Enhancing Ontology Matching: Lexically and Syntactically Standardizing Ontologies Through Customized Lexical Analyzers

Jomar Silva [a,*], Kate Revoredo [b], Fernanda Araujo Baião [c] and Cabral Lima [a]
[a] *Graduate Program in Informatics, Federal University of Rio de Janeiro (UFRJ), RJ, Brazil*
E-mails: jomar.silva@ufrj.br, cabrallima@ufrj.br
[b] *Humboldt-Universität zu Berlin, Germany*
E-mail: kate.revoredo@hu-berlin.de
[c] *Department of Industrial Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), RJ, Brazil*
E-mail: fbaiao@puc-rio.br

Abstract. Ontology matching systems commonly leverage linguistic metrics to establish mappings between entities within the ontologies undergoing alignment. However, due to the absence of standardized entity names across these ontologies, such metrics may lead to some correct mappings not to be selected. Existing methodologies, which focus on standardizing entity names, often do so without considering the ongoing matching, potentially resulting in inaccurate outcomes. These tools also, in general, are not concerned with the syntactic standardization of entity names. To address this issue, in this paper, we introduce a novel approach to standardize both lexically and syntactically the entity names through the development of a customized lexical analyzer tailored to the aligned ontologies. We evaluate the efficacy of this approach using ALIN, an interactive ontology matching system, along with the human and mouse ontologies from the Anatomy track of the OAEI. Our findings demonstrate an improvement in the quality of the alignment results.

Keywords: ontology matching, interactive ontology matching, ontology alignment, interactive ontology alignment , lexical analyzer, syntactic analyzer, linguistic tools, grammar, Wordnet, FMA

1. Introduction

Ontology matching is the task of searching for the best set of correspondences (mappings) between entities of different ontologies [1].

A mapping asserts a semantic relationship between the ontology entities, such as *disjunction*, *subsumption*, or *equivalence*. The ontology matching system is responsible for finding, among all the possible mappings, the ones to be included in the alignment.

Many ontology matching tools use similarity metrics to select mappings. The tools choose the mappings to be directly placed into the final alignment or to present to the expert in the case of interactive ontology matching tools. Before calculating the similarity metric, many ontology matching tools standardize the entity names. The lack of standardization in entity names can result in lower values than if the entity names were standardized, leading to the

*Corresponding author. E-mail: kate.revoredo@hu-berlin.de.

tool failing to select some correct mappings. Many ontology matching tools tackle this issue through a preprocessing step, where the entity names are standardized, enabling a more effective comparison (see section 2). In this step, various adjustments can be made, such as converting all characters to lowercase, splitting compound words, removing underscores, and so forth.

These standardizations, in general, do not take into account the ongoing matching. That can lead to an incorrect value in the similarity metric. For example, a tool may implement a standardization technique during preprocessing that removes all symbols except letters and numbers from the entity names. After preprocessing, an entity name 'head/neck' would be transformed into 'head neck'. If this name were compared to 'Head and Neck' in the other ontology, it would return a value below the maximum. Assuming that, for this similarity metric, the maximum value occurs when the two strings are identical. If someone had conducted a previous study on ontology names and discovered that the slash indicates 'and', they could implement the preprocessing to return 'head and neck'. If the tool also converted all letters to lowercase during preprocessing, the comparison would be between 'head and neck' and 'head and neck', and the maximum value would be reached. That would increase the likelihood of the tool chose this mapping. In the previous way, the probability of the tool discard it would increase.

Another aspect generally not considered is the syntactical standardization of entity names. This standardization can allow us to use techniques that would not be possible without this standardization.

This research proposes a method to standardize both lexically and syntactically the entity names of the ontologies through the customized generation of a lexical analyzer for each ontology, taking into account the ongoing matching process. The primary goal of this lexical analyzer is to standardize entity names. A secondary objective of the lexical analyzers is to ensure that entity names adhere to a context-free grammar that we have developed. To accomplish these objectives, the lexical analyzer is capable of either modifying entity names or rejecting entities.

To evaluate the benefit of our proposal we considered the interactive ontology alignment system ALIN[2]. For that, we have created a new semantic metric, henceforth called *the* ALIN *metric*, to leverage the characteristics of the context-free grammar. We will match the mouse and human ontologies from the Anatomy track of the OAEI.

