal coreferences,
- 553 pronominal coreferences,
- 198 features,
- 61 hyponyms/hypernyms,
- 48 metonyms;
- 48 synonyms,
- 36 adverbial coreferences,
- 17 ellipses,
- 2,178 expressions in total.

The current version of the corpus is open for access via the Clarin-LT repository [138].

2.5 Evaluation strategy

In order to address the shortcomings of current CR evaluation strategies, which were covered in Section 2.2, a new, linguistically aware evaluation strategy is proposed. It expands on other linguistically aware strategies by adding coreference types to the evaluation process and uses dominant mentions that better describe the semantically richest mention than alternatives used in other strategies.

During the evaluation, two sets of annotations are compared against each other. One of them is manually created by experts, it will be referred to as the **gold set**, while the second one is created by the CR approach that has to be evaluated, and it will be referred to as the **response set**.

In this section we first define the main concepts of evaluation strategy, Figure 2.2, and later, the evaluation process itself is presented.

Coreference evaluation conceptual model

The conceptual model is divided into two parts. The first part covers the annotation structure and concepts related to annotations. The main concept in this part is *Set* that represents the collection of one or more *Coreference annotation layer*. Each document in the corpus that is being used for evaluation has one coreference annotation layer that is represented by the *Coreference annotation layer* concept. The coreference annotation layer has one or more annotations that are represented by the concept of *Annotation*. Each annotation has a *type* property that describes its dependency a coreference type, based on the annotation scheme presented in Section

2.1. Each annotation is composed of one referent, assumed by the *Referent* concept, and one or more antecedents, assumed by the *Antecedent* concept.

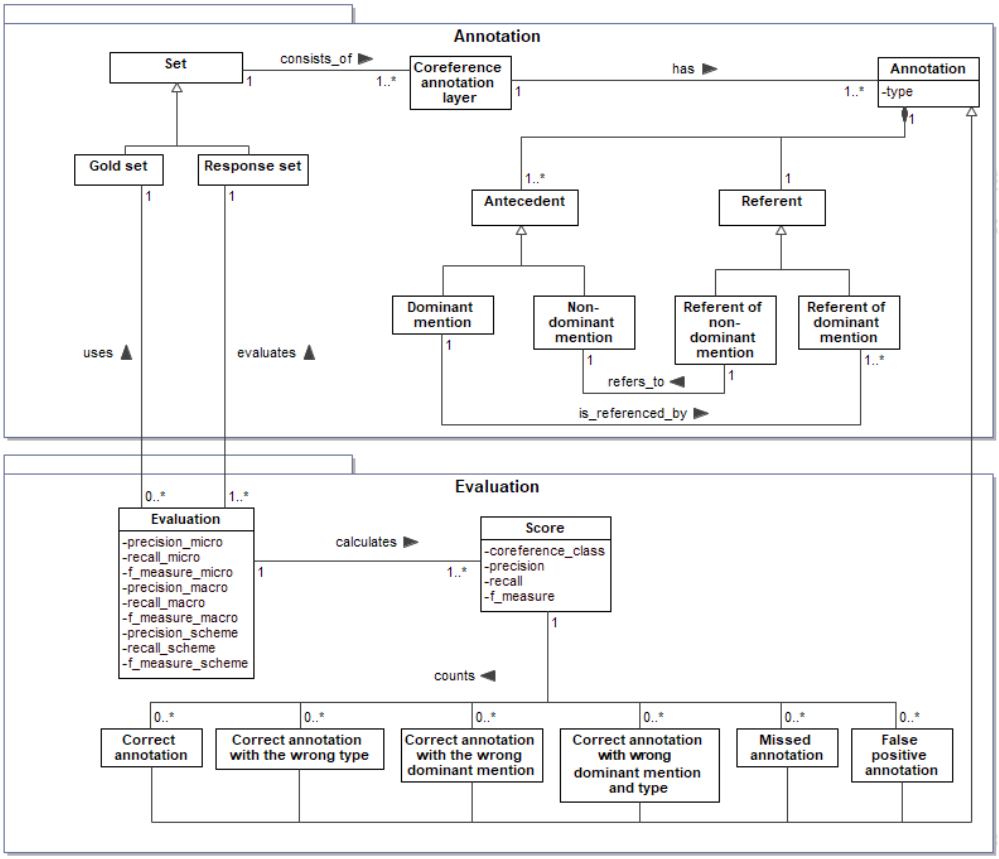


Figure 2.2 Conceptual evaluation model

In Section 2.2, it was explained that some antecedents, like pronouns and adverbs, cannot be dominant. In order to cover such cases, antecedents are specialized by the concepts *Dominant Mention* and *Non-dominant Mention*. Naturally, referents referring to them are also specialized by *Referent of dominant mention* and *Referent of non-dominant mention*. Dominant mentions can have more than one referent, while non-dominant mentions have only one referent.

The second part of the model covers the evaluation process. Evaluation itself is assumed by *Evaluation* and uses specialized *Set* concept *Gold set* that represents the previously mentioned gold annotation set and evaluates the specialized *Set* concept *Response set* that represents the previously mentioned CR-approach-created set of annotation layers. *Evaluation* has *precision_micro, recall_micro, f_measure_micro precision_macro, recall_macro, f_measure_macro, precision_scheme, recall_scheme* and *f_measure_scheme* properties that store final evaluation values.

Each evaluation calculates one or more scores depending on how many different coreference classes an annotation scheme has these scores are assumed by *Score*

concept. Each *Score* has a *coreference_class* property declaring for which coreference class *precision*, *recall* and *f_measure* properties were calculated.

Evaluation is performed by following the annotation scheme (Section 2.1) and annotation guidelines (2.3 section). Based on their correctness, each annotation is assigned to one of six different concepts: *Correct annotation*, *Correct annotation with the wrong type*, *Correct annotation with the wrong dominant mention*, *Correct annotation with wrong dominant mention and type*, *Missed annotation*, *False positive annotation*. Each of these concepts specializes the *Annotation* concept. These assigned annotations are used in *Score* calculations.

Correct annotation assumes annotations that have correct coreference type specified and are linked to correct dominant mention. Annotations that have correct coreference type specified but are linked to wrong dominant mention are assumed by *Correct annotation with the wrong dominant mention* concept. Annotations that are linked to the wrong entity (not only linked to the wrong dominant mention but completely different entity) are assumed by *False positive annotation* regardless if the identified referent is anaphoric or not. Annotations that are present in *Gold set* but were not found in *Response set* are assumed by *Missed annotation* concept. If the annotation type of the annotation is incorrectly identified then, depending if it also linked to the correct dominant mention or not, it is assumed by either *Correct annotation with the wrong type* or *Correct annotation with wrong dominant mention and type*.

For *Missed annotation*, annotations from *Gold Set* are used since they are not found in the *Response Set*. For the other five concepts, annotations from the *Response Set* are used.

Evaluation process

Initially, all annotations found in the *gold* and *response* sets are assigned to one of the six classes based on how correct or wrong they are. Then precision, recall and f measure are calculated for each coreference class (first letter in the annotation scheme, Section 2.1). For the calculation of precision and recall, additional coefficients are assigned to each class:

- The number of annotations assigned to *Correct annotation (TP)* concept get k_1 coefficient.
- The number of annotations assigned to *Correct annotation with the wrong type (WT)* concept get k_2 coefficient.
- The number of annotations assigned to *Correct annotation with the wrong dominant mention (WL)* concept get k_3 coefficient.
- The number of annotations assigned to *Correct annotation with wrong dominant mention and type (WTL)* concept get k_4 coefficient.
- The number of annotations assigned to *Missed annotation (FN)* and *False positive annotation (FP)* concepts do not get any coefficients.

The *TP*, *FN*, and *FP* classification of errors is common in other evaluation strategies as well. Some variation of the *WL* classification is usually found in linguistically aware evaluation strategies. With the proposed evaluation strategy, second linguistically aware classification of errors (*WT*) is added. Since we now have

two linguistically aware classes then a third one, a combination of both, is also required – *WTL*.

A range of values for coefficients: [0…1]. These coefficients allow differentiating between different types of errors. Since we have four different coefficients, we divide the range of the values into four equal parts and as a result, we get these coefficient values:

- k_1 – 1;
- k_2 – 0.75;
- k_3 – 0.5;
- k_4 – 0.25;

The separate calculations for each coreference class are useful in case we want to find a specialized CR approach that is suitable for a specific task. Furthermore, their precision and recall values can be used for macro average calculations of final *Evaluation* score.

In other evaluation metrics, it is usually not specified if micro or macro averages should be used when evaluating the CR approach. Micro average pools the performance over the smallest possible unit; in the context of CR it would be all coreference annotations. High micro F score indicates that the CR approach has good overall performance. On the other hand, macro average pools the performance from large groups; in the context of CR that would be different coreference classes. High macro F score indicates that the CR approach has good performance for each coreference class. An advantage of macro average is that it adjusts for an imbalanced coreference class distribution, which is usually found in the CR context. On the other hand, it could be argued that such imbalance actually represents discourse-world data and as such micro average is preferable. Hence, we propose to use both, micro and macro averages when evaluating coreference resolution approaches.

The annotation scheme that was presented in this work has five coreference classes, but the evaluation strategy is not tied to that number. There can be from one to *n* different coreference classes defined. Calculations are identical for precision (P_i) (2.2), recall (R_i) (2.3), and F-measure (F_i) (2.4) for each coreferences class.

$$P_i = \frac{k_1 TP + k_2 WT + k_3 WL + k_4 WTL}{TP + WT + WL + WTL + FP} = \frac{TP + 0.75 * WT + 0.5 * WL + 0.25 * WTL}{TP + WT + WL + WTL + FP} \quad \textbf{(2.2)}$$

$$R_i = \frac{k_1 TP + k_2 WT + k_3 WL + k_4 WTL}{TP + WT + WL + WTL + FN} = \frac{TP + 0.75 * WT + 0.5 * WL + 0.25 * WTL}{TP + WT + WL + WTL + FN} \quad \textbf{(2.3)}$$

$$F_i = \frac{2 P_i R_i}{P_i + R_i} \quad \textbf{(2.4)}$$

To diminish the impact of overrepresented classes of coreferences, macro precision (P_{macro}) (2.5), recall (R_{macro}), (2.6), and f-measure (F_{macro}) (2.7),are used for final evaluation scoring. Here n_a is a number of coreference classes that the CR approach attempted to resolve.

$$P_{macro} = \frac{\sum_i^{n_a} P_i}{n_a} \quad \textbf{(2.5)}$$

$$R_{macro} = \frac{\sum_i^{n_a} R_i}{n_a} \quad \textbf{(2.6)}$$

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}} \quad \textbf{(2.7)}$$

Next, we also calculate micro precision (P_{micro}) (2.8), recall (R_{micro}) (2.9), and f-measure (F_{micro}) (2.10).

$$P_{micro} = \frac{\sum_i^{n_a} k_1 TP_i + k_2 WT_i + k_3 WL_i + k_4 WTL_i}{TP_i + WT_i + WL_i + WTL_i + FP_i} \quad \textbf{(2.8)}$$

$$R_{micro} = \frac{\sum_i^{n_a} k_1 TP_i + k_2 WT_i + k_3 WL_i + k_4 WTL_i}{TP_i + WT_i + WL_i + WTL_i + FN_i} \quad \textbf{(2.9)}$$

$$F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \quad \textbf{(2.10)}$$

The purpose of these scores is to evaluate how well the CR approach resolves coreferences that it attempts to resolve. Naturally, the annotation scheme might have more coreference classes than the specific CR approach attempted to resolve. Separate calculations should be made to determine how well the proposed CR approach covers the used annotation scheme. For that purpose, we introduced the precision (P_{scheme}) (2.11), recall (R_{scheme}) (2.12), and f-measure (F_{scheme}) (2.13) values for annotation scheme coverage.

$$P_{scheme} = \frac{\sum_i^{n_a} w_{n_a} P_i}{n} \quad \textbf{(2.11)}$$

$$R_{scheme} = \frac{\sum_i^{n_a} w_{n_a} R_i}{n} \quad \textbf{(2.12)}$$

$$F_{scheme} = \frac{2P_{scheme}R_{scheme}}{P_{scheme} + R_{scheme}} \quad \textbf{(2.13)}$$

These look similar to previous formulas with two differences. The first difference is that division is performed not by n_a, but by n – the number of coreference classes present in the annotation scheme. Scheme coverage score heavily penalizes CR approaches that do not attempt to solve certain coreference classes. Another difference is that we add additional weight, w_{na}, for each P_i and R_i value. At the moment, we assign each of them the value of 1, therefore, it has no impact to the final score. We have it in place so that, if needed, the impact of different coreference class evaluation value could be altered. Macro averages are useful for dealing with imbalanced classes, but when we try to evaluate scheme coverage, they might be deemed giving too much value to the very small coreference classes, while micro

average would give it close to no value. Therefore, weighted macro average might provide a suitable middle ground. The sum of these weights should not be higher than *n* or otherwise, we could get results higher than 100%.

Overall, the presented evaluation strategy provides the following advantages:

1. The use of both macro and micro averages allow diminishing the impact of imbalanced classes to the final score and at the same time provides a score that is more representative of the discourse-world data.
2. Performing separate calculation for scheme coverage allows to distinguish between how well coreference resolution approach is doing what it attempts to do and how well it covers the annotation scheme.
3. The addition of coreference type identification in the evaluation process allows to better identify the weak points of the evaluated coreference resolution approach.
4. The addition of coreference type and dominant mention identification to the evaluation process allows to better evaluate to what extent additional semantic information is added by the coreferences resolution approach.

2.6 Conclusions of Chapter 2

1. Section 2.1 overviews the annotation scheme developed for the Lithuanian language and its classification of coreference expressions. Guidelines on how to use the scheme are presented in Section 2.3.
2. Dominant expressions, a mention that best describes discourse-world entity, and their place in proposed annotation scheme were covered in Section 2.2.
3. Using the proposed annotation scheme and the presented annotation guidelines, a corpus for Lithuanian language coreference expression was created in Section 2.4.
4. Finally, in order to take advantage of the developed annotation scheme and coreference corpus, a new, linguistically aware evaluation strategy was proposed in Section 2.5.

3 METHODS FOR SOLVING COREFERENCE EXPRESSIONS

The proposed CR approach is rule-based despite the machine and deep learning algorithms currently being more popular. This decision was made due to the following reasons:

- Rule-based solutions are still being developed since they have certain advantages, like easier adaptability, and provide comparable results when good training data is not available;
- Many of the more advanced solutions cannot be fully adapted for smaller languages due to a lack of available resources. Such is the case with the Lithuanian language as well;
- Having a working solution, even if limited, can be useful in the creation (and expansion) of additional resources like gold standard corpora;
- Solutions that are not heavy on linguistic resources can be very useful for resource-scarce languages in general.

Since this approach attempts to solve multiple types of coreferences, it is divided into a number of smaller algorithms that each deal with a specific type of references and their constructions. The context in which the algorithm operates is detailed in 3.1. A general view of the algorithm is given in Section 3.2. Section 3.3 specifies the main concepts of CR, while Section 3.4 discusses and formalizes each smaller algorithm. Dominant expressions are covered in Section 3.5 and additional knowledge bases that proposed approach uses, but are not directly related to NLP tasks, are presented in Section 3.6. Finally, Section 3.7 summarizes the presented methods and algorithms.

Results of this section, at their various stages of development, have been published in [139] [140] and [141].

3.1 NLP context for coreference resolution

The proposed resolution approach was implemented in SSFLL. The used NLP pipeline is shown in Figure 3.1. It does not cover all its components, but only those that are relevant to CR and this work in particular.

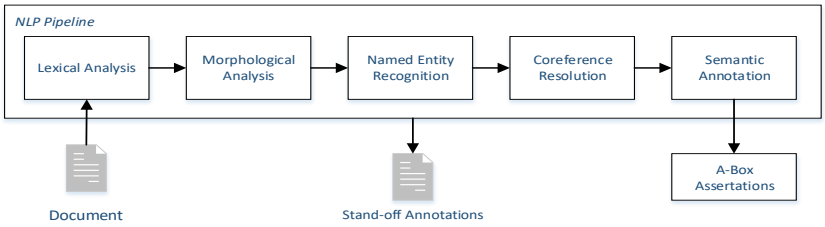


Figure 3.1 A complemented NLP pipeline of SSFLL

At first, a new document is taken and it is run through the chain of annotation components starting with Lexical Analysis and ending with Semantic Annotation. Each component produces stand-off annotations, meaning that the created annotations are saved in separate files and original documents, or previously created annotations are not modified. Annotations themselves are stored in the JSON data format.

Additionally, the Semantic Annotation component stores its results in the semantic database as well. OWL 2 [142] is used to later retrieve information from this database. The OWL 2 is a second major version of Web Ontology Language for the Semantic Web and is backwards compatible with the first one.

Below, each NLP component is overviewed and its output relevant to the proposed CR approach is detailed.

Lexical Analysis

The lexical analysis component performs a function that is usually performed by tokenizers. Normally, a tokenizer divides the text into distinct and meaningful tokens, these being: separate words, punctuation marks, etc. The lexical analysis component does that and additionally divides the text into sentences and paragraphs. So in total, it provides three layers of annotation: segments (in other works usually simply named as tokens), sentences, and paragraphs. The relevance of segments (tokens) to CR, and NLP, in general, were already covered in Section 1.1.4. An example of lexical annotation for the text fragment “Tomas praleido pamokas šiandien. Jis sirgo.” can be seen in Figure 3.2.

Figure 3.2 Lexical analysis annotation example

Morphological Analysis

The morphological Analysis component assigns a POS tag to each segment that has been marked by the Lexical Analysis component. It also performs a disambiguation task by ordering all possible interpretations from most likely to least likely.

The Lithuanian language is a morphologically rich language and therefore each word carries a significant amount of information about itself. Therefore, this component provides a wide variety of additional information next to the POS tag, all of which is encoded in multi-letter code. For example, segment “tarnautojų” can have two interpretations: Ncmpgn- and Ncfpgn-. An example of such annotation can be seen in Figure 3.3.

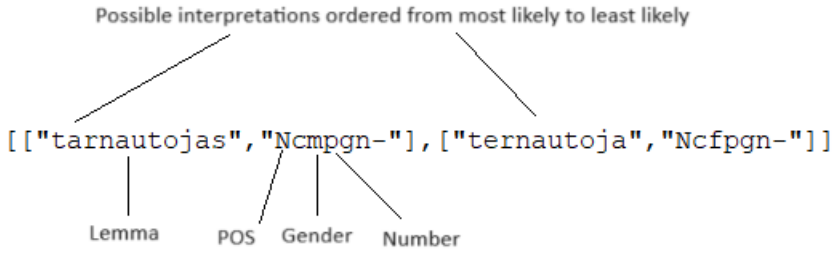


Figure 3.3 Morphological analysis annotation example

The first letter is used to filter out irrelevant segments based on their part-of-

speech. Third and fourth letters are used for gender (m – male, f – female) and number (s – singular, p – plural) agreements between the referent and candidate antecedents.

Named-Entity Recognition

The NER component creates a list of named entities (NE) that it identified in the provided document. Entities are classified into six classes: money, dates, products, organizations, locations, and persons.

Named entities are often referred to by referents and their identification allows the CR approach to better select the right antecedent that a referent should be linked to. Their additional classification allows applying different techniques and rules based on the type of named entity identified. What is true for persons might not be true for locations, etc. An example of annotation for the same sentence as in Lexical analysis can be seen in Figure 3.4.

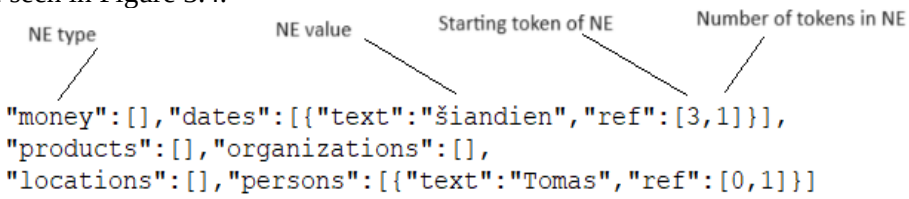


Figure 3.4 NER annotation example

Coreference Resolution

The workings of this component are detailed in Sections 3.2–3.6.

Semantic Annotation

The semantic Annotation component takes all previously created annotations and links discourse-world entities with relevant facts present in the text on their basis that later can be searched for by the end user. The extracted information is saved in RDF standard triples. A few examples of these triples can be seen in Table 3.1.

Table 3.1 RDF triple example

Triple	Explanation
<http://semantika.lt/ns/Agents#person~UID~Agents.person-4> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://semantika.lt/ns/Agents#person>	Declares that identified object Agents.person-4 is of person type.
<http://semantika.lt/ns/Agents#person~UID~Agents.person-4> <http://www.w3.org/2000/01/rdf-schema#label> \"E. Jesinas\"@lt	Declares label “E. Jesinas” for object Agents.person-4.
<http://semantika.lt/ns/Agents#person~UID~Agents.person-4> <http://semantika.lt/ns/Events#talked__talking> <http://semantika.lt/ns/Politics#talking~UID~Politics.talking-5>	Declares that Agents.person-4 has said something and that statement is identified as Politics.talking-5.
<http://semantika.lt/ns/Politics#talking~UID~Politics.talking-5> <http://www.w3.org/2000/01/rdf-schema#label> \"pristat\\u0117\"@lt	Declares label “pristatė“ for object Politics.Talking-5

These RDF triples are stored in the semantic database (OWLIM, newer versions are called GraphDB) and are used in accordance with a created ontology. This ontology [7] can be seen in Figure 3.5.

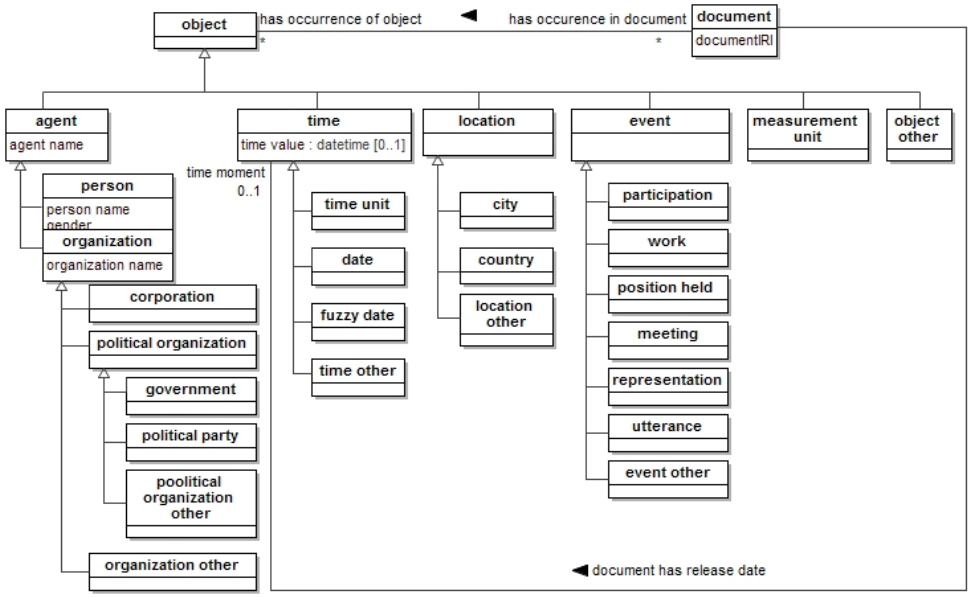


Figure 3.5 Example of ontology class hierarchy [7] (revised version)

3.2 Coreference resolution algorithm

For the resolution of a specific type of references, the proposed algorithm was divided into five smaller ones:

- A1: Specific rules resolution – an algorithm for the resolution of a certain usage of pronouns;
- A2: General pronoun resolution – an algorithm which focuses on the cases where pronouns refer to nouns (or NPs) that are recognized as named entities of "person" class;
- A3: PRA (partial, repetition, acronym) resolution – an algorithm for the resolution of nouns recognized as named entities and their repeated usage in the same text;
- A4: HHS (hypernym, hyponym, synonymous) resolution – an algorithm for the resolution of nouns recognized as profession names including their synonyms and hypernyms/hyponyms;
- A5: Feature resolution – and algorithm for the resolution of nouns that represent a certain feature (at the moment, only public position being held) of the named entity of "person" class that it refers to.

This does not mean that the algorithm correctly resolves, every single case of hypernym-hyponym relationship, but that it attempts to solve these kinds of expressions present in the text. At the time, no adverbial or ellipsis expressions are being solved by the proposed algorithm.

When the coreference annotation of a new text starts, each lexeme contained in the text is taken in the subsequent order. Based on the type of lexeme (pronoun, noun, other) and its attributes, or relationships to other sources of data, conditions are derived. If the condition is satisfied, a corresponding algorithm is activated.

The decision table with guidelines for the application of a certain resolution algorithm is shown in Table 3.2. 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, an answer '-' stands for 'not relevant'. In the lower right quadrant, an 'X' means that the algorithm should be applied and '-' means that it should not.

Table 3.2 A decision table for selection of the algorithm

Conditions	C1: Is a lexeme a pronoun?	Yes			No						Answers
	C2: Does a specific rule exist for this pronoun?	Yes		No	-						
	C3: Was the pronoun resolved by specific rules resolution?	Yes	No	-	-						
	C4: Is a lexeme a noun?	-	-	-	Yes					No	
	C5: Is the noun recognized as named entity?	-	-	-	Yes	No				-	
	C6: Does the noun exist in profession classification?	-	-	-	-	Yes		No		-	
	C7: Does the noun exist in the knowledge base of public persons?	-	-	-	-	Yes	No	Yes	No	-	
Algorithms	A1: Specific rules resolution	X		-	-	-	-	-	-	-	Decisions
	A2: General pronoun resolution	-	X	X	-	-	-	-	-	-	
	A3: PRA resolution	-	-	-	X	-	-	-	-	-	
	A4: HHS resolution	-	-	-	-	X		-	-	-	
	A5: Feature resolution	-	-	-	-	X	-	X	-	-	

For example, if the C2 condition is met then immediately A1 algorithm is activated.

The order of algorithms A3–A5 is not important since they do not overlap directly due to solving very different cases. On the other hand, the order of A1 and A2 is important. Specific rules tend to be more precise than general purpose algorithms and there is an overlap between cases that they attempt to solve. Switching them around could increase the number of false-positive results and decrease the overall precision of the solution.

Rules are based on the results of the analysis (specifically Sections 1.1.1, 1.1.2, and 1.3) and empirical observations during the development process. Certain resolution principles like those covered in Section 1.3.1 have been fully adapted for

the Lithuanian language. The adaptation of more specific rules found in other rule-based CR approaches was difficult due to the lack of syntax parser since many rules depend on syntactic tree parsing. Hence, rules and algorithms used by these approaches serve more as an inspiration than a source of adaptation.

The overall process of coreference resolution and the place of this decision table in it is formalized using pseudocode in (3.1).

```
Input: text document with various mentions that have to be solved
Output: fully formed coreference annotations for the provided text

List allCoreferenceAnnotations
List allEntities
List allDominantMentions

allMentions = getMentions(text)
For mention in allMentions
  determineAlgorithm(mention)
  //This is done using decision table presented in Table 3.2
  coreferenceAnnotation = resolve(mention)
  //Each algorithm is formalized in section 3.4
  allCoreferenceAnnotations.Add(coreferenceAnnotation)
Endfor

for coreferenceAnnotation in allCoreferenceAnnotations
  mention = coreferenceAnnotation.getMention
  referent = coreferenceAnnotation.getReferent                    (3.1)
  for entity in allEntities
    allEntityMentions = entity.getMentions
    if(allEntityMentions.Contains(mention))
      entity.addNewMention(referent)
    Else
      newEntity = createNewEntity(mention)
      newEntity.addNewMention(referent)
      allEntities.Add(newEntity)
    Endif
Endfor

for entity in allEntities
  dominantMention = determineDominantMention(entity)
 //Algorithm for dominant mentions is formalized in section 2.2
  allDominantMentions.Add(dominantMention)
Endfor

constructAnnotationFile(allEntities, allDominantMentions)
```

3.3 Concepts of coreference resolution

The algorithm was designed to serve the needs of semantic search in a Lithuanian text. The NLP pipeline of SSFLL presented in Section 3.1 provides the

main input flow for CR. The proposed algorithm takes into account 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. An algorithm uses two outside sources as additional input flows: Database of Public Persons and Classification of Professions. The algorithm is designed to provide coreference annotations in such a way that other parts of the system could interpret its results.

To formalize these algorithms, the concepts of CR domain are identified and expressed in the UML class diagram (Figure 3.6).

The main concepts of input flow the CR algorithms should analyse are *Text*, *Lexical_Unit*, and *Named_Entity*. The concept *Text* assumes an object such as a textual document or news article whose content should be analysed. The date of its publication is an important feature when solving coreferences related with a person's position. The text has a certain structure. Each text consists of at least one lexical unit. The paragraphs, sentences, words, punctuation, etc., are all assumed by the concept of *Lexical_Unit*, which, in our case, is classified into two categories – *Sentence* and *Lexeme*. The concept *Lexeme* assumes such lexical units as words, punctuations, and numbers. Each lexical unit has a certain value, starts at a certain position in the text, is of a certain length, belongs to only one text, can follow only one another lexical unit, and could be followed by only one another lexical unit in the text. Additionally, each lexeme is characterized by lemma and a part of speech, some of them (nouns, pronouns) – by gender and number also. Each lexeme is a part of only one sentence. Each sentence contains at least one lexeme. The lexeme could be specialized by the part of speech category. In our case, three categories are distinguished: *Noun*, *Pronoun*, and *Other_Part_Of_Speech*. In coreference resolution, only certain types of pronouns are of interest. Therefore, a type of pronoun should be specified. Special cases of *Other_Part_Of_Speech* are specializations *Comma (*it covers a punctuation mark comma, exclusively*)* and *Conjunction*. The concepts of *Comma* and *Conjunction* are required for the description of conditions of some CR rules.

A concept *Named_Entity* defines an object to whom pronouns or certain nouns can refer. A named-entity recognition (NER) algorithms usually recognize three types of entities: a person (*Person_NE*), an organization (*Organization_NE*), and a location (*Location_NE*). In each text, one or more named entities could be mentioned. Each named entity starts at a certain position in the text, is of a certain length, is expressed by at least one lexeme (for example, first and last name of person).

The named entities of person type require special attention in CR because not only pronouns are used for reference. Another way to mention a certain person is to use a position he/she holds. Additional reliable information about a person could help to resolve such coreferences more precisely. For example, a source of such information could be a Database of Public Persons (politicians, for example). The main concept of this database would be *Known_Person* – a well-known person, which can be mentioned as *Person_NE* in the text. The useful features of a known person would be his/her full name, gender, and positions he/she holds/held (*Position_Held*). It is important to know the name, the lemma of position name and dates which define a period a public person has held a specified position.

A profession can be used for referencing also. Therefore, additional source

about names of various professions, such as a classification of professions, would be helpful. A *Profession* is the main concept for such cases. Professions can be organized in a hierarchy – one profession can be broader than other professions. A certain profession can have more than one name (*Profession_Name*), in such case those names are synonymous. Each name has value and lemma.

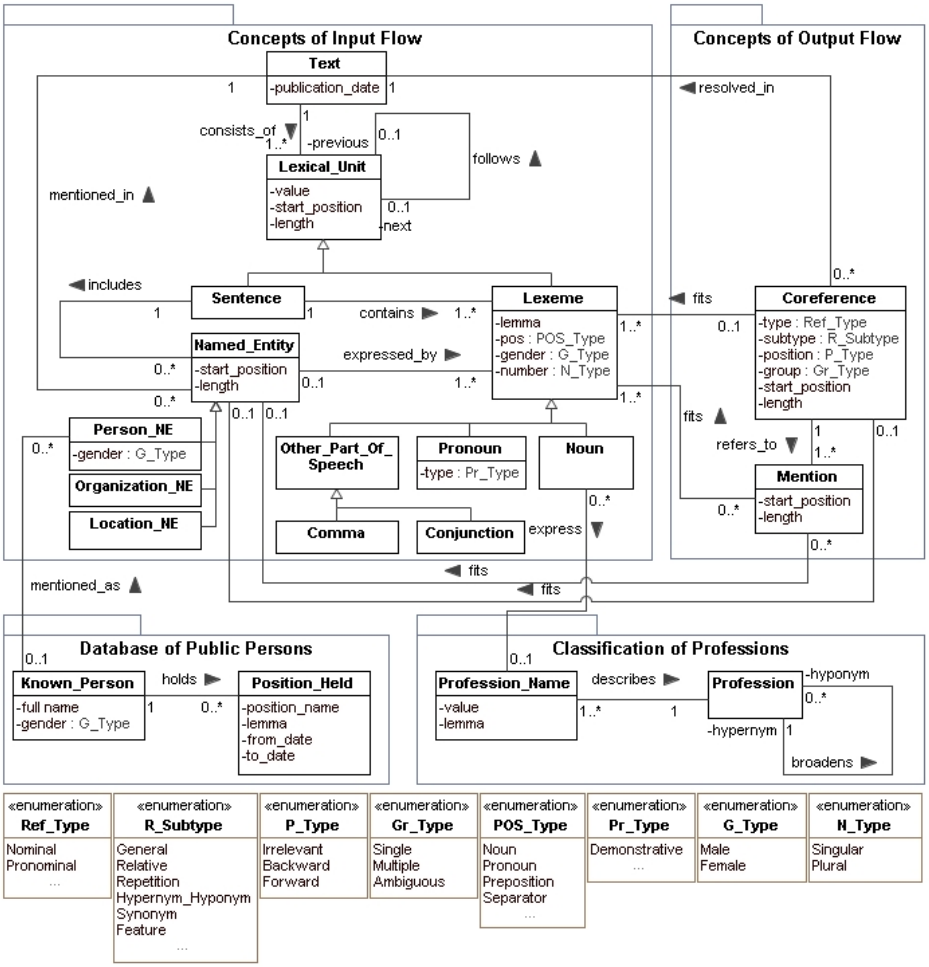


Figure 3.6 A concept model of coreference resolution domain

The main concept of output flow the CR algorithms produce is *Coreference* – a relationship between coreferents. For each coreference, its type (nominal, pronominal), subtype (relative pronoun, noun repetition), position (points backwards, forwards or irrelevant in case of repetitions), and group (is singular, refers to the group or is ambiguous) is specified. Also, each coreference holds a start position and length of the referent. The referent fits one *Lexeme* at least. Some of them can fit a certain *Named_Entity*. Each referent refers at least to one coreferent (a concept of *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*.

All concepts described in this section are used to formalize the proposed CR algorithm. A detailed explanation and a formalization of the algorithms are given in the next section. First-Order Logic (FOL) formulas are employed to define the main conditions the algorithms should check when resolving coreferences. All concepts of the model described above 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; the literals of enumerations – constants.

3.4 Explanation and formalization of coreference resolution algorithms

In this section, the logic of each smaller algorithm is explained and the main rules of their operation are formalized. The rules are described as FOL formulas and express the conditions for the existence of a certain type of coreference and its features. Examples for each case of resolution are provided in Lithuanian, Polish, and Russian languages to show that it can be successfully applied for CR in those languages as well. Additionally, examples are also provided in the English language.

A1. Specific rule resolution. In some cases, there exists a rather rigid structure for pronoun use and it can be easily defined by a specific rule. For example:

LT: Šiandien buvo atėjęs **vyras** [noun], **kuriuo** [pronoun] pasitikėjo Petras.
PL: Dzisiaj przychodził **mężczyzna** [noun], **który** [pronoun] skarżył się na ból pleców.
RU: Сегодня приходил **мужчина** [noun], **который** [pronoun] жаловался на боль в спине.
EN: A **man** [noun] **whom** [pronoun] Petras trusted have come today.

LT: Šiandien buvo atėjęs **vyras** [noun], **su** [preposition] **kuriuo** [pronoun] vakar išėjo Petras.
PL: Dzisiaj przychodził **mężczyzna** [noun], **z** [preposition] **który** [pronoun] skarżył się na ból pleców.
RU: Сегодня приходил **мужчина** [noun], **с** [preposition] **который** [pronoun] жаловался на боль в спине.
EN: A **man** [noun] **with** [preposition] **whom** [pronoun] Peter left yesterday has come today.

Both of these examples are similar in their construction: [noun] [comma] [optional preposition] [specific pronoun]. So, 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. Fundamentally it changes nothing about this construction, but it is important that the algorithm can consider such occurrences.

We can see that the structure of the sentence (number and order of lexemes) is similar in other languages as well. A pronoun goes after the comma and it refers to a

noun, compatibility of morphological features (gender, number) of noun and pronoun is kept. From the given example it can be understood that the coreference relation between pronoun and noun exists as well.

A condition for the existence of such reference formally could be 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 **sp1**, is of length **ln1**, follows comma **c** or follows prepositional lexeme **l1**, which follows comma **c**, and for every noun **l2**, which has a start position **sp2**, is of length **ln2**, 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 l**n2**, its referent starts at position **sp1** and has length **ln1**, and which fits only one lexeme **p** and refers to only one mention **m**, which starts at position **sp2**, has length **ln2**, and fits only one lexeme **l2**, exists, (3.2).*

$$\forall t, s, p, l1, c, l2, g, n, sp1, sp2, ln2.\ [\text{Text}(t) \wedge \text{Sentence}(s) \wedge \text{consists\_of}(t, s) \wedge \text{Pronoun}(p) \wedge \text{contains}(s, p) \wedge \text{has\_type}(p, \text{Relative}) \wedge \text{has\_start\_position}(p, sp1) \wedge \text{has\_length}(p, ln1) \wedge \text{Comma}(c) \wedge (\text{follows}(p, c) \vee (\text{Lexeme}(l1) \wedge \text{has\_pos}(l1, \text{Preposition}) \wedge \text{follows}(l1, c) \wedge \text{follows}(p, l1)) \wedge \text{Noun}(l2) \wedge \text{follows}(l2, c) \wedge \text{has\_gender}(p, g) \wedge \text{has\_gender}(l2, g) \wedge \text{has\_number}(p, n) \wedge \text{has\_number}(l2, n) \wedge \text{has\_start\_position}(l2, sp2) \wedge \text{has\_length}(l2, ln2) \rightarrow \exists! r\ \exists! m.\ [\text{Coreference}(r) \wedge \text{resolved\_in}(r, t) \wedge \text{has\_type}(r, \text{Pronominal}) \wedge \text{has\_subtype}(r, \text{Relative}) \wedge \text{has\_position}(r, \text{Backward}) \wedge \text{has\_ group}(r, \text{Single}) \wedge \text{has\_start\_position}(r, sp1) \wedge \text{has\_length}(r, ln1) \wedge \text{fits}(r, p) \wedge \text{Mention}(m) \wedge \text{refers\_to}(r, m) \wedge \text{has\_start\_position}(m, sp2) \wedge \text{has\_length}(m, ln2) \wedge \text{fits}(m, l2)]] \quad (3.2)$$

In other cases, the relative pronoun might be plural and refer to multiple singular (or multiple plural) nouns:

LT: Direktorius nerado **Tomo** [noun], **Lino** [noun], **Petro** [noun] **ir** [conjunction] **Eglės** [noun], **kurie** [pronoun] pabėgo iš mokyklos.

PL: Dyrektor nie znalazł **Tomas** [noun], **Linas** [noun], **Petras** [noun] **I** [conjunction] **Eglė** [noun], **którzy** [pronoun] uciekli ze szkoły.

RU: Директор не нашел **Томаса** [noun], **Линаса** [noun], **Петра** [noun] **и** [conjunction] **Эгле** [noun], **которые** [pronoun] сбежали из школы.

EN: The director did not find **Tom** [noun], **Linas** [noun], **Peter** [noun] **and** [conjunction] **Eglė** [noun], **who** [pronoun] ran away off the school.

In this situation, a plural pronoun "kurie" is referring to four singular nouns that have different genders. The previous rule would not be able to solve such a

coreference. For this case, the construction would be: [noun] [comma] [noun] [comma] [noun] [conjunction] [noun] [comma] [optional preposition] [specific pronoun]. For it, a special condition has to 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 **sp1**, is of length **ln1**, 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 **sp2**, is of length **ln2**, follows conjunction **j**, and for every noun **n2**, which precedes conjunction **j**, has a start position **sp3**, is of length **ln3**, and for every existing noun **n3**, which follows comma **c2**, and for every existing noun **n4**, which precedes comma **c2**, has a start position **sp4**, is of length **ln4**, the only one coreference relation **r**, which is resolved in text **t**, is of "Pronominal" type, "Relative" subtype, "Backward" position and "Multiple" group, its referent starts at position **sp1** and has length **ln1**, fits only one lexeme **p**, refers to only one mention **m1**, which starts at position **sp2**, has length **ln2**, and fits noun **n1**, refers to only one mention **m2**, which starts at position **sp3**, has length **ln3**, and fits only one noun **n2,** and refers at least to one mention **m3**, which starts at position **sp4**, has length **ln4,** and fits noun **n4**, exists, (3.3).*

$$\begin{aligned}&\forall t, s, p, l, c1, n1, sp1, ln1, sp2, ln2, j, n2, sp3, ln3.[Text(t) \wedge Sentence(s) \wedge consists\_of(t, s) \wedge Pronoun(p) \wedge contains(s, p) \wedge has\_number(p, Plural) \wedge has\_type(p, Relative) \wedge has\_start\_position(p, sp1) \wedge has\_length(p, ln1) \wedge Comma(c1) \wedge (follows(p, c1) \vee (Lexeme(l) \wedge has\_pos(l, Preposition) \wedge follows(p, l) \wedge follows(l, c1)) \wedge Noun(n1) \wedge follows(c1, n1) \wedge has\_start\_position(n1, sp2) \wedge has\_length(n1, ln2) \wedge Conjunction(j) \wedge follows(n1, j) \wedge Noun(n2) \wedge follows(j, n2) \wedge has\_start\_position(n2, sp3) \wedge has\_length(n2, ln3) \wedge (\exists n3, c2, n4, sp4, ln4.(Noun(n3) \wedge Comma(c2) \wedge Noun(n4) \wedge follows(n3, c2) \wedge follows(c2, n4) \wedge has\_start\_position(n4, sp4) \wedge has\_length(n4, ln4)) \\ &\rightarrow \exists!r \exists!m1 \exists!m2 \exists m3. [Coreference(r) \wedge resolved\_in(r, t) \wedge has\_type(r, Pronominal) \wedge has\_subtype(r, Relative) \wedge has\_position(r, Backward) \wedge has\_ group(r, Multiple) \wedge has\_start\_position(r, sp1) \wedge has\_length(r, ln1) \wedge fits(r, p) \wedge Mention(m1) \wedge refers\_to(r, m1) \wedge has\_start\_position(m1, sp2) \wedge has\_length(m1, ln2) \wedge fits(m1, n1) \wedge Mention(m2) \wedge refers\_to(r, m2) \wedge has\_start\_position(m2, sp3) \wedge has\_length(m2, ln3) \wedge fits(m2, n2) \wedge Mention(m3) \wedge refers\_to(r, m3) \wedge has\_start\_position(m3, sp4) \wedge has\_length(m3, ln4) \wedge fits(m3, n4)]]\end{aligned} \quad (3.3)$$

These rules are further formalized using pseudocode in Formula (3.4).

Input: pronoun that has to be resolved
Output: coreference annotation

```
List coreferenceAnnotations
List candidateNouns

if(pronoun.isRelativePronoun)
  gender = pronoun.getGender
  number = pronoun.getNumber
  previousToken = pronoun.getPreviousToken
  if(previousToken.isComma || (previousToken.isPreposition &&
previousToken.getPreviousToken.isComma))
    if(previousToken.isComma)
      candidateNoun = previousToken.getPreviousToken
    Else
      candidateNoun = previousToken.getPreviousToken.getPreviousToken
    Endif
    if(candidateNoun.isNoun)
      candidateNouns.Add(candidateNoun)
      previousToken = candidateNoun.getPreviousToken
      while True
        if(previousToken.isComma || previousToken.isConjunction)
          candidateNoun = previousToken.getPreviousToken
          if(candidateNoun.isNoun)
            candidateNouns.Add(candidateNoun)
            previousToken = candidateNoun.getPreviousToken
          Else
            break
          Endif
        Else
          break
        Endif
      Endwhile
    Endif
    if(candidateNouns.Count = 1)
      candidateNoun = candidateNouns.getFirst
      if(candidateNoun.getGender = gender && candidateNoun.getNumber =
number)
        coreferenceAnnotations.Add(candidateNoun, pronoun, "peas")
    ElseIf(candidateNouns.Count > 1)
      coreferenceAnnotations.Add(candidateNouns, pronoun, "peag")
    Endif
  Endif
Endif
```

(3.4)

Relative pronoun resolution is often considered a trivial task if syntactic parsing is available. But syntactic parsing is expensive, both in terms of knowledge required and in computing time for the creation, and later in interpretation, of the parsed

syntactic tree. On the other hand, these simple rules can serve as a cheaper alternative that can provide the same results in CR context.

A2. General pronoun resolution. This algorithm focuses on the cases where pronouns refer to nouns (or NPs) that are recognized as named entities of "person" class by NER. It is a slightly modified version of the algorithm that was presented in [140]. The algorithm starts with the identification of non-demonstrative pronouns. In a given example below, such a pronoun is in the second sentence – "Ji" ("She"):

LT: **Dalia Grybauskaitė** [person NP] nuvyko į Vilnių. **Ji** [pronoun] pasveikino vilniečius su šventėmis.
PL: **Dalia Grybauskaitė** [person NP] pojechała do Wilna. **Ona** [pronoun] powitała mieszkańców Wilna uroczystościami.
RU: **Даля Грибаускайте** [person NP] отправилась в Вильнюс. **Она** [pronoun] приветствовала жителей Вильнюса с торжествами.
EN: **Dalia Grybauskaitė** [person NP] went to Vilnius. **She** [pronoun] greeted citizens of Vilnius with holidays.

If the pronoun is in the relative clause, the algorithm moves backwards analysing words going before the pronoun. In a given example, the pronoun is at the beginning of the sentence, so the remaining parts of the sentence are not analysed and the algorithm moves one sentence backwards. In the next sentence, it starts from the end and moves back towards the beginning of the sentence. The first named entity encountered is "Vilnių", but since it is recognized by NER as a location and not as a person it is discarded and the algorithm moves further backwards. The next named entity encountered is "Dalia Grybauskaitė", which is recognized by NER as a person. In this case, the grammatical compatibility between the NP (which consists of two nouns) and the pronoun is determined. Both are singular and of the female gender, therefore the algorithm connects them. The algorithm does not look for further candidates. Due to that, it can be considered naive since alternatives are not considered.

Conditions for the existence of such reference formally could be defined as three alternatives. The first one describes conditions for reference existing in the same sentence ***s1*** before pronoun ***p***:

For each text's ***t*** *sentence* ***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 in the same sentence* ***s1****, is expressed by lexeme l, and has gender* ***g****, number* ***n,*** *start position* ***sp2*** *and is of length* ***ln2****, and is before pronoun* ***p*** *(****sp2*** *is lower than* ***sp1****), but 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 (3.5).*

∀t, s1, p, l, e1, g, n, sp1, ln1, sp2, ln2.[Text(t) ∧ Sentence(s1) ∧ consists_of(t, 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(s1, e1) ∧ Lexeme(l) ∧ expressed_by(e1, l) ∧ has_gender(e1, g) ∧ has_number(e1, n) ∧ has_start_position(e1, sp2) ∧ has_length(e1, ln2) ∧ sp2<sp1 ∧ ¬(∃e2, e3, sp3, sp4. (e1≠e2 ∧ e1≠e3 ∧ e2≠e3 ∧ Person_NE(e2) ∧ includes(s1, e2) ∧ has_gender(e2, g) ∧ has_number(e2, n) ∧ has_start_position(e2, sp3) ∧ Person_NE(p3) ∧ includes(s1, e3) ∧ has_gender(e3, g) ∧ has_number(e3, n) ∧ 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, Backward) ∧ 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)]] (3.5)

The second alternative describes a case when a pronoun ***p*** refers to the named entity in the previous sentence ***s2***:

For each text's ***t*** *sentence* ***s1, s2****, where* ***s1*** *follows* ***s2****, 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 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, (3.6).*

∀t, s1, s2, p, l, e1, g, n, sp1, ln1, sp2, ln2.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ follows (s1, s2) ∧ 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_gender(e2, g) ∧ has_number(e2, n) ∧ has_start_position(e2, sp3) ∧ Person_NE(p3) ∧ includes(s2, e3) ∧ has_gender(e3, g) ∧ has_number(e3, n) ∧ has_start_position(e3, sp4) ∧ sp2>sp3 ∧ sp4>sp2)) (3.6)

→ ∃!r ∃!m. [Coreference(r) ∧ resolved_in(r, t) ∧ has_type (r, Pronominal) ∧ has_subtype (r, General) ∧ has_position(r, Backward) ∧ 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 third alternative describes a case when a pronoun ***p*** in the sentence ***s1*** refers to the named entity in the sentence ***s3***, preceding sentences ***s2*** and ***s1***:

For each text's ***t*** *sentence* ***s1, s2***, ***s3***, *where* ***s1*** *follows* ***s2*** *and* ***s2*** *follows* ***s3***, *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* ***s3***, *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*, *(3.7).*

∀t, s1, s2, s3, p, l, e1, g, n, sp1, ln1, sp2, ln2.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ Sentence(s3) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ consists_of(t, s3) ∧ follows (s1, s2) ∧ follows (s2, s3) ∧ 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) ∧ Lexeme(l) ∧ expressed_by(e1, l) ∧ includes(s3, e1) ∧ 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(s3, e2) ∧ has_gender(e2, g) ∧ has_number(e2, n) ∧ has_start_position(e2, sp3) ∧ Person_NE(e3) ∧ includes(s3, e3) ∧ has_gender(e3, g) ∧ has_number(e3, n) ∧ has_start_position(e3, sp4) ∧ sp2>sp3 ∧ sp4>sp2)) (3.7)
→ ∃!r ∃!m. [Coreference(r) ∧ resolved_in(r, t) ∧ has_type (r, Pronominal) ∧ has_subtype (r, General) ∧ has_position(r, Backward) ∧ 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)]]

Another example presents a case when a coreferent of the pronoun "man" (the literal English translation "for me") is in the following sentence:

LT: Pastebėtina, kad prabangaus nekilnojamojo turto mokesčio surinkimo planas 2013 metams buvo 17 mln. litų, nepaisant to, kad 2012 m. šio mokesčio sumokėta mažiau nei 4 mln. litų (2013 m. surinkta beveik 5 mln. litų). GPM surinkimą labiausiai lėmė

minimalaus mėnesinio atlyginimo (MMA) padidinimas: „Kiek **man** [pronoun] teko analizuoti, padidinus MMA tik nedidelė dalis **Lietuvos** [location noun] įmonių sumažino etatą ar atleido darbuotojus, o tai lėmė nemažą papildomą indėlį į valstybės biudžetą" teigė Ž. **Mauricas** [person NP].