The hypothesis we will verify in this paper is that *the lexical and syntactic standardization, in accordance with the grammar proposed, improves the quality of the resulting alignment.*

The contributions of this paper are summarized as follows:

- taking ongoing matching into consideration to standardize entity names lexically and syntactically. We achieve this through custom lexical analyzers designed for the matching process;
- the availability of the lexical analyzer as separate files. That allows other ontology matching tools reuse them;
- the development of an iterative process for creating the lexical analyzer. Each iteration in the lexical analyzer construction process generates a new version of the lexical analyzer. Each new version builds upon its predecessor, to generate alignments of increasingly higher quality;
- the creation of a context-free grammar that allows for the use of techniques that improve alignment;
- the creation of the ALIN metric, which considers grammar to calculate the similarity value between two entity names;
- The use of string-based metrics that take grammar into account to enhance their ability to find correct mappings.

The rest of this paper is organized as follows. Section 2, we present the related works. In Section 3, we explain some concepts related to the creation of the lexical analyzer and show the process for creating it. In Section 4, we explain which ontologies we will use to test our hypothesis and the tool we will use, the ALIN. In this section, we also explain how we conducted the test and what its results were. In Section 5, we present the research conclusion and potential future works. In Appendix I, we offer a step-by-step guide detailing the creation process of the lexical analyzers for both the human and mouse ontologies. In Appendix II, we present the first version of the lexical analyzers and instructions on invoking them within a Java program. We can found the other versions of the lexical analyzer on OSF[1]. Finally, in Appendix III, we show how to calculate the value of the ALIN metric.

[1]https://osf.io/gc6jm/?view_only=d2d0aac33fb046908aa878b8ff05be73

2. Related Work

In this section, we review the works related to ours (i.e., ontology matching techniques that address the issue of lack of standardization of the entity names in the ontologies). Many approaches address this issue with a preprocessing step in which they modify the entity names towards a standardization that allows for a better comparison between entities.

Several tools employ general techniques for standardizing entity names, often independent from the ongoing matching process. Among these techniques, we can enumerate:

Normalization by uppercase or lowercase. In [3][4][5][6][7][8][9][10][11][12][13] the entities name are converted to either lower case or upper case before making comparisons. Doing this normalization is important. For example, the entity names 'Head and Neck' and 'head and neck', when compared using a string-based metric, return a lower value than if both were converted to lowercase. In the second case, we would compare 'head and neck' and 'head and neck', which would return a higher value when using the metric.

Stemming Another process used in various alignment tools[14][4][15][16][17][18][19][20][21] [22][23][24] [25][26][27][23][28][29][30][31][32][33][34][35][36][37][38][39][12][40] during the pre-processing phase is stemming. This technique reduces the words to their base form. For example, the tool changes a plural word to its singular form or a conjugated verb to its infinitive form.

Separation of the words. Not all ontologies use the same notation for separating words in the entity name. Commonly adopted notations include camel case, separating words with underscores, and with spaces. Some ontologies utilize one notation for one entity name and a different notation for another entity name. Multiple tools [18][41][42][43][44][45][46] [47][9][48][40][12][27] analyze the word to be compared and choose the correct method for separation. That allows for the standardization of notations, for instance, converting all entity names to the notation separated by underscores. In addition to words, the term can include punctuation marks such as commas, parentheses, and more. The term may also contain other symbols. Therefore, besides separating the words within the term, the tool must account these additional symbols. From this separation of terms and symbols, the tools can perform tasks such as removal of punctuation characters, removal of stop words, expansion, stemming, and more.

Conversion of Roman numbers to Arabic numbers. The method described in [49] converts Roman numerals to Arabic numerals.

Removal of stop words. Stop words are words such as articles, prepositions, pronouns, conjunctions, that are removed given that they do not add much information to the text (e.g., stop words in English 'the', 'a', 'an', 'so', 'what', "and", "or"). In [50][51][52][16][53] [54][55][22][23][44][56][57][24][58][59] [60] [25][61][26][62] [5][63] [17] [64][65] [29][30][66][31][32][33][67][68][34][69][35][36][37][47][38][70][71] [72][73][74][75][76][77][40][78] stop words are removed from entity names before comparing them.

Removal of non-alphanumeric characters or their replacement with white spaces. Tools like [79][80][81] [48][74][40] remove non-alphanumeric characters or replace then with white spaces.

Removal of punctuation characters. Tools as [4][24][13][82][83][84][8][85][69][35][70] [86][9][86][87] remove punctuation caracters from the entity names.