PL: Należy zauważyć, że plan poboru podatku od nieruchomości luksusowych na 2013 r. wyniósł 17 mln. lit, mimo że w 2012 roku za ten podatek zapłacono mniej niż 4 miliony lity (w 2013 r. zebrano prawie 5 milionów litów). Wzrost PDoOF wynikał głównie ze wzrostu minimalnego wynagrodzenia miesięcznego (MWM): „ Ile **ja** [pronoun] miałem przeanalizować, tylko niewielka część firm na **Litwie** [location noun] zmniejszyła swoją pozycję lub zwolniła swoich pracowników, co spowodowało znaczną dodatkową składkę do budżetu państwa", powiedział **Ž. Mauricas** [person NP].

RU: Следует отметить, что план сбора налога на элитную недвижимость на 2013 год составил 17 млн. лит, несмотря на то что в 2012 г. этого налога уплачено менше чем 4 млн. лит (почти 5 млн. литов было собрано в 2013 г.). Сбор ПН в основном был обусловлен увеличением минимальной месячной заработной платы (ММЗП): «Сколько **мне** [pronoun] приходилось анализировать, увеличев зарплату только небольшая часть компаний в **Литве** [location noun] сократили должность или уволили сотрудников, и это привело к значительному дополнительному вкладу в государственный бюджет», - сказал **Ž. Mauricas** [person NP].

EN: It is noteworthy that the real estate tax collection plan for 2013 was 17 million. Litas, even though in 2012 less than 4 million were collected in this tax. Litas (in 2013, nearly 5 million litas were collected). The collection of GPM was mainly due to an increase in the minimum monthly salary (MMA): "As far as **I** [pronoun] had analysed, only a small part of **Lithuanian** [location noun] companies have reduced their posts or dismissed employees due to increased MMA, which has led to a significant additional contribution to the state budget," said Ž. **Mauritius** [person NP].

In this case, the algorithm repeats the same steps as in the previous example. It does not find any named entities moving backwards; therefore, it moves back to our pronoun and proceeds forward. The first entity it finds is "Lietuvos", which means a location. The algorithm continues moving forward until it locates the "Ž. Mauricas" entity, which is recognized as a person. Since the pronoun "man" is ambiguous in gender (it can refer to both female and male persons), the pronoun and the NP are compared only in number. Both are singular; therefore, the algorithm picks "Ž. Mauricas" as a postecedent of the coreferring object "man". Conditions for the existence of such reference formally could be defined as two alternatives. The first

one describes the conditions for the reference existing in same sentence ***s1*** after a pronoun was mentioned:

*For each text's **t** sentence **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 in the same sentence **s1**, is expressed by lexeme l, and has gender **g**, number **n,** start position **sp2** and is of length **ln2**, and is after pronoun **p** (**sp2** is higher than **sp1**), but closer to pronoun **p** than possible named entities **e2** and **e3** (**sp2** higher than **sp3** and **sp4**), there exists 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**, (3.8).*

$$\begin{aligned}&\forall t, s1, p, l, e1, g, n, sp1, ln1, sp2, ln2.[\text{Text}(t) \wedge \text{Sentence}(s1) \wedge \text{consists\_of}(t, s1) \wedge \text{Pronoun}(p) \wedge \text{contains}(s1, p) \wedge \neg\text{has\_type}(p, \text{Demonstrative}) \wedge \text{has\_gender}(p, g) \wedge \text{has\_number}(p, n) \wedge \text{has\_start\_position}(p, sp1) \wedge \text{has\_length}(p, ln1) \wedge \text{Person\_NE}(e1) \wedge \text{includes}(s1, e1) \wedge \text{Lexeme}(l) \wedge \text{expressed\_by}(e1, l) \wedge \text{has\_gender}(e1, g) \wedge \text{has\_number}(e1, n) \wedge \text{has\_start\_position}(e1, sp2) \wedge \text{has\_length}(e1, ln2) \wedge sp1<sp2 \wedge \neg(\exists e2, e3, sp3, sp4.\ (e1\neq e2 \wedge e1\neq e3 \wedge e2\neq e3 \wedge \text{Person\_NE}(e2) \wedge \text{includes}(s1, e2) \wedge \text{has\_start\_position}(e2, sp3) \wedge \text{Person\_NE}(e3) \wedge \text{includes}(s1, e3) \wedge \text{has\_start\_position}(e3, sp4) \wedge sp2<sp3 \wedge sp4<sp2))\\ &\rightarrow \exists!r\ \exists!m.\ [\text{Coreference}(r) \wedge \text{resolved\_in}(r, t) \wedge \text{has\_type}\ (r, \text{Pronominal}) \wedge \text{has\_subtype}\ (r, \text{General}) \wedge \text{has\_position}(r, \text{Forward}) \wedge \text{has\_group}(r, \text{Single}) \wedge \text{has\_start\_position}(r, sp1) \wedge \text{has\_length}(r, ln1) \wedge \text{fits}(r, p) \wedge \text{Mention}(m) \wedge \text{refers\_to}(r, t) \wedge \text{has\_start\_position}(m, sp2) \wedge \text{has\_length}(m, ln2) \wedge \text{fits}(m, e1) \wedge \text{fits}(m, l)]]\end{aligned} \tag{3.8}$$

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 **s1, 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**), there exists 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**, (3.9).*

$$\forall t, s1, s2, p, l, e1, g, n, sp1, ln1, sp2, ln2.[Text(t) \wedge Sentence(s1) \wedge Sentence(s2) \wedge consists\_of(t, s1) \wedge consists\_of(t, s2) \wedge follows(s2, s1) \wedge Pronoun(p) \wedge contains(s1, p) \wedge \neg has\_type(p, Demonstrative) \wedge has\_gender(p, g) \wedge has\_number(p, n) \wedge has\_start\_position(p, sp1) \wedge has\_length(p, ln1) \wedge Person\_NE(e1) \wedge includes(s2, e1) \wedge Lexeme(l) \wedge expressed\_by(e1, l) \wedge has\_gender(e1, g) \wedge has\_number(e1, n) \wedge has\_start\_position(e1, sp2) \wedge has\_length(e1, ln2) \wedge \neg(\exists e2, e3, sp3, sp4. (e1 \neq e2 \wedge e1 \neq e3 \wedge e2 \neq e3 \wedge Person\_NE(e2) \wedge includes(s2, e2) \wedge has\_start\_position(e2, sp3) \wedge Person\_NE(e3) \wedge includes(s2, e3) \wedge has\_start\_position(e3, sp4) \wedge sp2 < sp3 \wedge sp4 < sp2)) \rightarrow \exists!r \exists!m. [Coreference(r) \wedge resolved\_in(r, t) \wedge has\_type(r, Pronominal) \wedge has\_subtype(r, General) \wedge has\_position(r, Forward) \wedge has\_group(r, Single) \wedge has\_start\_position(r, sp1) \wedge has\_length(r, ln1) \wedge fits(r, p) \wedge Mention(m) \wedge refers\_to(r, t) \wedge has\_start\_position(m, sp2) \wedge has\_length(m, ln2) \wedge fits(m, e1) \wedge fits(m, l)]] \quad (3.9)$$

The algorithm ignores demonstrative pronouns because they are often used to refer to entities that are not present in the written text and due to that are exophoric, while the proposed approach attempts to solve only endophoric coreferences. Such pronouns are also sometimes pleonastic – they do not carry any additional semantic information and do not refer to any NP. They are used mostly for syntactic reasons and due to that are usually ignored in CR.

Empirically it was determined that two sentences backwards and one sentence forward have produced the best results. This is similar to what one of the Russian language solutions have reported [143] while calculating the optimal word distance. They determined that the optimal distance is 25 words, while average sentences of publicistic style (the focus of SSFLL) in the Lithuanian language is 15.7 words [144][145]. Calculating the distance with sentences over words (tokens) was chosen because:

- Not all tokens (punctuation marks, quotations) can be treated equally when calculating distance.
- Different authors can have more or less verbose sentences. Therefore, calculation of distance by tokens would have to be different for each case.
- The Lithuanian language has a free word order, meaning that the same sentence can be written in many different ways. Thus, the distance calculated by the number of tokens can be unreliable.

In total, three sentences around coreferring objects are covered. These parameters might vary in different languages or different types of texts and should be adjusted accordingly.

Sometimes paragraphs are also used to determine the width of the analysed text. Referents tend to refer to antecedents that are presented in the same paragraph. But SSFLL focuses on internet news articles and in such texts paragraphs are used liberally. Sometimes each sentence is written in a different paragraph, while in other cases, the entire text is presented in one paragraph. Websites also sometimes have a

complex design that makes it difficult for a web crawler to correctly identify where one paragraph ends and another begins. Consequently, paragraphs are not used for distance calculation.

Most of the named entities that are recognized by NER as persons are singular, but sometimes families are mentioned, e.g., Paulauskai, Zuokai. Due to such cases, it is important to check for agreement in number between nouns (or NPs) and pronouns.

Formalization of these rules using a pseudocode is provided in Formula (3.10).

```
Input: pronoun that has to be resolved
Output: coreference annotation

List coreferenceAnnotations

parentSentence = pronoun.getParentSentence
namedEntities = parentSentence.getNamedEntities
For entity in namedEntities.reverse
  //Reverse is used because we move backwards at first
  if(entity.startingPosition < pronoun.startingPosition)
    if (entity.Gender = pronoun.Gender && entity.Number =
pronoun.Number)
      coreferenceAnnotations.Add(entity, pronoun, "pras")
      solved = true
      break
    Endif
  Endif
Endfor
if(!solved)
  For entity in namedEntities
    if(entity.startingPosition > pronoun.startingPosition)
      if (entity.Gender = pronoun.Gender && entity.Number =
pronoun.Number)
        coreferenceAnnotations.Add(entity, pronoun, "prps")
        solved = true
        break
      Endif
    Endif
  Endfor
Endif
if(!solved)
  While i=0; i < 2, i++
    parentSentence = parentSentence.getPreviousSentence
    namedEntities = parentSentence.getNamedEntities
    For entity in namedEntities.reverse
      if (entity.Gender = pronoun.Gender && entity.Number =
pronoun.Number)
        coreferenceAnnotations.Add(entity, pronoun, "pras")
        solved = true
        break
      Endif
```

(3.10)

```
    Endfor
  Endwhile
Endif

if(!solved)
  parentSentence = pronoun.getParentSentence
  While i=0; i < 1, i++
    parentSentence = parentSentence.getFollowingSentence
    namedEntities = parentSentence.getNamedEntities
    For entity in namedEntities
      if (entity.Gender = pronoun.Gender && entity.Number =
pronoun.Number)
        coreferenceAnnotations.Add(entity, pronoun, "prps")
        solved = true
        break
      Endif
    Endfor
  Endwhile
Endif
```

A3. PRA resolution. This algorithm is based mostly on exact (or partial) string matches and a number of rules for acronyms. Once the first named entity that can be matched with the initial named entity is found, then the algorithm does not look for further named entities in order to keep annotations simple: B → A, C → B and D → C. This allows to form coreference chains linking all mentions of same entity in a text that can be later re-used for semantic analysis. For example:

LT: **Tomaitis** [named entity] pavėlavo į darbą. Po pietų pertraukas direktorius pasikvietė **Tomaitį** [named entity] pokalbiui.

PL: **Tomaitis** [named entity] spóźnił się do pracy. Po przerwie obiadowej dyrektor zaprosił **Tomaitis** [named entity] na rozmowę.

RU: **Томайтис** [named entity] опоздал на работу. После обеднаго перерыва директор пригласил **Томайтиса** [named entity] на беседу.

EN: **Tomaitis** [named entity] was late to work. After lunch, the director invited **Tomaitis** [named entity] for a conversation.

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 could be 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 lexeme **l1** that has lemma **l** and for each same text’s **t** sentence **s2** that includes named entity **e2**, that has a start position **sp1** and is of length **ln1**, which is expressed by lexeme **l2** that has lemma **l**, there exists only one coreference relation **r**, which is resolved in text **t**, is of “Nominal” type, “Repetition” subtype, “Irrelevant” position and “Single” group between the noun **n1** and the noun **n2**, its referent starts at position **sp1** and has length **ln1**, and*

*which fits only one noun **n1** and refers to only one mention **m**, which starts at position **sp2**, has length **ln2**, and fits only one noun **n2**, Formula (3.11).*

$$\forall t, s1, s2, e1, e2, sp1, ln1, sp2, ln2.[\text{Text}(t) \wedge \text{Sentence}(s1) \wedge \text{Sentence}(s2) \wedge \text{consists\_of}(t, s1) \wedge \text{consists\_of}(t, s2) \wedge \text{Named\_Entity}(e1) \wedge \text{includes}(s1, e1) \wedge \text{has\_start\_position}(e1, sp1) \wedge \text{has\_length}(e1, ln1) \wedge \text{Named\_Entity}(e2) \wedge \text{includes}(s2, e2) \wedge \text{has\_start\_position}(e2, sp2) \wedge \text{has\_length}(e2, ln2) \wedge e1 \neq e2 \wedge (\exists l1\ \exists l2\ \exists l.(\text{Lexeme}(l1) \wedge \text{Lexeme}(l2) \wedge \text{expressed\_by}(e1, l1) \wedge \text{expressed\_by}(e2, l2) \wedge \text{has\_lemma}(l1, l) \wedge \text{has\_lemma}(l2, l)) \rightarrow \exists!r\ \exists!m.[\text{Coreference}(r) \wedge \text{resolved\_in}(r, t) \wedge \text{has\_type}(r, \text{Nominal}) \wedge \text{has\_subtype}(r, \text{Repetition}) \wedge \text{has\_position}(r, \text{Irrelevant}) \wedge \text{has\_group}(r, \text{Single}) \wedge \text{has\_start\_position}(r, sp1) \wedge \text{has\_length}(r, ln1) \wedge \text{fits}(r, l1) \wedge \text{fits}(r, e1) \wedge \text{Mention}(m) \wedge \text{refers\_to}(r, t) \wedge \text{has\_start\_position}(m, sp2) \wedge \text{has\_length}(m, ln2) \wedge \text{fits}(m, l2) \wedge \text{fits}(m, e2)]] \quad (3.11)$$

This is further formalized using a pseudo code in Formula (3.12).

```
Input: named entity that has to be resolved
Output: coreference annotation

List coreferenceAnnotations
List allNamedEntities

cleanLemma = namedEntity.getCleanLemma
For candidateNE in allNamedEntities
  if(namedEntity.getType = candidateNE.getType)
    candidateCleanLemma = candidateNE.getCleanLemma
    if(repetition(cleanLemma, candidateCleanLemma))
      coreferenceAnnotations.Add(candidateNE, namedEntity, "dtis")
      break
    Else
      solved = False
      if(namedEntity.getType = person)
        solved = PersonAcronym(cleanLemma, candidateCleanLemma)
      ElseIf(namedEntity.getType = organization)
        solved = OrganizationAcronym(cleanLemma, candidateCleanLemma)
      ElseIf(namedEntity.getType = location)
        solved = LocationAcronym(cleanLemma, candidateCleanLemma)
      Endif
      if(solved)
        coreferenceAnnotations.Add(entity, pronoun, "dais")
      Endif
    Endif
  Endif
Endfor
```

(3.12)

Acronym rules vary depending on the type of named entity (currently persons, locations, and organizations are covered). The especially challenging part is linking persons with foreign names, since their names are usually Lithuanized. For example, the former French president can be referred to as Hollande, Hollandas, Holandas, Hollande'as. For this purpose, specific language-dependant rules are required to link such entities efficiently.

A4. HHS resolution. This algorithm is based on the classification of professions. It attempts to resolve the use of synonyms and hypernyms/hyponyms. For example, in the text, the same person can be referred to as a "writer" and as a "novelist". "Writer" is a broader term and as such would be a hypernym, while "novelist" is more specific and would be a hyponym. An example of hypernym-hyponym:

LT: **Žurnalistas** [noun referring to profession] parašė naują straipsnį apie nacionalinę politiką. **Autorius** [noun referring to profession] buvo iškarto sukritikuotas valdančiosios partijos remėjų.

PL: **Dziennikarz** [noun referring to profession] napisał nowy artykuł na temat polityki krajowej. **Autor** [noun referring to profession] został natychmiast skrytykowany przez zwolenników partii rządzącej.

RU: **Журналист** [noun referring to profession] написал новую статью о национальной политике. **Автор** [noun referring to profession] был немедленно раскритикован сторонниками правящей партии.

EN: **Journalist** [noun referring to profession] wrote a new article today about national politics. The **author** [noun referring to profession] was immediately criticized by the supporters of the ruling party.

The algorithm determines that "Žurnalistas" in the classification of professions is a hyponym of "Autorius", they also agree in gender and number, therefore, the algorithm adds their pair to annotations. Conditions for the existence of such reference formally could be defined as follows:

For each text's ***t*** *sentence* ***s1*** *that has profession* ***p1****, which is either broader or narrower than profession* ***p2****, name* ***v1*** *expressing noun* ***n1****, which has gender* ***g****, number* ***m****, start position* ***sp1*** *and is of length* ***ln1****, and for each same text's* ***t*** *sentence* ***s2*** *that has profession* ***p2****, which is either broader or narrower than profession* ***p1****, name* ***v2*** *expressing noun* ***n2****, which has gender* ***g****, number* ***m****, start position* ***sp2*** *and is of length* ***ln2,*** *there exists only one coreference relation* ***r****, which is resolved in text* ***t****, is of "Nominal" type, "Hypernym_hyponym" subtype, "Irrelevant" position and "Single" group between the noun* ***n1*** *and the noun* ***n2****, its referent starts at position* ***sp1*** *and has length* ***ln1****, and which fits only one noun* ***n1*** *and refers to only one mention* ***m****, which starts at position* ***sp2****, has length* ***ln2****, and fits only one noun* ***n2****, Formula (3.13).*

∀t, s1, s2, n1, n2, sp1, ln1, sp2, ln2, v1, v2, p1, p2.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ Noun(n1) ∧ contains(s, n1) ∧ has_start_position(n1, sp1) ∧ has_length(n1, ln1) ∧ Noun(n2) ∧ contains(s2, n2) ∧ has_start_position(n2, sp2) ∧ has_length(n2, ln2) ∧ n1≠n2 ∧ Profession(p1) ∧ Profession(p2) ∧ p1≠p2 ∧ Profession_Name(v1) ∧ Profession_Name(v2) ∧ express(n1, v1) ∧ express(n2, v2) ∧ describes(v1, p1) ∧ describes(v2, p2) ∧ (broadens(p2, p1) ∨ broadens(p1, p2)) ∧ has_gender(n1, g) ∧ has_gender(n2, g) ∧ has_number(n1, n) ∧ has_number(n2, n)
→ ∃!r ∃!m. [Coreference® ∧ resolved_in(r, t) ∧ has_type (r, Nominal) ∧ has_subtype (r, Hypernym_Hyponym) ∧ has_position(r, Irrelevant) ∧ has_group(r, Single) ∧ has_start_position(r, sp1) ∧ has_length(r, ln1) ∧ fits(r, n1) ∧ Mention(m) ∧ refers_to(r, t) ∧ has_start_position(m, sp2) ∧ has_length(m, ln2) ∧ fits(m, n2)]] (3.13)

An example of synonym:

LT: J. Jonaitis nuo šiol yra šios įmonės **vadovas** [noun referring to profession]. Deja, darbuotojai nemėgsta savo naujojo **viršininko** [noun referring to profession].

PL: J. Jonaitis jest teraz **kierownikiem** [noun referring to profession] tej firmy. Niestety pracownicy nie lubią swojego nowego **szefa** [noun referring to profession].

RU: Я. Йонайтис от сих пор является **менеджером** [noun referring to profession] этой компании. К сожалению, сотрудники не любят своего нового **началника** [noun referring to profession].

EN: From now J. Jonaitis is the **head** [noun referring to profession] of this firm. Unfortunately, workers do not like their new **boss** [noun referring to profession].

Both “vadovas” and “viršininko” are synonymous therefore the condition for the existence of such reference formally could be defined as follows:

*For each text’s **t** sentence **s1** that has a profession’s **p** name **v1**, which is expressed by noun **n1** that has gender **g**, number **m**, start position **sp1** and is of length **ln1**, and for each same text’s **t** sentence **s2** that has same profession’s **p** name **v2** expressed by noun **n2** that has gender **g**, number **m**, start position **sp2** and is of length **ln2**, there exists 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 **n1** and the noun **n2**, its referent starts at position **sp1** and has length **ln1**, and which fits only one noun **n1** and refers to only one mention **m**, which starts at position **sp2**, has length **ln2**, and fits only one noun **n2**, Formula (3.14).*

∀t, s1, s2, n1, n2, sp1, ln1, sp2, ln2, v1, v2, p, g, n.[Text(t) ∧ Sentence(s1) ∧ Sentence(s2) ∧ consists_of(t, s1) ∧ consists_of(t, s2) ∧ Noun(n1) ∧ contains(s1, n1) ∧ has_start_position(n1, sp1) ∧ has_length(n1, ln1) ∧ Noun(n2) ∧ contains(s2, n2) ∧ has_start_position(n2, sp2) ∧ has_length(n2, ln2) ∧ n1≠n2 ∧ Profession_name(v1) ∧ Profession_name(v2) ∧ Profession(p) ∧ express(n1, v1) ∧ express(n2, v2) ∧ describes(v1, p) ∧ describes(v2, p) ∧ has_gender(n1, g) ∧ has_gender(n2, g) ∧ has_number(n1, n) ∧ has_number(n2, n) ∧ n1≠n2 → ∃!r ∃!m. [Coreference® ∧ resolved_in(r, t) ∧ has_type (r, Nominal) ∧ has_subtype(r, Synonym) ∧ has_position(r, Irrelevant) ∧ has_group(r, Single) ∧ has_start_position(r, sp1) ∧ has_length(r, ln1) ∧ fits(r, n1) ∧ Mention(m) ∧ refers_to(r, t) ∧ has_start_position(m, sp2) ∧ has_length(m, ln2) ∧ fits(m, n2)]] (3.14)

These rules are further formalized using a pseudocode in Formula (3.15).

```
Input: noun that has to be resolved
Output: coreference annotation

List coreferenceAnnotations

validSynonyms = synonymDictionary.getSynonyms(noun)
validHypernyms = synonymDictionary.getHypernyms(noun)
startingPosition = noun.getStartingPosition
gender = noun.getGender
number = noun.getNumber
parentSentence = noun.getParentSentence
solved = False
i = 0
while !solved || i > 3
  for candidateNoun in parentSentence.getNouns
    candidateGender = candidateNoun.getGender
    candidateNumber = candidateNoun.getNumber
    if(validSynonyms.contains(candidateNoun) && gender =
candidateGender && number = candidateNumber)
      coreferenceAnnotationsAdd(candidateNoun, noun, "gsis")
      solved = True
      break
    ElseIf(validHypernyms.contains(candidateNoun) && gender =
candidateGender && number = candidateNumber)
      if(candidateNoun.getStartingPosition < startingPosition)
        coreferenceAnnotationsAdd(candidateNoun, noun, "ghas")
      Else
        coreferenceAnnotationsAdd(candidateNoun, noun, "ghps")
      Endif
      solved = True
      break
    Endif
```

(3.15)

Endfor
if(i = 3)
parentSentence = noun.getParentSentence.getFollowingSentence
Else
parentSentence = parentSentence.getPreviousSentence
Endif
i++
Endwhile

A5. Feature resolution. This algorithm attempts to resolve only those cases when a person is being referred to by his/her public post (feature) that he/she holds, other types of features are not currently resolved. For example:

LT: Koks yra **S**. **Skvernelis** [person NP]? Pakalbėkime apie naujojo **premjero** [noun referring to held position] vaikystė, šeima ir karjera.

PL: Jakim jest **S. Skvernel** [person NP]? Porozmawiajmy o dzieciństwie, rodzinie ® karierze nowego **premiera** [noun referring to held position].

RU: Каким является **С. Сквернель** [person NP]? Давайте поговорим о детстве, семье и карьере нового **премьера** [noun referring to held position].

EN: What **S**. **Skvernelis** [person NP] is like? Let's talk about the new prime **minister's** [noun referring to held position] childhood, family, and career.

In this example, the noun "premjero" is selected, the algorithm moves backwards till it reaches "S. Skvernelis" and checks knowledge base if at the time of the publication of the article he has held the position of the prime minister. Since he did, the algorithm checks if "S. Skvernelis" and "premjero" agree in gender and number. They do, therefore their pair is added to annotation as a feature reference. A condition for the existence of such reference formally could be defined as follows:

*For each text's **t** sentence **s1** that has known person **k**, who during publication date **d** had a 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**, there exists 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**, Formula (3.16).*

∀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® ⋀ includes(s1, e) ⋀ Noun(n) ⋀ contains(s2, n) ⋀ Known_Person(k) ⋀ mentioned_as(k, e) ⋀ Position_held(h) ⋀ holds(k, h) ⋀ has_lemma(h, l) ⋀ has_lemma(n, l) ⋀ has_publication_date(t, d) ⋀ has_from_date(h, fd) ⋀ has_to_date(h, td) ⋀ fd≤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® ⋀ resolved_in(r, t) ⋀ has_type (r, Nominal) ⋀ has_subtype (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)]] (3.16)

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

LT: Koks yra naujasis **premjeras** [noun referring to held position]? **S. Skvernelio** [person NP] vaikystė, šeima ir karjera.

PL: Jakim jest nowy **premier** [noun referring to held position]? Dzieciństwo, rodzina ® kariera S. **Skvernelis** [person NP]..

RU: Каким является новый **премьер** [noun referring to held position]? Детство, семья и карьера **С. Сквернелиса** [person NP].

EN: What new prime **minister** [noun referring to held position] is like? The childhood, family and career of **S. Skvernelis** [person NP].

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

*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 ealier 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**, there exists 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**, Formula (3.17).*

∀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® ∧ includes(s1, e) ∧ Noun(n) ∧ contains(s2, n) ∧ Known_Person(k) ∧ mentioned_as(k, e) ∧ Position_held(h) ∧ holds(k, h) ∧ has_lemma(h, l) ∧ has_lemma(n, l) ∧ has_publication_date(t, d) ∧ has_from_date(h, fd) ∧ has_to_date(h, td) ∧ fd≤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® ∧ resolved_in(r, t) ∧ has_type (r, Nominal) ∧ has_subtype (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)]] (3.17)

These rules are further formalized using a pseudocode in Formula (3.18) formula.

```
Input: noun that has to be resolved, text in which noun is present
Output: coreference annotation

List coreferenceAnnotations
List allPublicPersons

if(noun.getNumber = singular)
  publicationDate = text.getPublicationDate
  For person in allPublicPersons
    for heldPosition in person.getHeldPositions
      startingDate = heldPosition.getStartingDate
      if(heldPosition = noun && person.getGender = noun.getGender &&
startingDate < publicationDate)
        if(person.getFirstMention.getStartingPosition <
noun.getStartingPosition)
          coreferenceAnnotationsAdd(candidateNE, namedEntity, "gfas")
        Else
          coreferenceAnnotationsAdd(candidateNE, namedEntity, "gfps")
        Endif
        break
      Endif
    Endfor
  Endfor
Endif
```
(3.18)

3.5 Dominant mentions

A list of dominant mentions is provided alongside coreference annotations as a separate annotation layer. As detailed in Section 2.2, dominant mentions would be useful for future research focusing on CR between different texts. It is also used in the

evaluation model of the proposed annotation scheme. The main principles of how to determine dominant mentions have been explained in Section 2.2. Here, an example will be provided for this specific implementation.

Coreference annotations are stored in JSON format. Annotation that would be created after CR and before dominant mentions are added can be seen in Table 3.3.

Table 3.3 Coreference annotation example without dominant mentions

{"coreferences":[{"Mentions":[[256,18]],"Referant":[378,3],"Type":"pras"},{"Mentions":[[1308,5]],"Referant":[1315,5],"Type":"peas"},{"Mentions":[[1143,15]],"Referant":[1346,2],"Type":"pras"},{"Mentions":[[1350,10]],"Referant":[1403,2],"Type":"pras"},{"Mentions":[[1510,14]],"Referant":[1526,4],"Type":"peas"},{"Mentions":[[2131,5]],"Referant":[2138,6],"Type":"peas"},{"Mentions":[[113,7]],"Referant":[162,8],"Type":"dtis"},{"Mentions":[[84,15]],"Referant":[299,2],"Type":"dbis"},{"Mentions":[[113,7]],"Referant":[353,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[383,8],"Type":"dtis"},{"Mentions":[[84,15]],"Referant":[426,2],"Type":"dtis"},{"Mentions":[[84,15]],"Referant":[523,2],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[564,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[606,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[654,8],"Type":"dtis"},{"Mentions":[[101,2]],"Referant":[730,2],"Type":"dtis"},{"Mentions":[[256,18]],"Referant":[743,15],"Type":"dais"},{"Mentions":[[113,7]],"Referant":[838,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[923,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1015,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1097,8],"Type":"dtis"},{"Mentions":[[256,18]],"Referant":[1143,15],"Type":"dais"},{"Mentions":[[113,7]],"Referant":[1253,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1280,8],"Type":"dtis"},{"Mentions":[[84,15]],"Referant":[1375,2],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1463,7],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1495,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1550,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1576,8],"Type":"dtis"},{"Mentions":[[113,7]],"Referant":[1692,8],"Type":"dtis"},{"Mentions":[[101,2]],"Referant":[1770,2],"Type":"dtis"},{"Mentions":[[0,7]],"Referant":[1914,7],"Type":"dtis"},{"Mentions":[[11,7]],"Referant":[1923,7],"Type":"dtis"},{"Mentions":[[11,7]],"Referant":[1971,8],"Type":"dtis"},{"Mentions":[[11,7]],"Referant":[2206,7],"Type":"dtis"},{"Mentions":[[0,7]],"Referant":[2217,7],"Type":"dtis"},{"Mentions":[[1350,10]],"Referant":[1473,2],"Type":"pras"},{"Mentions":[[362,8]],"Referant":[392,6],"Type":"gsis"},{"Mentions":[[256,18]],"Referant":[501,7],"Type":"gfas"},{"Mentions":[[584,16]],"Referant":[663,11],"Type":"gfas"},{"Mentions":[[179,21]],"Referant":[584,16],"Type":"dbis"},{"Mentions":[[743,15]],"Referant":[766,11],"Type":"gfas"},{"Mentions":[[392,6]],"Referant":[847,8],"Type":"gsis"},{"Mentions":[[847,8]],"Referant":[1106,11],"Type":"gmis"},{"Mentions":[[743,15]],"Referant":[1131,10],"Type":"gfas"},{"Mentions":[[1106,11]],"Referant":[1559,7],"Type":"gmas"},{"Mentions":[[1143,15]],"Referant":[1350,10],"Type":"gfas"},{"Mentions":[],"Referant":[1350,10],"Type":"ghps"},{"Mentions":[[141,19]],"Referant":[429,21],"Type":"dtis"},{"Mentions":[[1350,10]],"Referant":[2278,6],"Type":"gfas"},{"Mentions":[[141,19]],"Referant":[1197,21],"Type":"dtis"},{"Mentions":[[362,8]],"Referant":[1289,8],"Type":"gtis"},{"Mentions":[[362,8]],"Referant":[1701,9],"Type":"gtis"},{"Mentions":[[84,15]],"Referant":[101,2],"Type":"dbis"}]}

The structure of this annotation has been detailed in the 2.4 section. The algorithm that extracts dominant mentions from this structure can be defined in the following steps:

1. The algorithm takes the next element in “coreferences” array. If no objects are left to take then the algorithm moves to step 6.
2. If the annotation element has a type that starts with “a” (adverbial) or with “ey” (ellipsis of the same type) or “ev” (VP ellipsis) then that element is ignored and the algorithm moves back to step 1. Otherwise, it moves to step 3.
3. Each mention in “Mentions” array is added to the list of dominant mentions and “Referent”, alongside “Type” value is added as the referent for it. The

algorithm moves to step 4.

4. If added dominant mention *m1* already exists as a referent to another dominant mention *m2* then *m1* is removed from the list of dominant mentions and its referents, are added to *m2*. The algorithm moves to step 5.
5. If added referent already exists as dominant mention *m3* then it is removed from the list of dominant mentions and all of its referents are added to dominant mention *m1*. The algorithm moves back to step 1.
6. The algorithm adds a list of dominant mentions to the annotation and ends its work.

Coreference annotation with dominant mentions added can be seen in Table 3.4

Table 3.4 Coreference annotation example with dominant mentions

```
{"coreferences":[…],
"dominants": [[256,18], [1308,5], [1510,14], [2131,5], [113,7], [84,15], [0,7], [11,7], [362,8],
[179,21], [141,19]]}
```

3.6 Additional knowledge bases

The proposed approach uses semantic knowledge to resolve certain coreference expressions. In this section, the source of this semantic information is covered.

Features of the public persons

Wikipedia hosts lots of valuable information about public persons that can be used for NLP purposes, like their current and former occupations, information that is used in the proposed CR. DBpedia [146] allows to query and retrieve that information in a structured manner. An example of SPARQL is provided in Table 3.5.

Table 3.5 SPARQL example for DBPedia

```
SELECT ?politician WHERE {
?politician rdf:type <http://dbpedia.org/class/yago/WikicatLithuanianPoliticians> . }
```

This simple SPARQL query lists all public persons in Wikipedia that are categorized as “LithuanianPoliticians”. From that point, the query can be extended to retrieve their occupation and other stored information.

At the time, DBpedia is a copy of Wikipedia’s 2016-10 version, therefore, information is not up to date. Also, since it focuses on the English language, it does not cover many politicians, or other public persons, from Lithuania. Due to these reasons, a significant amount of information has to be added and updated manually. But at the same time, DBpedia provides a useful blueprint for semantic information storage and usage for various NLP tasks.

Synonyms

For the resolution of expressions based on the synonym relationship, “Sinonimų žodynas” by Antanas Lieberis was used. It is available publicly in the RDF format via [147].

Hypernym and hyponym relationship

To solve expressions based on this relationship, the hierarchy between different

names of wider and narrower profession names had to be constructed. For this purpose, a separate system, “Semantinio vaidmenų žodyno sudarymo informacinė sistema” was used [148].

3.7 Conclusions of chapter 3

1. Sections 3.1 and 3.6 detail the context in which the developed CR approach was implemented.
2. The main CR algorithm that connects five smaller algorithms was presented in Section 3.2. The decision table was used to determine which smaller algorithm should be used depending on the present conditions.
3. The main concepts that were used for coreference resolution were defined in Section 3.3. These concepts and relationships between them were later used in first-order logic formulas.
4. Section 3.4 explains and formalizes each algorithm using first-order logic and pseudocode.
5. Section 3.5 provided information on the specific implementation of the dominant mentions.

4 EVALUATION OF COREFERENCE CORPUS AND RESOLUTION METHODS

In order to evaluate the created resources and developed CR approach multiple experiments were carried out.

First, Section 4.1 tests the suitability of annotation scheme via the inter-annotator agreement using the created LCC corpus. In Section 4.2, the developed CR approach was evaluated using the proposed evaluation strategy and four other evaluation strategies; results of this experiment have also been published in [136]. Their results were compared to highlight certain advantages provided by the proposed linguistically aware evaluation strategy. The third experiment was carried out to determine the impact that the output of our developed CR approach can have on the results of the semantic annotator. Results of this experiment are provided in Section 4.3. Finally, Section 4.4 summarizes these experiments.

4.1 Evaluation of Lithuanian Coreference Corpus

Inter-annotator agreement experiments are usually performed for coreference annotations since different coders, that manually create annotations, can have different interpretations of the same text [149] [150]. Therefore, experiments are run with multiple coders to identify the problematic cases and evaluate the overall reliability of the corpus.

As with coreference annotation evaluation, there are multiple different metrics without an agreed upon standard [44]. It was decided to employ a percent-based agreement due to its simplicity. The main flaw of this approach is that it does not consider that certain agreement percentages can be reached simply by chance.

For this task, three independent annotators with different backgrounds were used. *Annotator A* has a linguistic background, *Annotator B* has an informatics background and *Annotator C* has a mathematics background. Each of them took annotations created by the algorithm presented in Section 3.2, corrected mistakes and added missing annotations according to their opinion. The annotation process was carried out by following the proposed scheme and guidelines in Sections 2.1-2.3. For manual annotation, separate tools were developed during the SCAF project[2].

The entire corpus was annotated by each annotator. Their annotations were compared against each other using the evaluation model proposed in Section 2.5. The only modification was that mistakes were not classified and each coreference relationship either got 1 or 0 scores. Each annotator was compared against each other and all three together, the results of this evaluation can be seen in Table 4.1.

When agreement between all three annotators was not achieved, the correct annotation was selected by voting. In rare cases when all three annotators disagreed, discussion was carried out to determine the correct annotation.

[2] Smart Cloud Application Framework (SCAF)“, Investment Action Programme measure "Intellect. General Science - Business Projects", coordinator: UAB „Sekasoft“.Partner – Kaunas University of Technology. (2017–2018)

Table 4.1 Inter-annotator agreement results

Evaluated annotators	Agreement percentage
Annotator A and *Annotator B*	94.3%
Annotator A and *Annotator C*	87.8%
Annotator B and *Annotator C*	86.1%
Annotator A and *Annotator B* and *Annotator C*	80.7%

Overall *Annotator A* had a higher agreement with *B* and *C* than they had with each other, but it was not the case for all texts. Therefore, the basis for updated LCC coreference annotations were selected for each text individually. Mistakes and omissions were fixed with few cases getting subtype indicating ambiguity.

Most common issues were the following:

1. Metonymic, hypernym-hyponym, and synonymous relationships can cause confusion. Vocabulary has to be consulted in some cases and there is also the question of the author's intent while writing a specific text fragment.
2. According to the proposed annotation scheme, a minimal marking approach has to be used for mention marking. Unfortunately, in some cases, interpretations vary of what is minimal marking that describes the entity sufficiently.
3. The marking of adverbial and ellipsis coreferences was not well defined in the earlier version of the annotation scheme [135]. Based on the feedback it was updated to the version that is currently presented in this work.
4. A similar situation was with order of markings based on dominance. Guidelines were updated accordingly.
5. Some coreferences were missed by one or two annotators. It is difficult to establish a pattern among those mistakes and it can probably be attributed to human error. At the same time, it is a good indication that manual annotations should always be carried out by multiple annotators that allow cross-checking their work.

4.2 Evaluation of coreference resolution approach for the Lithuanian Language

For the experiment, the SSFLL NLP pipeline was used that was detailed in Section 3.1. The same architecture is not required to implement this solution, but it requires lexical, morphological, and NE annotations that can be obtained from different NLP components. The proposed solution is not language or technology dependent, formalized rules can be implemented for any platform.

The purpose of the experiment was to evaluate our proposed algorithm against the corpus of Politics and Economy domains collected from Lithuanian Internet news sites in the environment of the SSFLL. The evaluation was made by analysing 100 articles that have been pre-annotated and are available in the LCC [138] corpus.

Cross-validation was not performed for two reasons. The main purpose of the cross-validation is to address the issue of overfitting. Overfitting happens when a predictive model is not able to generalize from training data to unseen data and cross-validation is usually used to better evaluate such models [151] [152]. Since our CR approach is rule-based, testing for overfitting is not as important. The second reason

is more pragmatic, the limited size of our corpus is not very suitable for performing cross-validation.

The proposed approach attempts to solve certain pronominal and nominal coreferences. The results of pronominal resolution are detailed in Table 4.2, the results for generic nominal resolution are in Table 4.3 and for definitive nominal resolution in Table 4.4. The results for adverbial and ellipsis coreferences are not detailed since the proposed approach does not attempt to solve them.

Type column indicates what type of coreferences (according to the proposed annotation scheme in Section 2.1) was attempted to solve. Only relative pronouns that refer to multiple antecedents are currently solved. Therefore, all other group (and ambiguous) references were put together since their further specification would not add anything. Columns correspond with six classes of annotations that were detailed in Section 2.5:

- *TP* – number of correctly resolved coreferences;
- *WT* – number of resolved coreferences with the wrong type specified;
- *WL* – number of resolved coreferences with the wrong dominant mention linked;
- *WTL* – number of resolved coreferences with the wrong type and dominant mention linked;
- *FN* – number of false negatives;
- *FP* – number of false positives;
- *S** – sum of *TP*, *WT*, *WL*, *WTL* and *FN*.

Table 4.2 Experiment results for pronominal coreference resolution

Type	TP	WT	WL	WTL	FN	FP	S*
ppas	103	19	12	14	83	20	231
ppps	9	4	15	6	7	2	41
pras	4	3	0	0	13	0	20
prps	0	0	0	0	2	0	2
poas	18	4	0	5	48	5	75
pops	1	0	0	0	3	2	4
peas	141	0	0	0	3	0	144
peag	13	0	0	0	6	0	19
Group references	0	0	0	0	14	0	14
Ambiguous	0	0	0	0	3	0	3
All	289	30	27	25	182	29	553

Based on the results of pronominal CR, the following observations can be made:

- Singular relative pronoun resolution (PEAS type) achieves high results due to having a well-defined usage structure.
- Plural relative pronoun (PEAG) usage is rarer, and their structure is not as well defined, which causes a higher number of errors.
- Personal pronouns used in quotations are often deictic, this causes the majority of false positive results.

- At the moment, for non-relative pronouns only named entities are considered as possible antecedents. This is a major reason for a high number of missed coreferences.
- Plural pronoun usage is problematic due to many variations possible that often ignore grammatical compatibility rules.

Table 4.3 Experiment results for generic nominal coreference resolution

Type	TP	WT	WL	WTL	FN	FP	S*
gais	0	0	0	0	54	0	54
gtis	0	0	0	0	109	0	109
gfas	107	0	15	0	62	13	184
gfps	3	0	4	0	7	3	14
ghas	4	2	2	6	47	1	61
gmis	0	0	0	0	34	0	34
gsis	9	2	0	3	34	6	48
Group references	0	0	0	0	31	0	31
Ambiguous	0	0	0	0	2	0	2
All	123	4	21	9	380	23	537

- Only feature resolution (GFAS and GFPS) produces encouraging results. Many other types of generic pronoun usage are either not covered or have very limited coverage.
- Linking the named entity to the position held considering the date of publication of the articles is limited considering that the article might be published today but write about things that happened 10 years ago. To solve such situations, tools that can identify the timeframe of a certain part of the text is required. At the moment, such tools for the Lithuanian language do not exist and this was the main cause for a number of false positive resolutions.
- The database of public persons has to be constantly updated as new information becomes available. Otherwise, the results will degrade when annotating newer texts.
- Writers of the articles often make mistakes when it comes to hypernym/hyponym and synonym usage. This problem was also noticed during the evaluation of the corpus, Section 4.1.
- As with pronouns, plural nouns are a problematic case.

Table 4.4 Experimental results for definitive nominal coreference resolution

Type	TP	WT	WL	WTL	FN	FP	S*
dtis	728	0	0	0	40	8	768
dais	22	0	5	0	14	4	41
dbis	223	0	9	0	13	5	245
dmis	0	0	0	0	14	0	14
Group references	0	0	0	0	3	0	3

Ambiguous	0	0	0	0	0	0	0
All	973	0	14	0	84	17	1,071

- Definitive nominal resolution produces the best results when compared to the previous two types of coreferences.
- This type of CR results in a small number of C1–C3 annotations, which makes it highly reliable.
- Some products and organizations have the same names (i. e. Google). The reader often needs some contextual information to correctly interpret such situations, but such contextual information is usually not available for automated algorithms. This is the main cause of false positive resolutions.
- The second problem is the Lithuanization of foreign names. There are official rules on how it should be done, but in practice, there are many variations, and it is difficult to account for all of them.

As seen in Section 3.2, the main algorithm can be divided into five modules: general pronoun resolution based on morphological and NER annotations, specific pronoun usage rules, PRA resolution, HHR resolution-based, and feature. In Table 4.5 aggregated results for each separate module are displayed. In *Total* column, only those cases that each module attempted to resolve are listed.

Table 4.5 Experiment results for each module

Algorithm	TP	WT	WL	WTL	FN	FP	S*
General pronoun resolution	135	30	27	25	156	29	373
Specific rules resolution	154	0	0	0	9	0	163
PRA	973	0	14	0	67	17	1,054
HHS	13	4	2	9	81	7	109
Feature resolution	110	0	19	0	69	16	198
All	1,385	34	62	34	382	69	1,897

Data provided in the first three tables have been used to evaluate the proposed resolution approach. First, we use the evaluation strategy proposed in Section 2.5 and at the end we provide the results with other evaluation strategies as well.

Precision, recall, and F-measure are calculated for pronominal (P_1, R_1, F_1), generic nominal (P_2, R_2, F_2), and definitive nominal (P_3, R_3, F_3) coreferences in Formulas (4.1)–(4.9).

$$P_1 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FP} = \frac{289+0.75*30+0.5*27+0.25*25}{289+30+27+25+29} = 82.8\% \quad \textbf{(4.1)}$$

$$R_1 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FN} = \frac{289+0.75*30+0.5*27+0.25*25}{289+30+27+25+182} = 59.9\% \quad \textbf{(4.2)}$$

$$F_1 = \frac{2P_1R_1}{P_1+R_1} = \frac{2*82.8*59.9}{82.8+59.9} = \frac{9919.44}{142.7} = 69.5\% \quad \textbf{(4.3)}$$

$$P_2 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FP} = \frac{123+0.75*4+0.5*21+0.25*9}{123+4+21+9+23} = 77.1\% \quad \textbf{(4.4)}$$

$$R_2 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FN} = \frac{123+0.75*4+0.5*21+0.25*9}{123+4+21+9+380} = 25.8\% \quad \textbf{(4.5)}$$

$$F_2 = \frac{2P_2R_2}{P_2+R_2} = \frac{2*77.1*25.8}{77.1+25.8} = \frac{3978.36}{102.9} = 38.7\% \quad \textbf{(4.6)}$$

$$P_3 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FP} = \frac{973+0.5*14}{973+14+17} = 97.6\% \quad \textbf{(4.7)}$$

$$R_3 = \frac{k_1TP+k_2WT+k_3WL+k_4WTL}{TP+WT+WL+WTL+FN} = \frac{973+0.5*14}{973+14+84} = 91.5\% \quad \textbf{(4.8)}$$

$$F_3 = \frac{2P_3R_3}{P_3+R_3} = \frac{2*97.6*91.5}{97.6+91.5} = \frac{17860.8}{189.1} = 94.5\% \quad \textbf{(4.9)}$$

Looking at the provided data and the *P*, *R*, *F* values calculated for each type of coreference, we can see that definitive nominal coreferences are solved noticeably better than pronominal or generic nominal ones. It is clear that definitive nominals are also over-represented in the corpus, and if we used micro averages, it would skew the overall results. Hence the proposed evaluation strategy suggests always using macro averages for the final score: precision (4.10), recall (4.11), and their F-measure (4.12). We calculate micro averages in Formulas (4.13)–(4.15). We can see that with micro average, the F-measure would get a 77.4% score, a significant difference.

$$P_{macro} = \frac{\sum_i^{n_a} P_i}{n_a} = \frac{82.8 + 77.1 + 97.6}{3} = 85.8\% \quad \textbf{(4.10)}$$

$$R_{macro} = \frac{\sum_i^{n_a} R_i}{n_a} = \frac{59.9 + 25.8 + 91.5}{3} = 59.1\% \quad \textbf{(4.11)}$$

$$F_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}} = \frac{2 * 85.8 * 59.1}{85.8 + 59.1} = 70\% \quad \textbf{(4.12)}$$

$$P_{micro} = \frac{(289+0.75*30+0.5*27+0.25*25)+(123+0.75*4+0.5*21+0.25*9)+(973+0.5*14)}{(289+30+27+25+29)+(123+4+21+9+23)+(973+14+17)} = 91.5\% \quad \textbf{(4.13)}$$

$$R_{micro} = \frac{(289+0.75*30+0.5*27+0.25*2)+(123+0.75*4+0.5*21+0.25*9)+(973+0.5*14)}{(289+30+27+25+182)+(123+4+21+9+380)+(973+14+84)} = 67.1\% \quad \textbf{(4.14)}$$

$$F_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} = \frac{2 * 91.5 * 67.1}{91.5 + 67.1} = 77.4\% \quad \textbf{(4.15)}$$

F_{maco} score shows how well the CR approach solves coreferences that it attempts to resolve, the following scores, Formulas (4.16)–(4.18), have been calculated to show how well it covers the used annotation scheme.

$$P_{scheme} = \frac{\sum_i^{n_a} w_{n_a} P_i}{n} = \frac{82.8 + 77.1 + 97.6}{5} = 51.5\% \quad \textbf{(4.16)}$$

$$R_{scheme} = \frac{\sum_i^{n_a} w_{n_a} R_i}{n} = \frac{59.9 + 25.8 + 91.5}{5} = 35.44\% \quad \textbf{(4.17)}$$

$$F_{scheme} = \frac{2P_{schem}R_{scheme}}{P_{scheme} + R_{scheme}} = \frac{2 * 51.5 * 35.44}{51.5 + 35.44} = 41.99\% \quad \textbf{(4.18)}$$

Additionally, ARCS, MUC, B^3, and CEAFE scores were calculated. Full results are displayed in Table 4.6. The overall results are not out of line with what other evaluation metrics scored our CR approach, but our evaluation strategy provides an additional dimension to the scoring process. Scheme coverage values are not presented since other evaluation metrics do not provide a similar metric.