Use of synonyms. In addition to the entity's own name, in [83][69][9][39][20][55] synonyms are used to search for matches in another ontology. These synonyms can be in the ontology (an ontology can associate two or more entity names as synonyms) or in external resources, such as WordNet, as used by [87][88][89], or FMA, as used by [90].

Expansion of abbreviations and acronyms. Tools as [70][39][87][55] expand abbreviations and acronyms..

Permutation of words. In [80] permutation of the words in the name are used. This technique compares the original name and its permutations with the names and their permutations from the other ontology.

A characteristic of applying these techniques in ontology matching tools is that the development team generally incorporates them into their code.

Our proposal aims to standardize the entity names, both at the lexical and syntactic levels, taking into account the ongoing matching process. We propose a grammar that defines the syntax to which all entity names should adhere after standardization. This grammar possesses characteristics that contribute to improving the quality of the generated alignment. We propose to accomplish this standardization through the use of customized lexical analyzers for the ongoing matching. These lexical analyzers will be separate files from the code, allowing other tools to reuse them for the same matching process.

3. Lexical analyzer creation process

3.1. Preliminaries

In this subsection, we will explain several concepts related to the lexical analyzer creation process.

Definition 3.1 (Standardization Technique)**.** *It is a technique employed by ontology matching tools to standardize entity names. Examples of such techniques include those we have seen in the section 2, such as Normalization of the name to lower case or upper case, Stemming, and others.*

To understand the lexical analyzer creation process, we need to know that the process will generate multiple versions of the lexical analyzer. Each new version of the lexical analyzer will incorporate all the standardization techniques (Definition 3.1) from the previous version and possibly some additional ones.

3.1.1. Regular expression and lexical analyzer

Definition 3.2 (Lexeme, token and word)**.** *The token name is an abstract symbol representing a kind of lexical unit, e.g., a particular keyword, or a sequence of input characters denoting an identifier. The token names are the input symbols that the parser (syntactic analyzer) processes[91]. We can see tokens as the categories into which the lexical analyzer can classify the lexemes.*

A lexeme is a sequence of characters in the source program that matches the pattern for a token and is identified by the lexical analyzer as an instance of that token[91]. We will refer to lexemes that are English words as 'words'.

According to Wikipedia, a regular expression (shortened as regex or regexp), sometimes referred to as rational expression, is a sequence of characters that specifies a match pattern in text. Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation. We can group various regex patterns and form a lexical analyzer.

A lexical analyzer generator takes as input a specification with a set of regular expressions and corresponding actions. It generates a program (a lexer) that reads input, matches the input against the regular expressions, and runs the corresponding action if a regular expression matched. Lexers usually are the first front-end step in compilers, matching keywords, comments, operators, etc, and generating an input token stream for the syntactic analyzer.

We can program the code associated with the matched regular expression to return a variant of the name that triggered the regular expression. If a non-standard entity name appears, we can modify the lexical analyzer to recognize this name and return a standardized name according to a standardization technique. We use these characteristics to generate standardized names in the two ontologies involved in the matching.

Definition 3.3 (Existing tokens in the target grammar)**.** *The lexical analyzers will recognize the tokens below and pass them to the syntactic analyzer:*

- *Noun or modifier - Lexemes composed solely of letters, with the first letter possibly being uppercase;*
- *Numeral - Lexemes composed only by numbers;*
- *Preposition - Lexemes that are an English preposition, such as 'of', 'for', 'at', etc.*

It's important to note that these tokens apply to the first version of the lexical analyzers. The lexemes that can be instances of a token may change with each version of the lexical analyzer. For example, in a future version, the 'noun or modifier' token may accept terms with hyphens.

Let's take the following entity name as an example:

3rd ventricle choroid plexus

It has four words. However, when the lexical analyzer, implementing the three tokens (Definition 3.3), reads this string, it doesn't output these four words. It will generate five strings as follows: 3, rd, ventricle, choroid, plexus.

As per our definition that a noun or modifier consists only of letters, the lexical analyzer outputs the last four strings, while the first string is returned based on the numeral definition.

The tokens it will pass to the syntactic analyzer will be (numeral,noun or modifier,noun or modifier,noun or modifier,noun or modifier). The five generated strings (3, rd, ventricle, choroid, plexus) are lexemes. It's worth noting that a lexeme may not necessarily align with a single word directly. The word '3rd' resulted in two distinct lexemes. Hence, the lexical analyzer categorizes lexemes, not necessarily words, into the three tokens.