Table 4.6 A comparison of different metrics

Evaluation strategy	P_{micro}	R_{micro}	F_{micro}	P_{macro}	R_{macro}	F_{macro}
Proposed strategy	91.5	67.1	77.4	85.8	59.1	70
ARCS	89.6	65.7	75.8	82.4	57.4	67.7
MUC	90.6	74.9	82	84.2	69.9	76.4
B^3	93.1	75.4	83.6	89.6	70.6	79
CEAFE	66.3	58.4	62.1	62.1	52.1	56.7

An additional experiment was carried out using medical records [141]. It was evaluated using the original MUC metric and got a macro-averaged F-measure of 56.7%. Due to privacy concerns, it will not be further detailed here. But it is evident that it scored noticeably lower than with the LCC data. It is well known that CR approaches tend to perform differently on different types of text. But to measure this variance, additional corpora focusing on different domains than LCC corpus would have to be created for the Lithuanian language. This task was out of scope for this research since the creation of corpora is a very expensive and time-consuming process. It could be the focus of future research in this field.

Overall, the algorithm provides encouraging results, comparable to other analysed resolution approaches, but at the same time it has certain limitations and room for improvements: it is domain specific, capable of resolving just a subset of coreference types, and was experimentally investigated for the relatively small set of articles. The future work can be directed towards investigating possibilities to adapt this solution for other relevant domains and expanding coreference coverage using emerging tools and resources for Lithuanian language that currently are under development. SSFLL provides a favourable environment for the creation and improvement of such tools.

4.3 Impact of coreference resolution on semantic annotations

It is difficult to determine how big of an impact CR can have on semantic annotations. Different annotators focus on different semantic information, their

methods of extracting semantic information might also differ.

The semantic annotator that is being used in SSFLL is rule-based. If the text fragment matches a certain pattern, then it extracts semantic information from it. Each block of semantic information has to be assigned to a certain object. If the focus of the text fragment is identified by NER, then the semantic annotator assigns a class to it based on the type of the NE. Otherwise, it is assigned the *Abstract Object* class. For example:

- [Dalia Grybauskaitė]$_{c1;c2}$ made a public statement yesterday. [She]$_{c1}$ criticized current government plans for healthcare reform. [President's]$_{c2}$ tone was very harsh.

In this case, a semantic annotator might be able to determine that Dalia Grybauskaitė, *Agent* of *Person* class, made a statement yesterday. It might also determine what she said in the statement and that her tone was very harsh. But since neither pronoun "she" nor noun "President" would be identified by NER, both of these mentions would get assigned the *Abstract Object* class. All three objects would be treated as separate entities. Furthermore, semantic search by generic pronoun like "she" is very limited. CR can solve this problem by linking pronouns and generic NPs to NEs. In such cases, only one object, "Dalia Grybauskaitė" *Agent* of *Person* class, would be created in the semantic database and all three blocks of semantic information would be linked to it.

Another issue is that NER annotation does not link different NE mentions of the same discourse-world entity with each other. Each NE is treated as unique. Therefore, a semantic annotator might consider that multiple different persons have been mentioned with Dalia Grybauskaitė's name in the text. CR solves this problem as well.

The best way to show the impact of CR on semantic annotations is to calculate how many named entities the NER identifies, how many pronouns and generic NPs are linked to them by CR, how many NE mentions of same discourse-world entity are linked to each other, and how many unique NEs (referring to different discourse-world entities) remain after that.

For the experiment, LCC was used as a source for texts. NER annotator from SSFLL was used for NE identification. Coreference annotations were created using the CR approach proposed in this work. First an NER annotator was run to annotate all NEs present in the corpus, then a CR annotator was used for a pronoun (or noun) linking with NEs and to establish links between different NE mentions of the same discourse-world entity. A list of dominant mentions and NEs that had no referents in the text was used to determine how many unique NEs there were after CR. Uniqueness is established only in the context of one text since the proposed approach solves only those coreference expressions that are present in the same text. The results of this experiment can be seen in Table 4.7.

Table 4.7 Coreference resolution impact on the semantic annotator

Type of NE	Number of identified NEs in the corpus	Number of pronouns and generic NPs linked to NEs	Number of established links between NEs	Unique NEs left in the context of one text
Person	572	373	209	363
Location	1,151	11	362	789
Organization	1,177	9	433	744

Results show that:

- With coreference annotations, the number of semantic information blocks linked to person NEs can be increased by 65%, from 572 to 945. It also significantly reduces the number of abstract objects in the semantic database.
- With coreference annotations, the amount of unique NEs for persons has decreased by 37%, from 572 to 363. The number of unique NEs would decrease even further with a resolution between different texts implemented.
- Location and organizations are rarely referred to by pronouns or generic NPs, therefore, the CR does not link many such expressions to them when compared to the results for persons. On the other hand, a decrease in the number of unique NEs is similar, 31% and 37%.
- Other types of NEs were not analyzed due to rarely occurring in CR context.
- It is important to note that CR does not ensure that semantic information will be extracted near each of those antecedents and referents. That depends entirely on the quality of semantic annotator.

Additionally, the same experiment was carried out focusing on cases where the antecedent would be generic NP. In total, 187 such cases were resolved. While a semantic search would still be limited due to semantic information being assigned the *Abstract Object* class, the number of separate abstract objects in the semantic database would significantly decrease.

4.4 Conclusions of chapter 4

1. Using the proposed annotation scheme and the created LCC corpus, the inter-annotator experiment was carried out. Three human annotators participated and reached 80.7% agreement. Based on these results, the annotation scheme and the corpus were updated.
2. The developed CR approach was evaluated using LCC corpus and the proposed evaluation strategy. It scored 85.8% for precision, 59.1% for recall, and 70% for F-measure.
3. The developed CR approach was also evaluated using four other evaluation strategies. While the results differ, they are generally in the same range: 62.1%–89.6% for precision, 52.1%–70.6% for recall, and 56.7%–79% for F-measure.
4. The experiment was also carried out to determine the impact that the developed CR approach had on the semantic annotator. The experiment showed an

increase in the number of semantic information blocks linked to a named entity of person type by up to 65%. Increases for named entities of location and organization types were not significant. But the decrease in the number of unique named entities was in the range from 31% to 37% for all three types of named entities.

5 CONCLUSIONS

1. An analysis of coreferences-related scientific literature showed that:
   1.1. In order to resolve coreferences, a coreference corpus with pre-annotated data is required. It is used for research, testing, and evaluation of the resolution approaches. The key component of the corpus is the annotation scheme detailing what and how, should be annotated. At the time there were no coreference corpora created for the Lithuanian language and the adaptation of corpora created for other languages is not feasible since it is the most language-dependent resource.
   1.2. Coreference resolution evaluation strategies are not language-dependent, but none of them have been accepted as the standard. While some strategies are more popular than others, all of them have various drawbacks.
   1.3. An analysis of the resolution approaches for English and Balto-Slavic languages revealed similar development. The initial resolution approaches are usually rule-based. After that, movement towards the machine and deep learning can be observed. Despite that, rule-based approaches are still relevant. They have higher adaptability and require less expensive language resources, which is very relevant to resource-scarce languages like Lithuanian.
   1.4. Adapting coreference resolution approaches from one language to another is problematic due to differences between different languages, lack of language-related resources, and variance in their quality.
2. Based on the analysis:
   2.1. The annotation scheme for the Lithuanian language was developed. Alongside the coreference annotations, it also includes a list of dominant mentions. Dominant mentions are semantically richest mentions of a discourse-world entity present in the text. They help with improving the results of semantic search and are useful for future research that could focus on coreference resolution between different texts. The core principles of the annotation scheme can be adapted to other languages even if the classification itself would require adjustments due to language differences.
   2.2. The developed annotation scheme was used in creating the first coreference corpus for the Lithuanian language – Lithuanian Coreference Corpus (LCC).
   2.3. A new strategy for evaluating coreference resolution approaches was proposed. It does not enforce transitivity, uses dominant mentions, and classifies coreference annotations into six different classes based on their accuracy, this allows differentiating between errors when calculating precision and recall.
   2.4. The conceptual model for rule-based coreference resolution in the Lithuanian language was created and specified applying a UML class diagram. It identifies key concepts and relationships between them. Using concepts from the model rules have been formalized with first-order logic. This allows to evaluate the adaptability of the proposed approach to other grammatically

similar languages.

3. The rule-based coreference resolution approach for the Lithuanian language was developed based on the models and rules specified in this work. The developed approach was integrated into semantic search framework. The output of the coreference resolution approach was used by a semantic annotator.
4. Three experiments have been carried out using the infrastructure of semantic search framework:

4.1. In order to evaluate the suitability of the proposed annotation scheme, the inter-annotator agreement experiment was performed using the LCC corpus. Three independent human annotators participated; they reached 80.7% agreement. Based on these results, common mistakes and problems were identified. The LCC and annotation scheme were updated to fix those shortcomings.

4.2. Using the proposed evaluation strategy and the LCC, corpus evaluation was performed for the developed coreference resolution approach. It reached 85.8% precision score, while using other evaluation strategies it got 62.1%–89.6% scores. The results are comparable to what the existing approaches focusing on other languages score, thus it can be said that the development of rule-based approach was justified.

4.3. Compared to other evaluation strategies, our proposed approach provided more detailed information. This additional information helps with error identification and highlights strong and weak points of the coreference resolution approach. The use of macro averages diminished the impact of imbalanced classes to the final score. The addition of scheme score allows to determine how well the coreference resolution approach covers the used annotation scheme.

4.4. Coreference resolution had a significant impact on the results of semantic annotator. The experiment showed that the developed coreference resolution approach can increase the number of semantic information blocks linked to a named entity by up to 65% and decrease the number of unique named entities by up to 37%. These results in turn can improve the results of semantic search or other higher-level applications.

6 SANTRAUKA

ĮVADAS

Sparčiai populiarėjant semantinio tinklo (angl. *Semantic Web*) technologijai, informacijos išgavimas ėmė keistis į teksto prasme paremtą informacijos išgavimą. Šis išgavimas dažnai vadinamas semantine paieška. Rastų dokumentų, susijusių su vartotojo poreikiais, kokybė labai priklauso ne tik nuo taikomų informacijos paieškos metodų, bet ir nuo naudojamų informacijos išgavimo metodų. Informacijos išgavimas yra veikla, skirta automatiškai išgauti struktūruotą informaciją iš nestruktūruoto informacijos šaltinio. Standartiniai teksto apdorojimo etapai, naudojami klasikiniuose informacijos išgavimo modeliuose, yra leksinė analizė, morfologinė analizė, įvardytos esybės atpažinimas. Kai kurie informacijos išgavimo sprendimai yra papildomi pažangesniais metodais, tokiais kaip koreferencijų sprendimas, semantinis anotavimo ir ontologijos užpildymas. Išgaunant informaciją pagrindinis iššūkis yra natūralios kalbos sudėtingumas ir dviprasmiškumas, todėl informacijos išgavimas priklauso nuo natūralios kalbos apdorojimo metodų pažangos. Nors naujausiomis technologijomis grįsti tyrimai gerai ištirtoms kalboms (pvz., anglų kalba) jau pasiekė sėkmingą praktinį taikymą dideliu mastu (pvz., *IBM Watson* projektas) [1], mažiau ištirtos kalbos, tokios kaip lietuvių kalba, išlieka atviru mokslinių tyrimų lauku.

Natūralios kalbos apdorojimo kontekste koreferencija yra, kai dvi skirtingos lingvistinės struktūros nurodo tą patį realaus pasaulio objektą. Šio ryšio identifikavimas ir sprendimas gali labai pagerinti semantinės paieškos, automatinio vertimo, klausimų atsakymo ir kitas panašias sistemas.

Pavyzdžiui, „Tomas šiandien neatėjo į mokyklą. Jis serga.“ Čia žodžiai „Tomas“ ir „Jis“ rodo tą patį realaus pasaulio objektą ir turi koreferencinį ryšį tarpusavyje. Neišsprendus šio ryšio nebūtų įmanoma nustatyti, nei kodėl Tomas neatėjo į mokyklą, nei kas toks sirgo. Tokiu atveju būtų prarandama semantinė informacija ir suprastėtų informacijos išgavimo procesas.

Šio darbo tikslas – sukurti metodus ir reikiamus išteklius koreferencijų sprendimui, skirtam lietuvių kalbai. Šiuo metu, mūsų žiniomis, sprendimų, skirtų lietuvių kalbai, nėra.

Motyvacija

Natūralios kalbos apdorojimo įrankių, skirtų lietuvių kalbai, vystymo pažanga leido sukurti semantinės paieškos karkasą[3] 2014 metais[3]. Šis karkasas yra orientuotas į klausimų atsakymą. Klausimai pateikiami struktūruota lietuvių kalba, paremta *Semantic of Business Vocabulary and Business Rules* [4] [5] [6] [7] [8]. Karkasas transformuoja šiuos klausimus į *SPARQL* užklausas ir įvykdo juos ontologijoje, kurią egzemplioriais užpildo semantinis anotatorius. Užklausos rezultatų kokybė (tikslumas ir atkūrimas) smarkiai priklauso nuo naudojamos natūralios kalbos apdorojimo

[3] „Syntactic and Semantic Analysis and Search System for Lithuanian Internet, Corpus and Public Sector Applications in Lithuanian Language” (No. VP2-3.1-IVPK-12-K-01-007) Projektas financuotas EU Struktūrinių fondų. Partneriai: Vytauto Didžiojo universitetas (koordinatorius), Kauno technologijos universitetas. (2012-2015)“.

grandinėlės, skirtos informacijai išgauti, ir kiekvieno jos komponento kokybės.

Pagrindinis semantinės paieškos komponentas yra semantinis anotatorius, išgaunantis semantinę informaciją iš teksto, kuris yra saugomas duomenų bazėje.Vėliau jame gali būti ieškoma informacija pagal vartotojo pateiktą užklausą. Tačiau problema kyla tada, kai tas pats realaus pasaulio objektas tekste yra minimas skirtingomis kalbinėmis struktūromis, tokiomis kaip įvardžiai, sinonimai, objekto savybės ir pan. Tai gali sukelti dvi problemas:

3) Dalis semantinės informacijos gali būti apskritai prarasta, jeigu realaus pasaulio objektas tekste minimas įvardžiu ar kita, dviprasmiška kalbine struktūra.
4) Net jei realaus pasaulio objektas yra minimas nedviprasmiška kalbine struktūra, vis tiek gali kilti problemų nustatant, ar ši struktūra nurodo tą patį realaus pasaulio objektą, kaip ir kita kalbinė struktūra, ar į skirtingą.

Dėl šių problemų semantinio anotatoriaus kokybė ir kartu semantinės paieškos gali suprastėti. Todėl buvo nuspręsta, kad reikalingas koreferencijų sprendimo komponentas, galintis spręsti šias problemas. Šis komponentas gali praturtinti ontologiją identifikuojant papildomus realaus pasaulio objektų paminėjimus ir ryšius tarp jų.

Deja, nėra daug kalbinių išteklių, skirtų lietuvių kalbai, kuri apskritai yra mažai tyrinėta natūralios kalbos apdorojimo srityje. Dėl to sudėtinga pritaikyti naujausiomis technologijomis grįstus koreferencijų sprendimus, sukurtus anglų ir kitoms plačiai ištirtoms kalboms. Todėl buvo nuspręsta, kad reikalingas naujas lietuvių kalbai skirtas koreferencijų sprendimas, kuris atsižvelgtų į turimus kalbinius išteklius.

Tyrimo sritis ir objektas

Tyrimo objektas yra koreferencijų sprendimo procesas, skirtas lietuvių kalbai.

Tyrimo sritis apima koreferencijų sprendimus, koreferencijų tekstynus ir susijusius natūralios kalbos apdorojimo išteklius bei orientuojasi į informacijos išgavimą, o ne į lingvistinę analizę.

Sprendžiama problema ir keliami klausimai

Nors tiriamieji darbai su įvairiomis lietuvių kalbai skirtomis natūralios kalbos apdorojimo dalimis yra atliekami, koreferencijų sprendimas išlieka neišnagrinėta sritis. Tyrimo klausimai:

1) Ar turint ribotus kalbinius išteklius (leksinės, morfologines ir įvardytų esybių anotacijos) galima sukurti tokį koreferencijų sprendimą, kurio veikimo rezultatai būtų naudingi aukštesnio lygio taikomosioms programoms?
2) Kokią įtaką koreferencijų sprendimas gali turėti semantinės paieškos rezultatams?
3) Ar verta dėti pastangų ir išteklių plėtojant taisyklėmis pagrįstus koreferencijų sprendimus, palyginti su mašininiu mokymusi pagrįstais sprendimais?
4) Kokiu laipsniu galimas tam tikros kalbos koreferencijų sprendimų pritaikymas kitoms, ribotus kalbinius išteklius turinčioms kalboms?

Tiksliai ir uždaviniai

Tyrimo tikslas – pagerinti koreferencijų sprendimo galimybes lietuvių kalbai sukuriant tam metodus ir reikalingus išteklius. Norint pasiekti šį tikslą, iškelti šie uždaviniai:

1) Analizuoti dabartinius metodus ir išteklius, naudojamus koreferencijų sprendimui anglų ir kitoms kalboms.
2) Sukurti anotavimo schemą ir koreferencijų tekstyną lietuvių kalbai, kurį būtų galima naudoti koreferencijų sprendimui testuoti ir vertinti.
3) Sukurti lingvistine informacija paremtą koreferencijų sprendimų vertinimo strategiją, kuri pasinaudotų teikiamais sukurtos anotavimo schemos privalumais.
4) Sukurti taisyklėmis paremtus modelius ir algoritmus, skirtus koreferencijų sprendimui lietuvių kalbai, kurie naudotų tik leksines, morfologines ir įvardytų esybių anotacijas.
5) Realizuoti sukurtus modelius ir algoritmus, skirtus koreferencijų sprendimui lietuviškiems tekstynams.
6) Atlikti eksperimentinį tyrimą ir įvertinti tinkamumą sukurtų modelių, algoritmų ir anotavimo schemoms.

Tyrimo metodika

Tyrimas atliktas naudojant *System Research Framework* (2004) [9].

Ginamieji teiginiai

1) Taisyklėmis grįstas sprendimas, naudojantis tik leksines, morfologines ir įvardytų esybių anotacijas, gali išspręsti dalį koreferencinių ryšių ir pasiekti patikimą tikslumą.
2) Semantinės paieškos rezultato kokybė priklauso nuo semantinio anotatoriaus galimybių identifikuoti objektus ir su jais susijusius faktus. Koreferencijų sprendimo rezultatas leidžia agreguoti išskaidytą semantinę informaciją. Dėl to koreferencijų sprendimas gali labai praturtinti semantines anotacijas ir kartu pagerinti semantinės paieškos rezultatus.
3) Koreferencijų sprendimo vertinimo strategija gali pateikti detalesnę ir naudingesnę informaciją, jeigu ji naudoja lingvistinę informaciją, esančią koreferencijų ryšiuose.

Asmeninis indėlis ir naujumas

Pagrindinis šio darbo indėlis:

- *Lithuanian Coreference Corpus* (LCC) tekstynas, skirtas koreferencijoms lietuvių kalboje.
- Koreferencijų sprendimo metodas, skirtas spręsti koreferencijų išraiškas lietuvių kalbai.
- Anotavimo schema, kuria remiantis buvo suanotuotas koreferencijų tekstynas. Koreferencijų sprendimas taip pat naudoja šią anotavimo schemą.
- Koreferencijų sprendimo įvertinimo modelis.

Tyrimo naujumas:

- Šiame darbe pristatyti LCC tekstynas ir koreferencijų sprendimo metodas yra pirmi tokie kalbiniai ištekliai lietuvių kalbai.
- Sukurta anotavimo schema yra labai lanksti ir nespecifikuoja realizacijos, todėl ją lengva pritaikyti kitoms kalboms ir integruoti į jau egzistuojančius sprendimus. Net jei koreferencijų klasifikavimą reikėtų pakeisti dėl skirtumų tarp kalbų (ar tyrimo specifikos), pagrindiniai siūlomos anotavimo schemos principai vis tiek būtų naudingi ir aktualūs.
- Sukurta koreferencijų sprendimo vertinimo strategija yra paremta lingvistine informacija. Esamas lingvistines metodikas papildo koreferencijų tipų identifikavimo vertinimu ir dominuojančiais paminėjimais. Ji taip pat sprendžia nesubalansuotų koreferencijų klasių dydžių problemą naudojant makro-, o ne mikroįverčius.
- Koreferencijų sprendimas naudoja minimalų kiekį kalbinių išteklių, tai jį daro naudingą ir kitoms kalboms, kurios neturi daug kalbinių išteklių.
- Taisyklėmis paremti koreferencijų sprendimai paprastai nėra formalizuoti, todėl dažnai sunku jų taisykles pritaikyti kitoms kalboms ar kontekstams. Sukurtas metodas yra taisyklėmis paremtas ir visos naudojamos taisyklės buvo formalizuotos naudojant pirmos eilės predikatų logiką. Tai jį padaro lengviau pritaikomą kitoms gramatiškai panašioms kalboms.

Praktinė reikšmė

Pristatytas sprendimas leidžia spręsti koreferencijas lietuvių kalbai ir taip pagerinti semantinio anotatoriaus rezultatus.

Šio tyrimo rezultatai buvo integruoti į *Semantic Search Framework* ir buvo naudoti anotuojant lietuvių kalbos interneto tekstyną.

Rezultatų aprobavimas

Šio darbo rezultatai buvo pristatyti trijose tarptautinėse ir vienoje nacionalinėje konferencijoje. Du straipsniai buvo paskelbti moksliniuose žurnaluose. Keturi straipsniai paskelbti kituose mokslo leidiniuose.

Disertacijos struktūra

Pirmame skyriuje pateikiama tyrimo analitinė dalis, ji padalinta į kelias mažesnes dalis. Pirmiausia apžvelgiamos pačios koreferencijos ir jų tipai. Tada koreferencijų tekstynai ir koreferencijų sprendimų vertinimo metrikos, o paskutiniame skyriuje egzistuojantys koreferencijų sprendimai kitoms kalboms.

Antrame skyriuje aprašoma sukurta anotavimo schema, anotavimo gairės, sukurtas tekstynas ir sprendimų vertinimo modelis. Trečiame skyriuje aprašomas koreferencijų sprendimas. Pateikiamas sprendimo algoritmas ir koncepcinis modelis, kuriuo remiantis formalizuotos sprendimo taisyklės.

Ketvirtame skyriuje aprašomas atliktas eksperimentas, skirtas sukurtiems sprendimams ir ištekliams įvertinti. Darbo apibendrinimas ir išvados pateikiami penktame skyriuje. Šeštame skyriuje pateikiama santrauka lietuvių kalba. Po santraukos pateikiami naudotos literatūros ir autoriaus publikacijų darbo tema sąrašai.

KOREFERENCIJŲ SPRENDIMŲ SRITIES ANALIZĖ

Koreferencijos ir jų tipai

Koreferencijų sprendimas yra procesas, kurio metu sujungiami objekto paminėjimai su išraiškomis, kurios juos nurodo [10]. Pavyzdžiui:

- Tomas praleido pamokas. Jis sirgo.

Šiuo atveju „Tomas“ yra objekto paminėjimas, o „Jis” yra jo referentas. Jei tekste būtų daugiau galimų paminėjimų, tai jie būtų vadinami paminėjimo kandidatais. Referentas nebūtinai turi būti įvardis, tai gali būti ir daiktavardis, nurodantis asmens profesiją, asmens vardo kartojimas ir t. t.

Paprastai literatūroje išskiriami atskiri reiškiniai, tokie kaip anafora, katafora, deiksė [11]. Šiame darbe šie reiškiniai nėra specifikuojami ir visi jie vadinami koreferencijomis. Svarbus išskyrimas yra tarp koreferencijų, esančių tame pačiame tekste (angl. *endaphora*), ir esančių skirtinguose tekstuose (angl. *exophora*). Šiame darbe tiriamos tik koreferencijos, esančios tame pačiame tekste.

Koreferencines išraiškas galima klasifikuoti pagal tai, kokia kalbos dalimi yra išreikštas referentas:

- Įvardžiuotinės (angl. *pronominal*), kai referentas yra išreiškiamas tam tikra įvardžio forma [15].
- Daiktavardinės (angl. *nominal*), kai referentas yra išreiškiamas tam tikra daiktavardžio forma ar daiktavardine fraze. Juos toliau galima išskaidyti į tikrinius ir bendrinius daiktavardžius [20].
- Prieveiksmio (angl. *adverbial*), kai referentas yra išreiškiamas tam tikra prieveiksmio forma [10].
- Elipsės (angl. *ellipsis*), kai referentas yra išreiškiamas praleistu žodžiu [26].

Taip pat dažnai prie koreferencijų yra priskiriamos tokios išraiškos, kaip prielaidos (angl. *pressupositions*), priedėliai (angl. *appositions*), asociacijos (angl. *associative*) ir kiti panašūs dariniai. Nėra aiškaus sutarimo, ar šie reiškiniai tikrai yra koreferencijos, ar tik turi panašias konstrukcijas, todėl šiame tyrime jie nėra plačiau nagrinėjami.

Koreferencijų tekstynai

Tekstynų ir jų anotavimo schemų analizė buvo atlikta analizuojant populiarius anglų ir kitų didžiųjų kalbų tekstynus: MUC [41], MATE/GNOME [42], ACE [52], OntoNotes [53].

Taip pat analizuoti čekų kalbos [49] ir lenkų kalbos [50] tekstynai, nes slavų kalbos turi panašumo su lietuviu kalba – yra morfologiškai turtingos ir turi laisvą žodžių tvarką.

Tekstynai ir jų anotavimo schemos skiriasi priklausomai nuo pasirinktos kalbos ir nagrinėjamos srities, todėl tiesiogiai jų pritaikyti kitai kalbai negalima.

MATE/GNOME padalina schemą į du sluoksnius. MATE yra bendro naudojimo anotavimo schema, o GNOME yra pritaikyta prie tam tikros realizacijos ir dalykinės srities. Šia darbe siūloma anotavimo schema yra panaši į MATE sluoksnį, tačiau GNOME ekvivalento nespecifikuoja ir palieka jį visiškai atvirą.

Lenkų kalbos tekstynas papildo savo schemą dominuojančiomis išraiškomis.

Dominuojanti išraiška – tai semantiškai turtingiausia teksto išraiška, apibūdinanti tam tikrą objektą. Šiame darbe naudojama modifikuota dominuojančių išraiškų versija – dominuojantys paminėjimai.

Išanalizuotos koreferencijų sprendimo vertinimo metrikos [64] [65] [66] [67] [71]. Kai kurios metrikos yra populiaresnės už kitas, tačiau nė viena nėra pripažinta kaip šios srities standartas.

Visos jos taip pat turi savų trūkumų. Vienas iš pagrindinių trūkumų yra tai, kad jos paremtos tranzityvumu ir visos to paties objekto paminėjimus tekste laiko lygiaverčiais, nors tai ne visada yra tiesa. Kita problema, jog rezultatai nėra diferencijuojami pagal klaidos sunkumą – neatsižvelgiama į lingvistinę informaciją. Šiame darbe siūloma metrika nėra paremta tranzityvumu ir leidžia diferencijuoti rezultatus pagal klaidos sunkumą.

Koreferencijų sprendimai, orientuoti į kitas kalbas

Analizuojant esamus sprendimus buvo apžvelgti klasikiniai [81] [98] ir naujausi [90] [108] [112] [116] sprendimai. Taip pat analizuoti lietuvių kalbai giminingoms kalboms sukurti sprendimai: latvių [119], lenkų [120], rusų [125] ir čekų [131].

Šiuo metu populiarėja mašininiu mokymusi grįsti sprendimai, tačiau jiems reikia daugiau kalbinių išteklių negu taisyklėmis grįstiems sprendimams. Kai kurie sprendimai parodė kylančias problemas, kai bandoma pritaikyti sprendimą, kuriam trūksta išteklių [122], arba bandoma reikiamus išteklius gauti atliekant projekciją [132].

Lietuvių kalba natūralios kalbos apdorojimo srityje yra mažiau ištirta negu lenkų, rusų ar čekų kalbos, todėl kalbinių išteklių trūkumas yra dar rimtesnė problema.

KOREFERENCIJŲ TEKSTYNAS LIETUVIŲ KALBAI

Koreferencijų anotavimo schema

Sukurta koreferencijų anotavimo schema matoma 6.1 lentelėje. Schema padalinta į keturis lygmenis. Pirmame nurodomas koreferencijos tipas, antrame potipis. Trečiame lygmenyje nurodoma referento pozicija paminėjimo atžvilgiu, ar jis eina pirmiau, ar vėliau, o gal tai nėra svarbu konkrečiu atveju. Ketvirtas lygmuo nurodo, ar referentas rodo į vieną, ar į kelis paminėjimus, o gal yra dviprasmybė ir neaišku, į kurį paminėjimą jis rodo.

Nustačius koreferencijos tipą, potipį, poziciją ir grupę, reiškiniui suformuojamas keturių raidžių kodas. Pavyzdžiui:

Sakinys: *Tomas Petrauskas praleido pamokas. Petrauskas sirgo.*
Kodas*:* *dais*

Šiuo atveju referentas „Petrauskas“ rodo į paminėjimą „Tomas Petrauskas“. Tai yra daiktavardinio tipo koreferencija, naudojanti daiktavardžiui tikrinti, todėl pirmoji kodo raidė yra „d“. Tai yra dalinis pakartojimas, praleistas vardas, todėl antroji gaunama raidė yra „a“. Šiuo atveju nesvarbu, ar pirma buvo paminėtas visas vardas, ar dalinis, todėl trečia gaunama raidė yra „I“. Referentas rodo tik į vieną paminėjimą, todėl paskutinė kodo raidė yra „s“.

6.1 lentelė. Koreferencijų anotavimo schema

Pirmas lygmuo	Antras lygmuo	Trečias lygmuo	Ketvirtas lygmuo
Įvardžiuotinė (p)	Asmenins (p)	Pozicija (a/p/i)	Grupė (g/a/s)
	Sangrąžinis (r)		
	Savybinis(o)		
	Santykinis (e)		
Daiktavardinė (g/d)	Pakartojimas (t)		
	Dalinis pakartojimas (a)		
	Sutrumpinimas (b)		
	Savybė (f)		
	Hipernimas-hiponimas (h)		
	Metonimas (m)		
	Sinonimas (s)		
Prieveiksmio (a)	-		
Elipsės (e)	Tas pats objektas (i)		
	Tokio paties tipo objektas (y)		
	Veiksmažodinė frazė (v)		

Kadangi visa anotavimo schema susiveda į keturių raidžių kodą, ją yra nesudėtinga pritaikyti įvairioms sistemoms ar dalykinėms sritims.

Anotavimo gairės

Dominuojantys paminėjimai – tai tokie paminėjimai, kurie yra semantiškai turtingiausi ir geriausiai apibūdina paminėtą objektą [50]. Šalia koreferencijų anotacijos koreferencijų sprendimo anotatorius turėtų pateikti ir sąrašą dominuojančių paminėjimų. Paminėjimus galima išrikiuoti pagal jų svorį: visas pavadinimas, sutrumpintas pavadinimas, dalinis pavadinimas, bendrinio daiktavardžio frazė ir to paties objekto elipsė. Tokie paminėjimai, kaip įvardžiai ar prieveiksmiai, negali būti dominuojantys, kadangi patys neturi semantinės informacijos. Jei du ar daugiau objekto paminėjimų yra tokio paties tipo, tada prioritetas teikiamas tam, kuris tekste paminėtas anksčiausiai.

Nėra nusistovėjusio standarto, kaip turėtų būti žymimi paminėjimai, maksimaliai ar minimaliai. Pavyzdžiui, „Lietuvos Respublikos Prezidentė Dalia Grybauskaitė“, naudojant maksimalų žymėjimą būtų pažymėta visa daiktavardinė frazė. Naudojant minimalų žymėjimą būtų pažymėta tik „Dalia Grybauskaitė“. Nustatyti minimalų paminėjimą, kuris geriausiai apibūdina objektą, yra sudėtingiau, tačiau toks žymėjimas yra naudingesnis. Ateityje, plečiant koreferencijų sprendimą tarp skirtingų tekstų, būtų lengviau tai padaryti naudojant minimalius paminėjimus, nes maksimalūs paminėjimai gali būti skirtingi skirtinguose tekstuose.

Bendriniai paminėjimai, kai kalbama apie kažką abstrakčiai, nėra žymimi. Singletonai, kai tekste yra tik vienas objekto paminėjimas ir jis neturi referentų, taip pat nėra žymimi.

Koreferencijų tekstynas

Sukurta koreferencijų tekstyną iš viso sudaro 100 straipsnių, dengiančių šias koreferencijas:

- 1217 daiktavardinių pakartojimų, dalinių pakartojimų ir sutrumpinimų,
- 553 įvardžiuotinų koreferencijų,
- 198 savybių,
- 61 hipernimas ir hiponimas,
- 48 metonimai,
- 48 sinonimai,
- 36 prieveiksmio koreferencijos,
- 17 elipsių.

Dabartinę tekstyno versiją galima pasiekti per Clarin-LT saugyklą [138]. Duomenys yra saugomi JSON formatu, vienos anotacijos pavyzdys:

- [{"Mentions": [["0, 3"], ["8, 3"]], “Referant": ["35, 4"], "Type": "ppag"}]

„Mentions“ masyve nurodomas sąrašas paminėjimų, į kuriuos rodo „Referant“ referentas. Pirmas skaičius nurodo paminėjimo (arba referento) startinę poziciją tekste, o antras nurodo teksto fragmento ilgį. „Type“ nurodo kodą, suformuotą pagal anotavimo schemą. Šiuo atveju referentas turi įvardžiuotinę („p“), asmeninę („p“), einančią po paminėjimo („a“) ir rodančią į kelis paminėjimus („g“), koreferenciją.

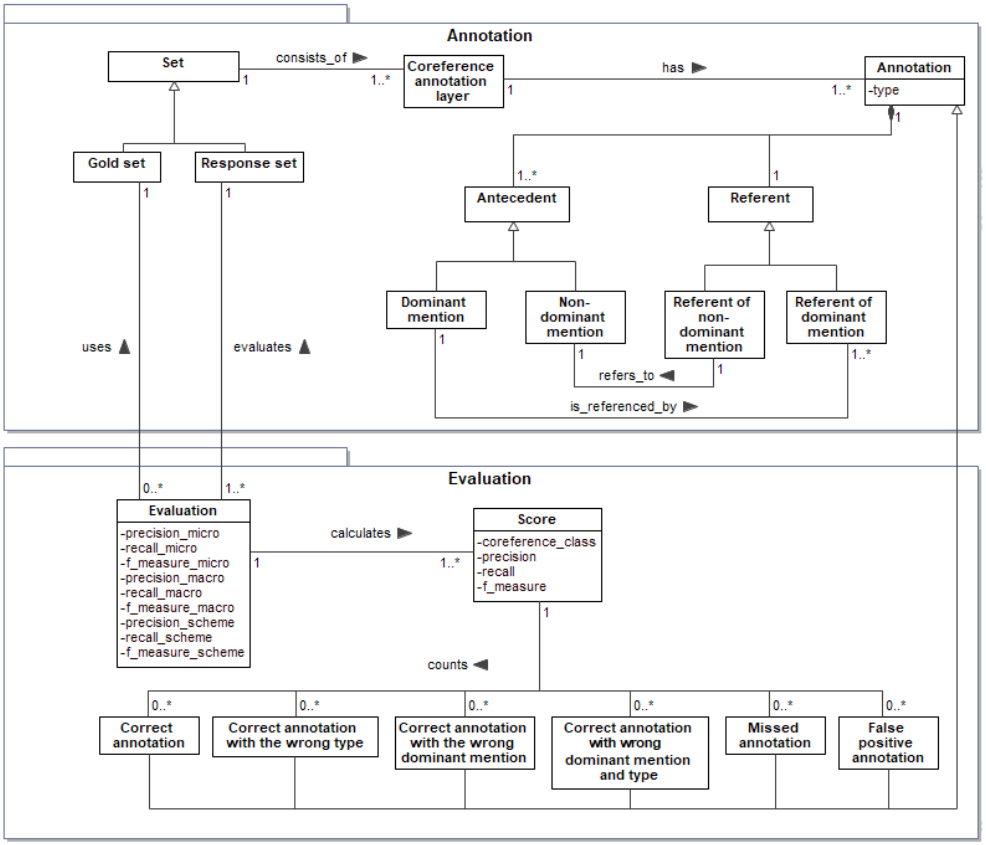
Koreferencijų vertinimo modelis

Šiame darbe siūloma vertinti koreferencijų anotacijas atsižvelgiant į jų lingvistinę informaciją ir į kitų kalbos apdorojimo komponentų poreikius. Vertinimo modelis pateiktas 6.1 pav.

Vertinimo modelį sudaro dvi dalys: anotacijos (*Annotation* paketas) ir vertinimas (*Evaluation* paketas). Anotacijų rinkinį (*Set*) sudaro vienas ar daugiau koreferencijų sluoksnių (angl. *Coreference annoation layer*). Vienas tekstas turi tik vieną koreferencijų sluoksnį. Vieną sluoksnį sudaro viena ar daugiau anotacijų (angl. *Annotation*), kuri atitinkamai susideda iš pirmtako (angl. *Antecedent*) ir referento (angl. *Referent*). Pirmtakai ir jų referentai gali būti skirstomi į dominuojančius (angl. *Dominant Mention*) ir nedominuojančius (angl. *Non-dominant mention*).

Anotacijų rinkiniai gali būti dviejų tipų: auksinis (angl. *Golden Set*) ir sistemos (angl. *Response Set*). Auksinis rinkinys yra toks rinkinys, kuris buvo sukurtas rankiniu būdu ir yra patikrintas ekspertų. Sistemos rinkinys yra toks rinkinys, kurį sukūrė koreferencijų sprendimo komponentas.

Įvertinimo (angl. *Evaluation*) metu sistemos anotacijų rinkinys yra lyginamas su auksiniu anotacijų rinkiniu. Įvertinimą sudaro vienas ar daugiau įverčių (angl. *Score*). Kiekvienai koreferencijų klasei (klasifikuojama pagal pasiūlytą anotavimo schemą, 3.1 skyrius) skaičiuojamas atskiras įvertis.



6.1 pav. Koreferencijų sprendimų vertinimo modelis

Kiekvienai koreferencijų klasei vertinami anotacijų elementai priskiriami vienai iš šešių klasių remiantis anotacijos elemento tikslumu. Tikslumui (angl. *Precision*) ir atkūrimui (angl. *Recall*) skaičiuoti kiekvienam anotacijos elementui yra priskiriamas svoris, priklausomai nuo to, į kurią klasę jis buvo priskirtas:

- Anotacijos elementas gauna koeficientą k_1, jeigu jis išspręstas visiškai teisingai (angl. *Correct element (TP)*).
- Anotacijos elementas gauna koeficientą k_2, jeigu jis išspręstas teisingai, bet neteisingai nustatytas tipas (angl. *Correct element with wrong type (WT)*).
- Anotacijos elementas gauna koeficientą k_3, jeigu buvo teisingai identifikuotas referentas, bet jis sujungtas su klaidingu dominuojančiu paminėjimu (angl.*Correct element with wrong dominant mention (WL)*).
- Anotacijos elementas gauna koeficientą k_4, jeigu buvo teisingai identifikuotas referentas, bet jis sujungtas su klaidingu dominuojančiu paminėjimu ir nustatytas neteisingas tipas (angl. *Correct element with wrong dominant mention and type (WTL)*).
- Anotacijos elementas, kuris buvo praleistas (angl. *Missed element (FN)*) arba jo žymėti nereikėjo (angl. *False positive element (FP)*), koeficiento negauna.

Koeficientų reikšmių aibė yra [0…1]. Šie koeficientai leidžia diferencijuoti skirtingo sunkumo klaidas. Kadangi naudojami makrovidurkiai, atliekami identiški skaičiavimai kiekvienai koreferencijos klasei atskirai: tikslumas – (6.1) formulė, atkūrimas – (6.2) formulė ir F matas, harmoninis tikslumo ir atkūrimo vidurkis, – (6.3) formulė.

$$P_i = \frac{k_1 TP + k_2 WT + k_3 WL + k_4 WTL}{TP + WT + WL + WTL + FP} \quad \textbf{(6.1)}$$

$$R_i = \frac{k_1 TP + k_2 WT + k_3 WL + k_4 WTL}{TP + WT + WL + WTL + FN} \quad \textbf{(6.2)}$$

$$F_i = \frac{2 P_i R_i}{P_i + R_i} \quad \textbf{(6.3)}$$

Palyginimo tikslais skaičiuojami ir mikroįverčiai. Tam naudojamos tokios pačios formulės, bet nėra skaičiuojama skirtingoms koreferencijų klasėms atskirai. Papildomai skaičiuojamas ir anotavimo schemos (F_{scheme}) įvertis, parodantis, kaip gerai koreferencijų sprendimas padengia tam tikrą anotavimo schemą.

KOREFERENCIJŲ SPRENDIMAS LIETUVIŲ KALBAI

Koreferencijų sprendimo algoritmas

Siūlomas algoritmas atsižvelgia į gramatines lietuvių kalbos taisykles, kurios yra paremtos leksemų morfologinėmis savybėmis ir jų pozicija sakinyje bei tekste.

Koreferencijų sprendimo algoritmą sudaro penki mažesni algoritmai, sprendžiantys skirtingas koreferencijų išraiškas:

- *A1: Specific rules resolution* – algoritmas, skirtas tam tikrų įvardžių panaudojimų atvejams spręsti;
- *A2: General pronoun resolution* – algoritmas, sprendžiantis atvejus, kai referentas yra įvardis, o paminėjimas yra įvardyta esybė;
- *A3: PRA (partial, repetition, acronym) resolution* – algoritmas, sprendžiantis įvairių formų, tikrinių daiktavardžių pakartojimus;
- *A4: HHS (hypernym, hyponym, synonymous) resolution* – algoritmas, sprendžiantis bendrinių daiktavardžių koreferencijas, kai naudojami hiponiminiai, hiperniminiai arba sinoniminiai profesijos paminėjimai;
- *A5: Feature resolution* – algoritmas, sprendžiantis koreferencijų atvejus, kai referentas yra tam tikra objekto savybė, pavyzdžiui, jo profesija.

Norint nuspręsti, kada naudoti vieną ar kitą algoritmą, buvo sudaryta sprendimų lentelė (6.2 lentelė).

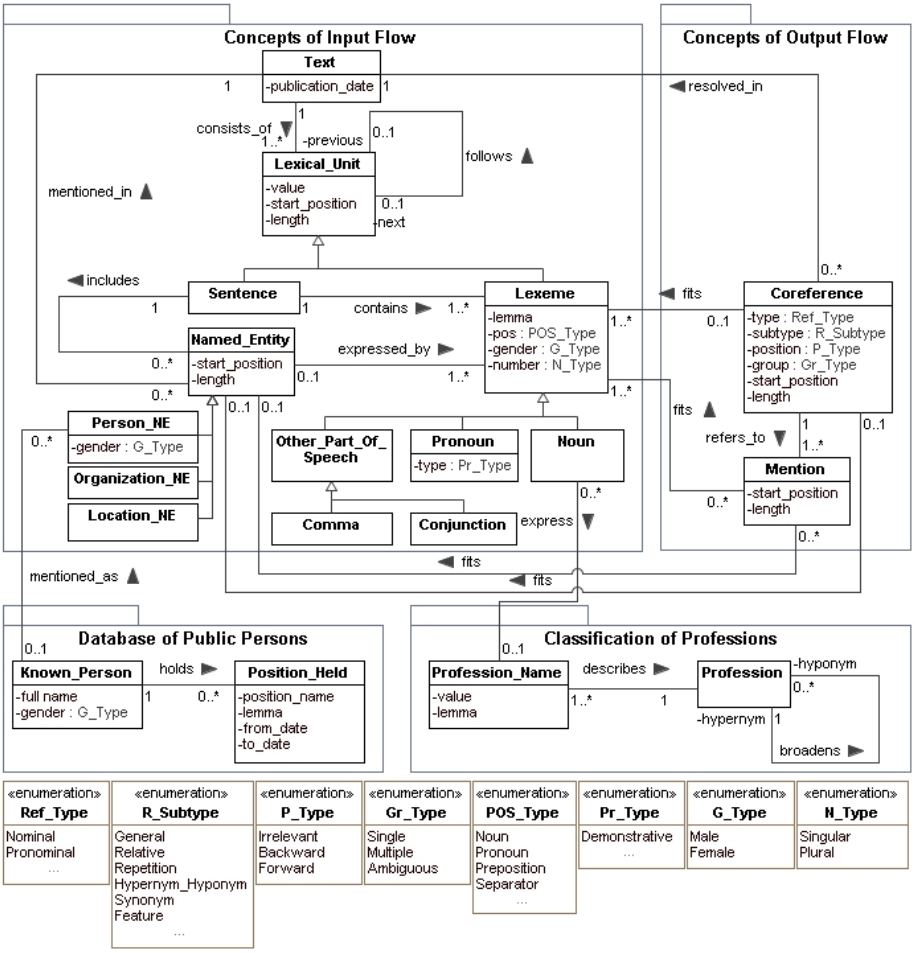
6.2 lentelė. Reikiamo algoritmo parinkimo sprendimų lentelė

Sąlyga	S1: Ar leksema yra įvardis?	Taip			Ne						Atsakymas
	S2: Ar egzistuoja specifinė taisyklė šiam įvardžiui?	Taip		Ne	-						
	S3: Ar įvardis buvo išspręstas specifinės taisyklės?	Taip	Ne	-	-						
	S4: Ar leksema yra daiktavardis?	-	-	-	Taip					Ne	
	S5: Ar daiktavardis atpažintas kaip įvardyta esybė?	-	-	-	Taip	Ne				-	
	S6: Ar daiktavardis egzistuoja profesinėje klasifikacijoje?	-	-	-	-	Taip		Ne		-	
	S7: Ar daiktavardis egzistuoja viešų asmenų žinių bazėje?	-	-	-	-	Taip	Ne	Taip	Ne	-	
Algoritmas	A1: Specific rules resolution	X		-	-	-	-	-	-	-	Sprendimas
	A2: General pronoun resolution	-	X	X	-	-	-	-	-	-	
	A3: PRA resolution	-	-	-	X	-	-	-	-	-	
	A4: HHS resolution	-	-	-	-	X		-	-	-	
	A5: Feature resolution	-	-	-	-	X	-	X	-	-	

Koncepcinis modelis

Siekiant formalizuoti koreferencijų sprendimo algoritmus, buvo identifikuoti aktualūs konceptai. Jie išreikšti UML klasių diagrama, matoma 6.2 pav. Visi šio modelio konceptai yra naudojami koreferencijų sprendimo formalizavimui pirmos eilės predikatų logika. Šioje santraukoje formalizuotai pateikiama viena *A1* taisyklė, todėl aprašomi tik jai aktualūs konceptai ir jų savybės.

Pagrindiniai įvesties konceptai yra tekstas (angl. *Text*) ir leksinis vienetas angl. (*Lexical_Unit*). Teksto konceptas atstovauja tekstiniam dokumentui ar naujienų straipsniui, kurio turinys yra analizuojamas. Kiekvienas tekstas yra sudarytas iš bent vieno leksinio vieneto. Paragrafai, sakiniai, žodžiai, skyrybos ženklai ir t. t., visi jie yra laikomi leksiniais vienetais. Kiekvienas leksinis vienetas turi tam tikrą reikšmę (angl. *value*), pradžios poziciją (angl. *start_position*), yra tam tikro ilgio (angl. *length*), priklauso tik vienam tekstui, gali eiti tik po vieno leksinio vieneto ir po jo gali eiti tik vienas leksinis vienetas tekste. Leksiniai vienetai klasifikuojami į sakinius (angl. *Sentence*) ir leksemas (angl. *Lexeme*). Leksemos konceptas atstovauja tokiems leksiniams vienetams, kaip žodžiai, skyrybos ženklai ir skaičiai. Kiekviena leksema turi pagrindinę žodžio formą, lemą (angl. *lemma*), kalbos dalį (angl. *pos*), giminę (angl. *gender*) ir skaičių (angl. *number*). Kiekviena leksema yra tik viename sakinyje, sakinys turi bent vieną leksemą.