One of the goals of the ultimate version of the implemented lexical analyzers is for each lexeme to be a word.

3.1.2. Syntactic analyzer

Definition 3.4 (Noun Phrase)**.** *A noun phrase[92] is a phrase made up of a head noun that can be modified with words before it (pre-modifiers), such as adjectives and nouns; or after it (post-modifiers), by, for example, prepositional phrases.*

Pre-modifiers:

- *academic language (adjective + noun)*
- *email address (noun + noun)*

Post-modifiers:

- *articles by leading academics (noun + prepositional phrase)*

If we only consider the noun phrases shown above, when there isn't a preposition, its head noun is the last word. When there is a preposition, it is the word before the preposition.

Definition 3.5 (Concept and terms)**.** *A concept is a mental category that helps us organize objects. In a domain, a term is generally associated with a concept. For example, a category of objects in a classroom used for sitting, we associate with the term 'chair'. Terms refer to the same concept in a domain if they refer to the same objects. We can say that entity names are terms.*

Definition 3.6 (Nested terms)**.** *Nested terms are terms in their own right incorporated as part of larger terms. If the noun phrase 'accessory olfactory bulb' is a term in a domain and 'accessory olfactory bulb glomerular layer' is other term then 'accessory olfactory bulb' is a nested term in 'accessory olfactory bulb glomerular layer' (Table 1).*

Table 1
Terms and nested terms

Term	Nested term
accessory olfactory bulb glomerular layer (mouse ontology)	accessory olfactory bulb
Deep_Temporal_Vein (human ontology)	temporal vein

Definition 3.7 (Term Variation)**.** *When two terms in a domain refer to the same concept, we say they are variants of the same term. A way to obtain a variation of a term is by replacing an adjective with a modifier noun. We find some of these variations in human and mouse ontologies. This type of variation is called derivation. For instance, 'enzymatic activity' has the variation 'enzyme activity'. Another way to obtain a variation of the term is through permutation. Permutation can occur when a term contains prepositions. For example, 'activity of enzyme' can have the variation 'enzyme activity'.*

Syntax deals with the way that words are put together to form phrases, clauses, and sentences. For instance, let's consider that the terms used as entity names in an ontology consist of noun phrases, and we can classify the words in these noun phrases into the three tokens mentioned in Definition 3.3, as a noun or modifier, a preposition, or a numeral. A noun or modifier, a preposition, or a numeral can follow another noun or modifier. Only a noun or modifier can follow a preposition. Only a preposition or a noun or modifier can follow a numeral. A term with two consecutive numeral words would not conform to this syntax. Similarly, a term that begins with a numeral or a preposition would also not comply with this syntax. A term that includes symbols not falling into the three tokens above, such as a comma, would also not conform to the syntax.

We wrote, in Definition 3.8, a context-free grammar for the syntax described above.

Definition 3.8 (Target Grammar)**.** *Being q - noun or modifier, n - numeral and p - preposition, We refer to the following context free grammar:*

$$S \rightarrow q \mid qB \tag{1}$$

$$B \rightarrow pS \mid n \mid nC \mid q \mid qB \tag{2}$$

$$C \rightarrow pS \mid q \mid qB \tag{3}$$

as the Target Grammar.

We chose this grammar to represent the syntax we want for the entity names of the ontologies participating in the matching process. It seems simple to us and is an LL grammar. This grammar type is recognized for its human-friendly readability relative to other types and for its efficient implementation. There are other advantages of the target grammar. If an entity name follows the syntax defined by the target grammar, we can:

– Identify the head noun of the entity name. The head noun will always be a 'noun or modifier', typically the last word of the entity name, unless the last word is a 'numeral'. It may also not be the last word if there are prepositions in the entity name, which we can determine with the classification 'preposition'. Knowing the head noun can enable the implementation of useful strategies for finding mappings[92]. We leverage knowledge of the head noun in the ALIN metric.
– Generate variations of the entity name. We can generate two types of variations (Definition 3.7) for the entity name:

  * One of them involves replacing the adjectives of the entity name with nouns. To accomplish this, we search for words classified as 'noun or modifier' that are not the head noun and then query them in Wordnet. We retrieve all associated nouns if Wordnet classifies the word as an adjective and produces the variations.
  * Another possible variation is permutation, for which we need to know the words of the entity name that are prepositions.

  We utilize variants both in the ALIN metric and to improve string-based metrics.

Syntatic Analysis involves scrutinizing whether a given input string conforms to the rules and structure of a formal grammar. In a syntactic analyzer, when the input string matches with one of the grammar rules, we can execute an algorithm. We developed the Syntactic Analyzer for the target grammar. All entity names must comply with the target grammar in the latest version of the lexical analyzer.

Definition 3.9 (Entity with standard name)**.** *An entity has a standardized name when the lexical analyzer can classify all lexemes found in the entity name into one of the three tokens described in Definition 3.3. An entity with a non-standard name is one that has a lexeme that the lexical analyzer cannot classify into one of the three tokens defined in Definition 3.3, such as when the name contains a comma or other punctuation marks. It is important to emphasize that with each version of the lexical analyzer, we can modify a token to include new lexemes.*

Definition 3.10 (Entity with syntactically correct name)**.** *It is an entity with a standard name (Definition 3.9) whose name adheres to the target grammar (Definition 3.8). An entity with a non-syntactically correct name is one that does not follow the target grammar, for example, when the lexical analyzer classifies the first lexeme of the term as a numeral.*

Definition 3.11 (Rejected entity)**.** *It is an entity flagged as invalid for comparison by the lexical analyzer. When this happens, we no longer consider the entity as part of the entities eligible for inclusion in a mapping in the alignment. That can occur in the following situations:*

– *The entity is an entity with a non-standard name, but it has an entity with the standard name associated with it, for example, as a synonym.*
– *The entity is an entity with a non-standard name, and after conducting searches in the other ontology, we observe that it has a low probability of being included in a mapping.*

Throughout lexical analysis, the lexical analyzers can modify names or reject entities with non-standard or non-syntactically correct name.

3.2. Lexical analyzer creation process

The integration in a lexical analyzer of a standardization technique (Definition 3.1) consider the ontologies involved in matching. Consequently, we assess techniques like the Removal of non-alphanumeric characters. If we deem it necessary to retain any non-alphanumeric character used in the ontologies, we will preserve it rather than remove it.

We have developed a series of programs to assist the NLP expert in generating a new version of the lexical analyzer. These programs are specific to the standardization techniques we wish to implement in one of the lexical analyzers. For instance, if our intention is to remove non-alphanumeric characters, yet upon inspection, we discover that both ontologies in question employ hyphens, which we consider to be meaningful characters deserving retention. But we've noticed that not all words with hyphens in one ontology have hyphens in the other. For instance, the word 'hind-brain' may be written in the other ontology as 'hindbrain', 'hind brain', or 'hind_brain'. We've developed a program specifically to address these hyphenation inconsistencies. This program reads the involved ontologies and furnishes us with the necessary lines for the lexical analyzer to manage all these hyphenation variations. At the end of the program, we need to copy the lines and paste it into the lexical analyzer, and from there, we can use this new version of the lexical analyzer.

To create a new version of the lexical analyzer, the NLP expert must:

1. Run the ontology matching tool using the lexical analyzer: the lexical analyzer will compile a list of all entities with non-standard (Definition 3.9) or non-syntactically correct names (Definition 3.10);
2. Select the standardization technique that he wants to insert into the lexical analyzer;
3. Analyze the list of step 1 and assess how to implement the standardization technique, considering the ontologies involved in the matching process;
4. Check to see if there is an existing program that could help modify the lexical analyzer. If available, execute the program. These programs can generate suggestions or directly generate lines for the lexical analyzer. We provide examples of how these programs function in Appendix I;
5. Adjust the lexical analyzer: The NLP expert will modify the lexical analyzer to include new lexemes in a token, modify entity names, or reject entities;

During the execution of the lexical analyzer, it will perform the following actions:

1. Classify lexemes into the tokens (Definition 3.3). There is a possibility for a lexeme to be modified before classification;
2. Return the tokens to the syntactic analyzer, and return the lexemes to the ontology matching tool;
3. Reject an entity if at least one lexeme is marked for rejection;
4. Generate a list of entities with non-standard or non-syntactically correct names.

The process of creating the lexical analyzer will be:

1. Create the initial lexical analyzer: In this first lexical analyzer, we will only categorize the lexemes (Definition 3.2) into the three tokens as described in the Definition 3.3;
2. Create a new version of the lexical analyzer (as explained earlier in this subsection);
3. Proceed to step 2 as long as there are entities with non-standard or non-syntactically correct names.

In Figure 1, we can observe the interaction between the lexical analyzer and the NLP Expert for the creation of a new version.

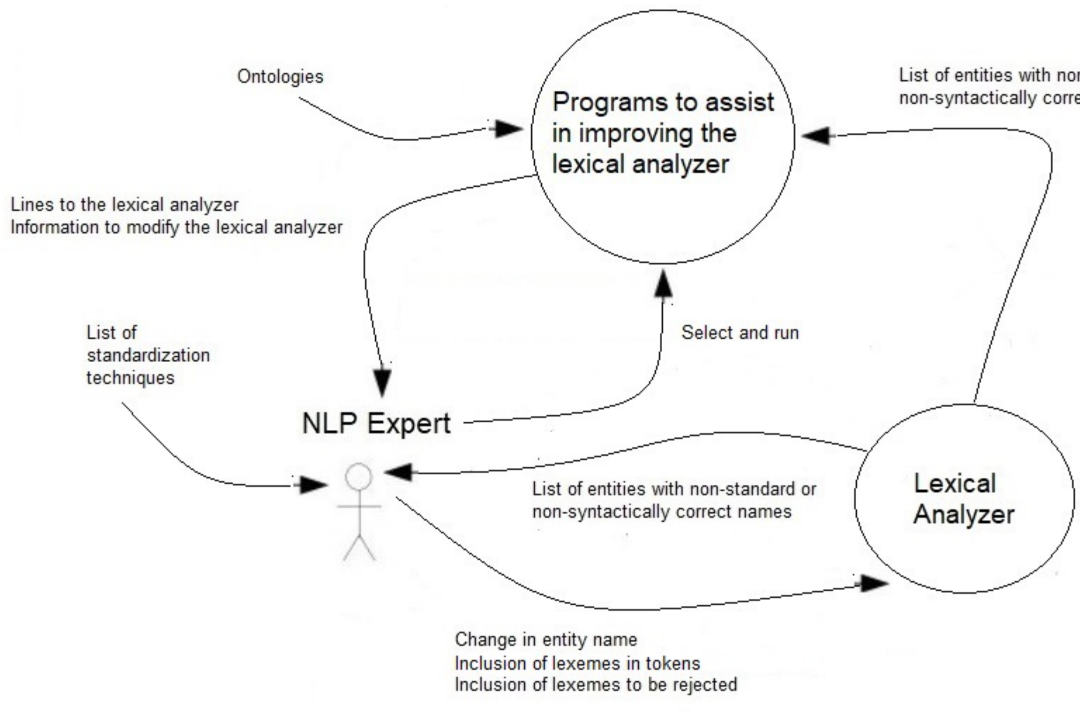

Fig. 1. Interaction between Lexical Analyzer and NLP Expert for the creation of a new version

4. Experimental Evaluation

In this section, we will evaluate our hypothesis. To conduct this analysis, we will use the ALIN[2] interactive ontology matching tool. We will use the human and mouse ontologies from the Anatomy track of the OAEI.

4.1. Anatomy track

The Ontology Alignment Evaluation Initiative (OAEI) is a coordinated international initiative whose one of the goals is to assess the strengths and weaknesses of ontology matching systems. The OAEI Interactive Matching Track [93] has two datasets: Conference and Anatomy. We conducted this test solely with the Anatomy dataset because it was the dataset for which we developed the lexical analyzers. The Anatomy track involves finding an alignment between an ontology describing the adult mouse anatomy and an ontology describing a part of the NCI Thesaurus (describing the human anatomy). The number of possible mappings between the two ontologies is 9,066,176. There are 1,516 mappings in the reference alignment.

4.2. ALIN *Overview*

ALIN is an interactive ontology matching tool that has participated in the OAEI interactive matching track since 2016. Since 2020, ALIN has been using the ALIN metric and lexical analyzers to standardize entity names. ALIN has participated in OAEI 2023 and was evaluated in the interactive matching track, with the Conference and Anatomy data sets. When compared with the other tools, ALIN got the highest F-Measure in both data sets[93]. Since ALIN's participation in OAEI 2023, we have automated part of the process of generating new versions of the lexical analyzers, resulting in better outcomes compared to OAEI 2023 in terms of the number of interactions with the expert.

ALIN is implemented in Java using the following APIs: Stanford CoreNLP API [94] to put a word in canonical form; Simmetrics API [95], for string-based similarity metrics; HESML