6.2 pav. Koreferencijų sprendimo koncepcinis modelis

Leksemą galima specializuoti pagal kalbos dalį. Šiuo atveju išskiriamos trys leksemų kategorijos: daiktavardis (angl.*Noun*), įvardis (angl.*Pronoun*) ir kita kalbos dalis (angl.*Other_Part_Of_Speech*). Koreferencijų sprendime ne visi įvardžių tipai yra aktualūs, todėl įvardžių konceptas turi tipo (*type*) savybę. Kita kalbos dalies sąvoka specializuota į kablelio (angl. *Comma*) ir jungtuko (angl. *Conjuction*) konceptus, nes jie reikalingi tam tikroms struktūroms apibrėžti.

Pagrindinis koreferencijų sprendimo algoritmo išeigos konceptas yra koreferencija (angl. *Coreference*) – ryšys tarp paminėjimo ir referento. Kiekvienai koreferencijai yra nurodyti jos tipas (angl. *type*), potipis (angl. *subtype*), pozicija (angl. *position*) ir grupė (angl. *group*). Taip pat kiekviena koreferencija turi referento pradžios poziciją (angl. *start_position*) ir ilgį (angl. *length*). Referentas atitinka bent vieną teksto leksemą (angl. *Lexeme*). Kiekvienas referentas rodo į bent vieną paminėjimą (angl. *Mention*). Kiekvienas paminėjimas turi pradžios poziciją (angl. *start_position*), ilgį (angl. *length*) ir atitinka bent vieną leksemą.

A1. Specifinės taisyklės sprendimas (angl. *Specific rule resolution***)**. Tam tikrais atvejais natūralioje kalboje yra gana griežtai apibrėžtos tam tikrų įvardžių naudojimo struktūros. Pavyzdžiui:

LT: *Šiandien buvo atėjęs* ***vyras*** *[daiktavardis],* ***kuriuo*** *[įvardis] pasitikėjo Petras.*

PL: *Dzisiaj przychodził* ***mężczyzna*** *[daiktavardis],* ***który*** *[įvardis] skarżył się na ból pleców.*

RU: *Сегодня приходил* ***мужчина*** *[daiktavardis],* ***который*** *[įvardis] жаловался на боль в спине.*

EN: *A* ***man*** *[daiktavardis]* ***whom*** *[įvardis] Petras trusted have come today.*

Arba:

LT: *Šiandien buvo atėjęs* ***vyras*** *[daiktavardis],* ***su*** *[prielinksnis]* ***kuriuo*** *[įvardis] vakar išėjo Petras.*

PL: *Dzisiaj przychodził* ***mężczyzna*** *[daiktavardis],* ***z*** *[prielinksnis]* ***który*** *[įvardis] skarżył się na ból pleców.*

RU: *Сегодня приходил* ***мужчина*** *[daiktavardis],* ***с*** *[prielinksnis]* ***который*** *[įvardis] жаловался на боль в спине.*

EN: *A* ***man*** *[daiktavardis]* ***with*** *[prielinksnis]* ***whom*** *[įvardis] Peter left yesterday has come today.*

Abu šie pavyzdžiai turi tokią pačią konstrukciją: [daiktavardis] [kablelis] [neprivalomas prielinksnis] [specifinis įvardis]. Abiem atvejais įvardis „kuriuo“ rodo į daiktavardį „vyras“, vienu atveju turime prielinksnį „su“, kitu atveju – ne. Kaip matome iš tų pačių sakinių vertimų į lenkų ir rusų kalbas, struktūra nesikeičia ir taisyklę būtų galima pritaikyti ir šioms kalboms. Sąlyga šitokio ryšio egzistavimui formaliai gali būti apibrėžta taip:

Kiekvienam teksto ***t*** *sakiniui* ***s*** *ir kiekvienam „Relative“ tipo įvardžiui* ***p****, kuris yra sakinyje* ***s*** *ir turi startinę poziciją* ***sp1****, yra* ***ln1*** *ilgio, eina po kablelio* ***c*** *arba prielinksio leksemos* ***l1****, kuri eina po kablelio* ***c****, ir kiekvienam daiktavardžiui* ***l2****, kuris turi startinę poziciją* ***sp2****, yra* ***ln2*** *ilgio, eina prieš kablelį* ***c****, yra tos paties giminės* ***g*** *ir to paties skaičiaus* ***n*** *kaip įvardis* ***p****, tik vieną koreferencijos ryšį* ***r****, kuris yra išspręstas tekste* ***t****, yra „Pronominal“ tipo, „Relative“ potipio, „Backward“ pozicijos ir „Single“ grupės, tarp įvardžio* ***p*** *ir daiktavardžio* ***l2****, jo referentas prasideda pozicijoje* ***sp1*** *ir turi* ***ln1*** *ilgį, kuris atitinka tik vieną leksemą* ***p*** *ir rodo tik į vieną paminėjimą* ***m****, kuris prasideda pozicijoje* ***sp2*** *ir turi* ***ln2*** *ilgį ir atitinka tik vieną leksemą* ***l2****, egzistuoja (6.4) pirmos eilės predikatų formulė.*

$$\begin{aligned}&\forall t, s, p, l1, c, l2, g, n, sp1, sp2, ln2.\ [Text(t) \wedge Sentence(s) \wedge consists\_of(t, s) \wedge Pronoun(p) \wedge contains(s, p) \wedge has\_type(p, Relative) \wedge has\_start\_position(p, sp1) \wedge has\_length(p, ln1) \wedge Comma(c) \wedge (follows(p, c) \vee (Lexeme(l1) \wedge has\_pos(l1, Preposition) \wedge follows(l1, c) \wedge follows(p, l1)) \wedge Noun(l2) \wedge follows(l2, c) \wedge has\_gender(p, g) \wedge has\_gender(l2, g) \wedge has\_number(p, n) \wedge has\_number(l2, n) \wedge has\_start\_position(l2, sp2) \wedge has\_length(l2, ln2)\\ &\rightarrow \exists!r\ \exists!m.\ [Coreference(r) \wedge resolved\_in(r, t) \wedge has\_type(r, Pronominal) \wedge has\_subtype(r, Relative) \wedge has\_position(r,\end{aligned} \quad (6.4)$$

Backward) ∧ has_ group(r, Single) ∧ has_start_position(r, sp1) ∧ has_length(r, ln1) ∧ fits(r, p) ∧ Mention(m) ∧ refers_to(r, m) ∧ has_start_position(m, sp2) ∧ has_length(m, ln2) ∧ fits(m, l2)]]

EKSPERIMENTINIS TYRIMAS

Sudarius anotuotą tekstyną buvo atliktas sutapimo testas tarp trijų skirtingų žmonių anotatorių: *Anotatorius A*, *Anotatorius B* ir *Anotatorius C*. Jie rankiniu būdu anotavo visą LCC tekstyną, ir jų anotacijos buvo sulygintos norint nustatyti bendrą sutapimo lygį ir identifikuoti dažnas klaidas bei kylančias problemas. Šio eksperimento rezultatai matomi 6.3 lent.

6.3 lentelė. Sutapimo tarp anotatorių eksperimento rezultatai

Vertinami anotatoriai	Sutapimo procentas
Anotatorius A ir *Anotatorius B*	94,.3%
Anotatorius A ir *Anotatorius C*	87,8%
Anotatorius B ir *Anotatorius C*	86,1%
Anotatorius A, Anotatorius B ir *Anotatorius C*	80,7%

Remiantis gautais rezultatais buvo identifikuotos pasikartojančios klaidos ir kylantys neaiškumai anotavimo schemoje. LCC tekstynas ir anotavimo schema buvo atnaujinti siekiant ištaisyti identifikuotas problemas.

Kitame etape, naudojant LCC tekstyną, buvo atliktas pasiūlyto koreferencijų sprendimo algoritmo eksperimentas. Eksperimento rezultatai matomi 6.4 lent. Rezultatai pateikiami atskirai kiekvienam smulkesniam algoritmui ir išskaidyti pagal siūlomą vertinimo modelį.

6.4 lentelė. Koreferencijų sprendimo eksperimento rezultatai

Algoritmas	TP	WT	WL	WTL	FN	FP	S*
General pronoun resolution	135	30	27	25	156	29	373
Specific rules resolution	154	0	0	0	9	0	163
PRA	973	0	14	0	67	17	1054
HHS	13	4	2	9	81	7	109
Feature resolution	110	0	19	0	69	16	198
Visi	1385	34	62	34	382	69	1897

Galutiniai rezultatai skaičiuojami naudojant pirmąją MUC metriką, 6.5 formulė, ir pagal šiame darbe siūlomą vertinimo modelį, 6.6 formulė.

$$\mathrm{F} = \frac{2PR}{P+R} = \frac{2*88{,}7*59{,}6}{88{,}7+59{,}6} = \frac{10573{,}04}{148{,}3} = 71{,}3\% \qquad \textbf{(6.5)}$$

$$F_{final} = \frac{F_p+F_g+F_d+F_a+F_e}{5} = \frac{69{,}5+38{,}3+94{,}5}{5} = 40{,}46\% \qquad \textbf{(6.6)}$$

Rezultatas pagal pasiūlytą vertinimo metodiką yra prastesnis, tačiau labiau atitinkantis realią situaciją. Pavyzdžiui, *PRA*, algoritmas sprendė daugiau kaip pusę koreferencijų, esančių tekstyne, ir pasiekė aukštą rezultatą (94,5%), todėl pagal MUC metriką iškreipė galutinį rezultatą, nors kito tipo koreferencijos buvo sprendžiamos prasčiau arba visai nesprendžiamos.

Norint įvertinti koreferencijų sprendimo įtaką semantiniam anotatoriui ir semantinei paieškai, buvo atliktas eksperimentas naudojant LCC ir semantinės paieškos karkaso anotatorius. Šio karkaso semantinis anotatorius yra taisyklėmis paremtas, išgautą semantinę informaciją priskiria aktualiam objektui. Objektas gauna arba įvardytos esybės tipą (asmuo, vieta, organizacija), arba abstraktaus objekto, jei jis nebuvo atpažintas kaip įvardyta esybė. Kiekvienas toks objektas laikomas unikaliu, net jei ir mini tą patį realaus pasaulio objektą. Koreferencijų sprendimas pagerina semantinio anotatoriaus (ir semantinės paieškos) kokybę, susiedamas šiuos skirtingus to paties objekto paminėjimus.

Eksperimento metu buvo nustatyta, kiek kiekvienos įvardytos esybės tipo egzempliorių yra tekstyne, kiek prie jų buvo prijungta abstrakčių objektų (įvardžiai, bendriniai daiktavardžiai) ir kiek sukurta ryšių tarp įvardytų esybių po koreferencijų sprendimo. Taip pat suskaičiuota, kiek liko unikalių įvardytų esybių. Šio eksperimento rezultatai pateikiami 6.5 lent.

6.5 lentelė. Koreferencijų sprendimo poveikio semantiniam anotatoriui tyrimo rezultatai

Įvardytos esybės tipas	**Įvardytų esybių skaičius tekstyne**	**Abstrakčių objektų nuorodų į įvardytas esybes skaičius**	**Ryšių tarp įvardytų esybių skaičius**	**Unikalių įvardytų esybių likutis**
Asmenys	572	373	209	363
Vietos	1151	11	362	789
Organizacijos	1177	9	433	744

Rezultatai parodė, kad su koreferencijų anotacijomis:

- Semantinės informacijos blokų, rodančių į asmens įvardytas esybes, skaičius gali padidėti 65 %, nuo 572 iki 945. Šis padidėjimas smarkiai sumažina abstrakčių objektų skaičių semantinėje duomenų bazėje.
- Unikalių asmens įvardytų esybių skaičius sumažėjo 37 %, nuo 572 iki 363.
- Unikalių vietos ir organizacijos esybių skaičius sumažėjo 31 % ir 37 %. Vietovės ir organizacijos retai yra minimos įvardžiais arba bendriniais daiktavardžiais, todėl ir koreferencijų sprendimas neišsprendė daug tokių atvejų.

IŠVADOS

1. Koreferencijų srities mokslinės literatūros analizė parodė, kad:
   1.1. Norint spręsti koreferencijas, koreferencijų tekstynas su iš anksto suanotuotais duomenimis yra reikalingas. Jis naudojamas sprendimui vystyti, testuoti ir įvertinti. Esminė tekstyno sudedamoji dalis yra koreferencijų anotavimo schema, kuri nurodo, ką ir kaip reikia anotuoti. Analizės metu nebuvo koreferencijų tekstyno lietuvių kalbai, o tekstynų sukurtų kitoms kalboms pritaikytas nėra galimas dėl skirtumų tarp kalbų.
   1.2. Koreferencijų sprendimų vertinimo strategijos nėra priklausomos nuo apdorojamos kalbos, skirtos koreferencijų sprendimo kokybei įvertinti,

tačiau nė viena nėra priimta kaip šios srities standartas. Kai kurios strategijos yra populiaresnės negu kitos, tačiau visos turi savų trūkumų.

1.3. Koreferencijų sprendimo metodų analizė anglų ir baltų-slavų kalboms parodė panašią raidą. Pirmieji sprendimo metodai buvo paremti taisyklėmis, vėliau buvo pereita prie mašininio mokymosi metodų. Nepaisant to, taisyklėmis paremti metodai nepraranda savo vertės. Jie pasižymi lankstumu, jiems reikia mažiau išsamių ir reprezentatyvių kalbinių išteklių. Tai labai svarbu tokioms kalboms, kaip lietuvių, kurios neturi daug kalbinių išteklių.

1.4. Pritaikyti vienos kalbos koreferencijų sprendimus kitai kalbai yra sudėtinga dėl skirtumų tarp skirtingų kalbų kalbinių išteklių trūkumo ir kokybinio skirtumo tarp jų.

2. Remiantis atlikta analize:

2.1. Sukurta anotavimo schema, skirta lietuvių kalbai. Šalia koreferencijų anotacijų pateikiamas ir dominuojančių paminėjimų sąrašas. Dominuojantys paminėjimai yra semantiškai turtingiausi paminėjimai, kurie geriausiai apibūdina realaus pasaulio objektą. Jie leidžia pagerinti semantinės paieškos rezultatus. Dominuojantys paminėjimai taip pat gali būti naudingi tolimesniems tyrimams, kuriuose būtų sprendžiamos koreferencijos tarp skirtingų tekstų. Pagrindiniai schemos sudarymo principai gali būti pritaikyti kitoms kalboms, net jei pati koreferencijų klasifikacija turėtų būti keičiama dėl skirtumų tarp kalbų.

2.2. Anotavimo schema buvo išbandyta sukuriant pirmą koreferencijų tekstyną lietuvių kalbai – Lithuanian Coreference Corpus (LCC).

2.3. Pasiūlyta nauja koreferencijų sprendimų vertinimo strategija nesiremia tranzityvumu, naudoja dominuojančius paminėjimus ir klasifikuoja anotacijas į šešias skirtingas klases, priklausomai nuo jų tikslumo. Šios vertinimo modelio savybės leidžia geriau diferencijuoti klaidas skaičiuojant koreferencijų atpažinimo tikslumą ir atkūrimą.

2.4. Koreferencijų sprendimas lietuvių kalbai konceptualizuotas ir specifikuotas naudojant UML klasių diagramą. Jis identifikuoja esmines sąvokas ir ryšius tarp jų. Sprendimo taisyklės formalizuotos panaudojant šio koncepcinio modelio sąvokas ir pirmos eilės predikatų logiką. Tai leidžia įvertinti pasiūlyto sprendimo pritaikomumą kitai, gramatiškai panašiai, kalbai.

3. Taisyklėmis paremtas koreferencijų sprendimas lietuvių kalbai buvo realizuotas remiantis šiame darbe specifikuotais modeliais ir taisyklėmis. Realizuotas sprendimas integruotas į semantinės paieškos karkaso aplinką. Koreferencijų sprendimo rezultatai buvo naudojami semantinio anotatoriaus.

4. Naudojant semantinės paieškos karkaso infrastruktūrą atlikti 3 eksperimentiniai tyrimai parodė:

4.1. Norint įvertinti pasiūlytos anotavimo schemos tinkamumą buvo atliktas sutapimo eksperimentas naudojant LCC tekstyną. Dalyvavo trys nepriklausomi žmonės anotatoriai ir pasiekė 80,7 % sutapimo įvertį. Remiantis šiais rezultatais pasikartojančios klaidos ir problemos buvo identifikuotos, LCC ir anotavimo schema buvo atnaujinti.

4.2. Taikant darbe pasiūlytą koreferencijų sprendimo vertinimo strategiją ir LCC tekstyną sukurtam koreferencijų sprendimui, gautas 85,8 % tikslumo įvertis. Taikant kitas metrikas gauti 62,1–89,6 % tikslumo įverčiai. Palyginti su sprendimais, skirtais kitoms kalboms, sukurto sprendimo rezultatai yra panašūs. Tai leidžia teigti, kad taisyklėmis grįsto sprendimo kūrimas pasiteisino.

4.3. Lyginant su kitomis vertinimo strategijomis pasiūlyta vertinimo strategija pateikia detalesnę informaciją. Ši informacija padeda identifikuojant klaidas ir išryškina stipriąsias ir silpnąsias koreferencijų sprendimo vietas. Makrovidurkių naudojimas sumažina nesubalansuotų koreferencijų klasių įtaką galutiniam rezultatui. Schemos įvertis leidžia atsižvelgti į tai, kaip gerai koreferencijų sprendimas padengia naudojamą anotavimo schemą.

4.4. Koreferencijų sprendimas daro reikšmingą įtaką semantinio anotatoriaus rezultatui. Eksperimentas parodė, kad sukurtasis koreferencijų sprendimas gali padidinti semantinės informacijos blokų, rodančių į įvardytą esybę, skaičių iki 65 % ir sumažinti unikalių įvardytų esybių skaičių iki 37 %. Šie rezultatai gali pagerinti semantinės paieškos ir kitų aukštesnio lygio sistemų rezultatus

REFERENCES

1. **Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Welty, C. et al.** Building Watson: An overview of the DeepQA project. AI magazine, 31(3), pp. 59-79, 2020.
2. **Liu, Z., Shi, K., Chen, N. F.** (2021). Coreference-aware dialogue summarization. *arXiv preprint arXiv:2106.08556*.
3. **Vileniškis, T., Šukys, A., Butkienė, R**. Searching the web by meaning: a case study of Lithuanian news websites // Communications in computer and information science : Knowledge discovery, knowledge engineering and knowledge management : 7th international joint conference, IC3K 2015 Lisbon, Portugal, November 12–14, 2015 : revised selected papers / Fred A, Dietz J, Aveiro D, Liu K, Filipe J. (eds). Cham: Springer. ISSN 1865-0929. 2016, vol. 631, pp. 47-64.
4. **OMG**: Semantics of Business Vocabulary and Business Rules (SBVR). SBVR 1.2, Version 1.2, OMG Document Number: formal/2013-11-04, pp. 1–292, 2012.
5. **Sukys, A., Nemuraite, L., Sinkevicius, E., Paradauskas, B**.: Querying Ontologies on the Base of Semantics of Business Vocabularies and Business Rules. In: Information Technologies' 2011: Proceedings of the 17th International Conference on Information and Software Technologies, IT 2011, Kaunas, Lithuania, April 27–29, pp. 247–254, 2011.
6. **Sukys, A., Nemuraite, L., Paradauskas, B.:** Representing and Transforming SBVR Question Patterns into SPARQL. In: Information and Software Technologies: 18th International Conference, ICIST 2012, Kaunas, Lithuania, September 13–14,. CCIS, vol. 319, pp. 436-451. Berlin, Heidelberg: Springer, 2012.
7. **Bernotaityte, G., Nemuraite, L., Butkiene, R., Paradauskas, B**.: Developing SBVR Vocabularies and Business Rules from OWL2 Ontologies. In: Information and Software Technologies, 19th International Conference, ICIST 2013, CCIS, vol. 403, pp. 134−145. Springer, Berlin, Heidelberg, 2013.
8. **Karpovic, J., Krisciuniene, G., Ablonskis, L., Nemuraite, L.**: The Comprehensive Mapping of Semantics of Business Vocabulary and Business Rules (SBVR) to OWL 2 Ontologies. Information Technology and Control 43(3), pp. 289–302, 2014.
9. **Hevner, A.R., March, S. T., Park, J., Ram, S.** Design Science in Information Systems Research. In *MIS Quarterly*, Volume 28, Issue 1, USA, March 2004, pp. 75-105.
10. **Mitkov, R**. *Anaphora resolution*. Routledge, 2014.
11. **Krahmer, E., Piwek, P.** Varieties of anaphora. *Introduction. In dies.(Hg.), Varieties of Anaphora. Reader ESSLLI*, 2000, pp. 1-15.
12. **Elango, P**. Coreference Resolution: A Survey. Technical Report, University of Wisconsin-Madison, USA, 2005.
13. **Gardelle**, **L.** "'Anaphora', 'anaphor' and 'antecedent' in nominal anaphora: definitions and theoretical implications", Cercles, 2012, 22, pp. 25–40.
14. **Mel'čuk, I**. *Dependency in Linguistic Description*, CA, 2009.
15. **Fischer, S.** Pronominal anaphora. *Syntax–Theory and analysis: An international handbook*, 2015, 1: pp. 446-477.
16. **Büring, D**. Pronouns. 2011.

17. **Delmonte, R.** Relative clause attachment and anaphora: A case for short binding. In: *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 6)*. 2002. pp. 84-89.
18. **Gundel, J., hedberg, N., zacharski, R.** Pronouns without NP antecedents: How do we know when a pronoun is referential. *Anaphora processing: linguistic, cognitive and computational modelling*, 2005, pp. 351-364.
19. **Dimitriadis, A**. Beyond identity: Topics in pronominal and reciprocal anaphora. 2000.
20. **Rösiger, I., Teufel, S.** Resolving coreferent and associative noun phrases in scientific text. In: *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 2014. pp. 45-55.
21. **Hou, Y., Markert, K., Strube, M.** Global inference for bridging anaphora resolution. In: *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 2013. pp. 907-917.
22. **Souza, M., et al.** Nominal Coreference Annotation in IberEval2017: The Case of FORMAS Group. In: *IberEval@ SEPLN*. 2017. p. 92-101.
23. **Branco, A., Mcenery, T., Mitkov, R. (ed.).** *Anaphora processing: linguistic, cognitive and computational modelling*. John Benjamins Publishing, 2005.
24. **Ceberio, K., et al**. Coreferential relations in Basque: the annotation process. *Journal of psycholinguistic research*, 2018, 47.2: pp. 325-342.
25. **Van Deemter, K., Kibble, R.** What is coreference, and what should coreference annotation be?. In: *Proceedings of the Workshop on Coreference and its Applications*. Association for Computational Linguistics, 1999. pp. 90-96.
26. **Saeboe, K.** Anaphoric presuppositions and zero anaphora. *Linguistics and Philosophy*, 1996, 19.2: 187-209.
27. **King, Jeffrey C. and Lewis, Karen S**., "Anaphora", *The Stanford Encyclopedia of Philosophy* (Winter 2021 Edition), Edward N. Zalta (ed.), URL = https://plato.stanford.edu/archives/win2021/entries/anaphora/, [Accessed 3 June 2019].
28. **Beaver, David I. and Geurts, Bart**, "Presupposition", *The Stanford Encyclopedia of Philosophy* (Spring 2021 Edition), Edward N. Zalta (ed.), URL = https://plato.stanford.edu/archives/spr2021/entries/presupposition/, [Accessed 7 January 2024].
29. **Pagin P., Westerståhl D.** Predicate logic with flexibly binding operators and natural language semantics, *Journal of Logic*, Language and Information 1993, Volume 2, Issue 2, pp 89-128.
30. **Geurts, Bart and Beaver, David I.** Discourse Representation Theory, *The Stanford Encyclopedia of Philosophy (*Spring 2020 Edition). Available from: http://plato.stanford.edu/archives/spr2020/entries/discourse-representation-theory/, [Accessed 5 January 2024].
31. **Stanford CoreNLP**: Annotations and Annotators, URL = https://stanfordnlp.github.io/CoreNLP/pipelines.html, [Accessed 3 June 2019].
32. **Maynard, D., et al.** "A framework for real-time semantic social media analysis." *Journal of Web Semantics* 44 (2017): 75-88.
33. **Confluence**: Default Clinical Pipeline, URL = https://cwiki.apache.org/confluence/display/CTAKES/Default+Clinical+Pipeline, [Accessed 3 June 2019].

34. **Poesio, M., Stuckardt, R., Versley, Y. (ed**.). *Anaphora resolution: Algorithms, resources, and applications*. Springer, 2016.

35. **Kaplan, Ronald M.** "A method for tokenizing text." *Inquiries into words, constraints and contexts* 55 (2005).

36. **Adhvaryu, N., Balani, P.** Survey: Part-Of-Speech Tagging in NLP. In: *International Journal of Research in Advent Technology (E-ISSN: 2321-9637) SpecialIssue 1st International Conference on Advent Trends in Engineering, Science and Technology "ICATEST 2015"*. 2015.

37. **Sanders, A. F., Sanders, R., H.** Syntactic parsing: A survey. *COMP. HUM.*, 1989, 23.1: 13-30.

38. **Nadeau, D., Sekine S.** A Survey of Named Entity Recognition and Classification. In *Lingvisticae Invetigationes*, Volume 30, Issue 1, 2007, pp. 3-26.

39. **Recasens, M.** Coreference, Theory, Annotation, Resolution and Evaluation, University of Barcelona, 2010.

40. **Deemter, K., Kibble, R.** "On coreferring: Coreference in MUC and related annotation schemes", Computational Linguistics, 2000, 26(4), pp. 629–637.

41. **Chinchor, N., Hirschmann, L**. MUC-7 coreference task definition, version 3.0. In: *Proceedings of MUC*. 1997.

42. **Poesio, M**. "The MATE/GNOME Proposals for Anaphoric Annotation, Revisited", SIGDIAL Workshop, 2004.

43. **Krasavina, O., Chiarcos, C.** "PoCoS: Potsdam coreference scheme", Proceedings of the Linguistic Annotation Workshop, 2007, pp. 156–163.

44. **Recasens, M., Martí, M. A.** "AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan", Lang Resources & Evaluation, 2009, 44(4), pp. 315–345.

45. **Toldova, S. Ju., et al**. "RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian", Komp'juternaja Lingvistika i Intellektual'nye Tehnologii, 2014, pp. 681–694.

46. **Krasavina, O., Chiarcos, Ch., Zalmanov, D**. "Aspects of topicality in the use of demonstrative expressions in German, English and Russian", Proc. of DAARC-2007, Lagos, Portugal, 2007.

47. **Nedoluzhko, A.,Mírovský, J.** "How Dependency Trees and Tectogrammatics Help Annotating Coreference and Bridging Relations in Prague Dependency Treebank", Proceedings of the Second International Conference on Dependency Linguistics, Depling, 2013, pp. 244–251.

48. **Nedoluzhko, A., Mírovský, J.** "Annotating Extended Textual Coreference and Bridging Relations in the Prague Dependency Treebank. Annotation manual", Technical report No. 44, UFAL MFF UK, Prague, 2011.

49. **Nedoluzhko, A., et al.** "Coreference in Prague Czech-English Dependency Treebank", Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), Paris, France, 2016, pp. 169–176.

50. **Ogrodniczuk, M., et al.** "Polish Coreference Corpus", Challenges for Computer Science and Linguistics, 2016, pp. 215–226.

51. **Ogrodniczuk, M., et al.** "Interesting linguistic features in coreference annotation of an inflectional language", In: Sun, M., Zhang, M., Lin, D., Wang, H. (eds.) CCL and NLP-NABD 2013. LNCS, vol. 8202, 2013, pp. 97–108.

52. **Doddington, G., et al** "The Automatic Content Extraction (ACE) program tasks, data, and evaluation", Proceedings of LREC. 2, 2004.

53. **Pradhan, S., et al.** "Unrestricted coreference: Identifying entities and events in OntoNotes", Proceedings of ICSC 2007, 2007, pp. 446–453.

54. **Bamman, D., Lewke, O., Mansoor, A.** (2019). An annotated dataset of coreference in English literature. *arXiv preprint arXiv:1912.01140*.

55. **Eirew, A., Cattan, A., Dagan, I.** (2021). WEC: Deriving a large-scale cross-document event coreference dataset from Wikipedia. *arXiv preprint arXiv:2104.05022*.

56. **Rudinger, R., Naradowsky, J., Leonard, B., Van Durme, B.** (2018). Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*.

57. **Schmolz, H., Coquil, D., Döller, M.** In-depth analysis of anaphora resolution requirements. In: *2012 23rd International Workshop on Database and Expert Systems Applications*. IEEE, 2012. p. 174-179.

58. **Gupta, A., et al.** (2022). ezCoref: Towards Unifying Annotation Guidelines for Coreference Resolution. *arXiv preprint arXiv:2210.07188*.

59. **Indurkhya, N., Damerau, F. J.** (Eds.). (2010). *Handbook of natural language processing* (Vol. 2). CRC Press.

60. **Yörük, E., Hürriyetoğlu, A., Duruşan, F.,Yoltar, Ç**. (2022). Random sampling in corpus design: Cross-context generalizability in automated multicountry protest event collection. *American Behavioral Scientist*, *66*(5), 578-602.

61. **Brezina, V., Hawtin, A., McEnery, T.** (2021). The written british national corpus 2014–design and comparability. *Text & Talk*, *41*(5-6), 595-615.

62. **Zeldes, A.** (2022). Opinion Piece: Can we Fix the Scope for Coreference? Problems and Solutions for Benchmarks beyond OntoNotes. *Dialogue & Discourse*, *13*(1), 41-62.

63. **Byron, D. K.** The Uncommon Denominator: A proposal for Consisten Reporting of Pronoun Resolution Results. In *Computational Linguistics*, Volume 27, Issue 4, 2001, pp. 569-578.

64. **Vilain, M.,** et al. A model-theoretic coreference scoring scheme. In: *Proceedings of the 6th conference on Message understanding*. Association for Computational Linguistics, 1995. p. 45-52.

65. **Bagga, A., Baldwin, B.** Algorithms for scoring coreference chains. In: *The first international conference on language resources and evaluation workshop on linguistics coreference*. 1998. p. 563-566.

66. **Xiaoqiang L.** 2005. On coreference resolution performance metrics. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (HLT '05). Association for Computational Linguistics, Stroudsburg, PA, USA, 25-32.

67. **Pradhan, S., et al.** CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In: *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics, 2012. p. 1-40.

68. **Denis, P., Baldridge, J**. Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 2009, 42.

69. **Moosavi, N. S., Strube, M.** Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

2016. p. 632-642.

70. **LUO, X., et al.** An extension of BLANC to system mentions. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. NIH Public Access, 2014. p. 24.
71. **Pradhan, S., et al.** Scoring coreference partitions of predicted mentions: A reference implementation. In: *Proceedings of the conference. Association for Computational Linguistics. Meeting*. NIH Public Access, 2014. p. 30.
72. **Chen, C., Ng, V.** Linguistically aware coreference evaluation metrics. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. 2013. p. 1366-1374.
73. **Holen GI**. Critical reflections on evaluation practices in coreference resolution. In: *Proceedings of the 2013 NAACL HLT Student Research Workshop*. 2013, pp. 1–7.
74. **Tuggener D**. (2016) Incremental coreference resolution for German.
75. **Novák M**. (2018) Coreference from the Cross-lingual Perspective.
76. **Kaczmarek A, Marcińczuk M**. Evaluation of coreference resolution tools for Polish from the information extraction perspective. In: *The 5th Workshop on Balto-Slavic Natural Language Processing*. 2015, pp. 24–33.
77. **Kummerfeld, Jonathan K.; Klein, Dan**. Error-driven analysis of challenges in coreference resolution. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013. pp. 265-277.
78. **Manning C, Raghavan P**, **Schütze H**. Introduction to information retrieval. *Nat Lang Eng* 2010; 16: pp. 100–103.
79. **Mitkov R**. Process of automatic anaphora resolution, In *R. Mitkov. Anaphora resolution*, Routlege, USA, 2013, 28-52.
80. **Kamune, K. P., Agrawal, A.** Hybrid approach to pronominal anaphora resolution in English newspaper text. *International Journal of Intelligent Systems and Applications*, 2015, 7.2: 56.
81. **Hobbs, J. R.** Resolving Pronoun References. In *Readings in natural language processing*, B. Grosz, K. Sparck-Jones, and B. Webber, editors, USA, 1986, pp. 339-352.
82. **Kibble, R.** A Reformulation of Rule 2 of Centering Theory. In *Computational Linguistics*, Volume 27, Issue 4, December 2001, pp. 579-587.
83. **Brennan, S. E., Friedman, M. W., Pollard, C. J.** A Centering Approach To Pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA, 1987, pp. 155-162.
84. **Tetreault, J. R.** A Corpus-Based Evaluation of Centering and Pronoun Resolution. In *Computational Linguistics*, Volume 27, Issue 4, December 2001, pp. 507-520.
85. **Lappin, S., Leass, H. J.** An Algorithm for Pronominal Anaphora Resolution. In *Computational Linguistics*, Volume 20, Issue 4, December 1994, pp. 535-561.
86. **Cohen A.** Anaphora resolution and minimal models, *Proceedings of the Fifth International Workshop on Inference in Computational Semantics*, 2006.
87. **Cohen A.** Anaphora Resolution as Equality by Default, *Anaphora: Analysis, Algorithms and Applications Lecture Notes in Computer Science Volume 4410*, 2007, pp 44-58.
88. **Fischer, W.** *Linguistically Motivated Ontology-Based Information retrieval*, DE, February 2013.

89. **Balaji, J., Geetha, T. V., Parthasarathi, R. & Karky, M.** Anaphora Resolution in Tamil using Universal Networking Language. In *Proceedings of the Indian International Conference on Artificial Intelligence (IICAI-2011)*, Karnataka, India, 2011.

90. **Zeldes A, Zhang S**. When annotation schemes change rules help: A configurable approach to coreference resolution beyond ontonotes. In: Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016), pp 92–101.

91. **Fonseca, E., Vanin, A., Vieira, R.** Nominal Coreference Resolution Using Semantic Knowledge. In: *International Conference on Computational Processing of the Portuguese Language*. Springer, Cham, 2018. p. 37-45.

92. **Oliveira, H. G., Gomes, P.** ECO and Onto. PT: a flexible approach for creating a Portuguese wordnet automatically. *Language resources and evaluation*, 2014, 48.2: 373-393.

93. **Dagan, I., et al.** Syntax and lexical statistics in anaphora resolution. *Applied Artificial Intelligence an International Journal*, 1995, 9.6: 633-644.

94. **Ge, N., Hale, J., Charniak, E.** A Statistical Approach to Anaphora Resolution. In *Proceedings of the Sixth Workshop of Very Large Corpora*, 1998, pp. 161-170.

95. **Bies, Ann, Justin M., Colin W.** English News Text Treebank: Penn Treebank Revised LDC2015T13. Web Download. Philadelphia: Linguistic Data Consortium, 2015.

96. **Mitkov, R**. (1998). Robust pronoun resolution with limited knowledge. COLING-ACL 1998, 869-875.

97. **Mitkov R, Evans R, Orasan C** (2002) A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, pp 168-186.

98. **Soon, W. M., Ng, H. T., Lim, D. C. Y.** A Machine Learning Approach to Coreference Resolution of Noun Phrases. In *Computational Linguistics*, Volume 27, Issue 4, December 2001, pp. 521-544.

99. **Ng, V., Cardie, C.** Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA, July 2002, pp. 104-111.

100. **Finkel JR, Manning** CD (2008) Enforcing transitivity in coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Association for Computational Linguistics, pp 45-48.

101. **Fernandes, E. R., Dos Santos, C. N., Milidiú, R L.** Latent structure perceptron with feature induction for unrestricted coreference resolution. In: *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics, 2012. p. 41-48.

102. Bonial, C., et al. "Current directions in english and arabic propbank." *Handbook of linguistic annotation* (2017): 737-769.

103. **Schuler, K. K.** *Verbnet: A Broad-Coverage, Comprehensive Verb Lexicon*, USA, 2005.

104. **Ruppenhofer, J., et al.** FrameNet II: Extended Theory and Practice. 2006.

105. **Matsubayashi, Y., Miyao, Y., Aizawa, A.** Framework of Semantic Role Assignment based on Extended Lexical Conceptual Structure: Comparison with VerbNet and FrameNet. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational* Linguistics, Avignon, France, April 23-27 2012, pp. 686-695.

106. **Ponzetto, S. P., Strube, M.** Semantic role labeling for coreference resolution. *Demonstrations*, 2006.

107. **Pradhan, S. S., et al.** Shallow semantic parsing using support vector machines. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. 2004.

108. **Veena, G., et al.** A learning method for coreference resolution using semantic role labeling features. In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017. p. 67-72.

109. **Lin, CH., et al.** Event-based textual document retrieval by using semantic role labeling and coreference resolution. In: *IADIS International Conference WWW/Internet 2007*. 2007.

110. **Cunningham, H., et al.** "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics." *PLoS computational biology* 9.2 (2017): e1002854.

111. **Wiseman, Sam, Alexander M. Rush, and Stuart M. Shieber.** "Learning global features for coreference resolution." arXiv preprint arXiv:1604.03035, 2016.

112. **Lee, K., et al.** End-to-end Neural Coreference Resolution. arXiv preprint arXiv:1707.07045, 2017.

113. **Zhang, H., et al.** "Knowledge-aware pronoun coreference resolution." *arXiv preprint arXiv:1907.03663* (2019).

114. **Lai, T., et al.** "A context-dependent gated module for incorporating symbolic semantics into event coreference resolution." *arXiv preprint arXiv:2104.01697* (2021).

115. **Yang, Xiaohan, et al**. What gpt knows about who is who. arXiv preprint arXiv:2205.07407, 2022.

116. **Le, Nghia T.; Ritter, Alan**. Are Large Language Models Robust Zero-shot Coreference Resolvers?. arXiv preprint arXiv:2305.14489, 2023.

117. **Liu, R., Mao, R., Luu, A. T., Cambria, E**. (2023). A brief survey on recent advances in coreference resolution. *Artificial Intelligence Review*, 1-43.

118. **Mitkov, R., Lappin, S., Boguraev, B**.: Introduction to the Special Issue on Computational Anaphora Resolution. Computational Linguistics 27(4), 473–477 (2001).

119. **Znotins, A., Paikens, P.** Coreference resolution for latvian. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014). 2014.

120. **Ogrodniczuk, M., Kopec, M.** Rule-based coreference resolution module for Polish. In: *Proceedings of the 8th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2011)*. 2011. p. 191-200.

121. **Versley, Y., et al**. BART: A modular toolkit for coreference resolution. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*. Association for Computational Linguistics, 2008. p. 9-12.

122. **Kopec, M., Ogrodniczuk, M.** Creating a Coreference Resolution System for Polish. In: LREC. 2012. p. 192-195.

123. **Nitoń, B., Morawiecki, P., Ogrodniczuk, M.** Deep neural networks for coreference resolution for Polish. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. 2018..

124. **Nitoń, B., Ogrodniczuk, M.** Multi-pass Sieve Coreference Resolution System for Polish. In: *International Conference on Language, Data and Knowledge*. Springer,

Cham, 2017. p. 222-236. (2017).

125. **Toldova, S., et al.** Error analysis for anaphora resolution in Russian: new challenging issues for anaphora resolution task in a morphologically rich language. In: *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016)*. 2016. p. 74-83.
126. **Khadzhiiskaia, A., Sysoev, A.** Coreference resolution for Russian: taking stock and moving forward. In: *2017 Ivannikov ISPRAS Open Conference (ISPRAS)*. IEEE, 2017. p. 70-75.
127. **Budnikov, A. E., Toldova, S. Y., et al.** (2019). Ru-eval-2019: Evaluating anaphora and coreference resolution for russian. Computational Linguistics and Intellectual Technologies-Supplementary Volume.
128. **Hajičová, E., Kuboň, V., Kuboň, P.** Stock of shared knowledge: A tool for solving pronominal anaphora. In: *Proceedings of the 14th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1992. p. 127-133.
129. **Kučová, L., Žabokrtský, Z.** Anaphora in czech: Large data and experiments with automatic anaphora resolution. In: *International Conference on Text, Speech and Dialogue*. Springer, Berlin, Heidelberg, 2005. p. 93-98.
130. **Linh, N. G., et al.** Comparison of classification and ranking approaches to pronominal anaphora resolution in Czech. In: *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2009. p. 276-285.
131. **Novák, M.** Coreference resolution system not only for Czech. In: Proceedings of the 17th conference ITAT 2017: Slovenskoceský NLP Workshop (SloNLP 2017). 2017. "Coreference Resolution System Not Only for Czech." (2017).
132. **Novák, M., Nedoluzhko, A., Žabokrtský, Z.** Projection-based coreference resolution using deep syntax. In: Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017). 2017. p. 56-64. Projection-based Coreference Resolution Using Deep Syntax. 56-64.
133. **Pražák, O., Miloslav K., and Jakub S.** "Multilingual coreference resolution with harmonized annotations." *arXiv preprint arXiv:2107.12088* (2021).
134. **Žitkus**, **V**.; **Nemuraitė**, **L**. Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus. *Informacinės technologijos (IVUS 2014), Kaunas, Technologija*, 2014, 177-182.
135. **Žitkus, V., Butkiene, R.** Coreference Annotation Scheme and Corpus for Lithuanian Language. In: *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2018. p. 243-250.
136. **Žitkus, V**., **Butkienė, R**., & **Butleris, R**. (2023). Linguistically aware evaluation of coreference resolution from the perspective of higher-level applications. *Natural Language Engineering*, 1-30.
137. **Lee, H., et al.** Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the 15th conference on computational natural language learning: Shared task*, Association for Computational Linguistics. 2011, pp. 28–34.
138. **Žitkus, V.** 2018, Lithuanian Coreference Corpus, CLARIN-LT digital library in the Republic of Lithuania, http://hdl.handle.net/20.500.11821/19 ,[Accessed 3 June 2019]
139. **Žitkus, V., Nemuraitė, L**. (2015). Automated Anaphora and Co-reference Resolution

for Lithuanian Language Combining Results from Different Text Analysis Stages. In *BIR Workshops* (pp. 164-172).

140. **Žitkus, V., Nemuraitė, L.** First Steps in Automatic Anaphora Resolution in Lithuanian Language Based on Morphological Annotations and Named Entity Recognition. In: *International Conference on Information and Software Technologies*. Springer, Cham, 2015. p. 480-490.
141. **Žitkus, V., et al.** Minimalistic Approach to Coreference Resolution in Lithuanian Medical Records. *Computational and Mathematical Methods in Medicine*, 2019, 2019.
142. **W3C.** OWL 2 Web Ontology Language Document Overview (Second Edition). In *W3C Recommendation 11 December 2012*. Available from: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/, [Accessed 6 January 2023].
143. **Kutuzov, A., Ionov, M.** The impact of morphology processing quality on automated anaphora resolution for Russian. *Computational Linguistics and Intellectual Technologies*, 2014, 13.
144. **Leonavičienė, A., et al**. Sakinių ilgis–publicistinio ir šnekamojo stiliaus sandūros tekstuose požymis. *Kalbotyra*, 2010, 62 (3): 95-107.
145. **Stukaitė, A.** Publicistinio Stiliaus Diferenciacija: Magistro Darbas. Vilnius: Lietuvos Edukologijos Universitetas. Prieiga per ELABa – Nacionalinė Lietuvos Akademinė Elektroninė Biblioteka, 2005. Web.
146. **Lehmann, J., et al.** DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 2015, 6.2: pp. 167-195.
147. **Antanas, L.** Sinonimų žodynas. *Vilnius: Lietuvių kalbos institutas*, 2002. URL = https://www.raštija.lt/apie/i%C5%A1tekliai/%C5%BEodynai/6189?did=1, [Accessed 3 June 2019].
148. **Zagorskytė, A.** *Semantinio Vaidmenų žodyno Sudarymo Informacinė Sistema: Bakalauro Darbas*. Kaunas: Kauno Technologijos Universitetas. Prieiga per ELABa – Nacionalinė Lietuvos Akademinė Elektroninė Biblioteka, 2016.
149. **Artstein, R., Poesio, M.** "Bias decreeases in proportion to the number of annotators", Proceedings of FG-MoL, 2005, pp. 141–150.
150. **Artstein, R., Poesio, M.** "Inter-coder agreement for computational linguistics", Computational Linguistics, 2008, 34(4), 555–596.
151. **Bates, S., Hastie, T., & Tibshirani, R**. (2023). Cross-validation: what does it estimate and how well does it do it?. *Journal of the American Statistical Association*, 1-12.
152. **Ying, X.** (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.

LIST OF AUTHOR'S PUBLICATIONS ON DISSERTATION THEME

Scientific articles in periodicals

1. **Žitkus, Voldemaras**; Butkienė, Rita; Butleris, Rimantas. Linguistically aware evaluation of coreference resolution from the perspective of higher-level applications // Natural language engineering. Cambridge : Cambridge university press. ISSN 1351-3249. eISSN 1469-8110. 2023, Early access, p. 1-30. DOI: 10.1017/S1351324923000293.
2. **Žitkus, Voldemaras**; Butkienė, Rita; Butleris, Rimantas; Maskeliūnas, Rytis; Damaševičius, Robertas; Wózniak, Marcin. Minimalistic approach to coreference resolution in Lithuanian medical records // Computational and mathematical methods in medicine. London : Hindawi. ISSN 1748-670X. eISSN 1748-6718. 2019, vol. 2019, art. no. 9079840, p. 1-14. DOI: 10.1155/2019/9079840.

Scientific articles in conference proceedings

1. **Žitkus, Voldemaras**; Butkienė, Rita. Coreference annotation scheme and corpus for Lithuanian language // 2018 fifth international conference on social networks analysis, management and security (SNAMS), Valencia, Spain, 15-18 October 2018. Piscataway, NJ : IEEE, 2018. ISBN 9781538695890. eISBN 9781538695883. p. 245-250. DOI: 10.1109/SNAMS.2018.8554892.
2. **Žitkus, Voldemaras**; Nemuraitė, Lina. First steps in automatic anaphora resolution in Lithuanian language based on morphological annotations and named entity recognition // Information and software technologies: proceedings of the 21st international conference, ICIST 2015, Druskininkai, Lithuania, October 15-16, 2015 / G. Dregvaite, R. Damasevicius (eds.). Cham : Springer, 2015. ISBN 9783319247694. eISBN 9783319247700. p. 480-490. (Communications in computer and information science, ISSN 1865-0929, eISSN 1865-0937 ; Vol. 538). DOI: 10.1007/978-3-319-24770-0_41.
3. **Žitkus, Voldemaras**; Nemuraitė, Lina. Automated anaphora and co-reference resolution for Lithuanian language combining results from different text analysis stages // CEUR workshop proceedings : joint BIR 2015 workshops and doctoral consortium, BIR-WS 2015, co-located with 14th international conference on perspectives in business informatics research, BIR 2015, Tartu, Estonia, August 26-28, 2015 / edited by: Raimundas Matulevičius, Fabrizio Maria Maggi, Peep Küngas. Aachen : CEUR-WS. ISSN 1613-0073. 2015, vol. 1420, p. 164-172. Prieiga per internetą: <http://ceur-ws.org/Vol-1420/dc-paper3.pdf> [žiūrėta 2017-02-25].
4. **Žitkus, Voldemaras**; Nemuraitė, Lina. Taxonomy of anaphoric expressions as a starting point for anaphora resolution in Lithuanian corpus // Informacinės technologijos : 19-oji tarpuniversitetinės magistrantų ir doktorantų konferencija "Informacinė visuomenė ir universitetinės studijos" (IVUS 2014) : konferencijos pranešimų medžiaga / Kauno technologijos universitetas, Vytauto Didžiojo universitetas, Vilniaus universiteto Kauno humanitarinis fakultetas. Kaunas : Technologija. ISSN 2029-249X. eISSN 2029-4832. 2014, p. 177-182.

CURRICULUM VITAE

Voldemaras Žitkus, voldemaras.zitkus@ktu.lt

Išsilavinimas:

2013 – 2019. Informatikos inžinerijos mokslo krypties doktorantūros studijos KTU Informatikos fakultete.

2011 – 2013. Informacinių sistemų inžinerijos magistro studijos KTU Informatikos fakultete. Įgytas informacinių sistemų inžinerijos magistro kvalifikacinis laipsnis.

2007 – 2011. Bakalauro studijos KTU Informatikos fakultete. Įgytas informatikos bakalauro kvalifikacinis laipsnis.

Profesinė patirtis:

2013 – 2016. Analitikas programuotojas. KTU, Informacinių sistemų projektavimo technologijų centras.

2016 – . Lektorius. KTU, Informacijos sistemų katedra.

Mokslinių interesų sritys:

Natūralios kalbos apdorojimas, anaforų ir koreferencijų sprendimas, ontologijos, semantinės technologijos, duomenų bazės, informacinių sistemų projektavimas ir dalykinės srities modeliavimas.

UDK: 81'322.2+004.934](043.3)
SL344. 20xx-xx-xx, xx leidyb. apsk. l. Tiražas 14 egz. Užsakymas 71.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas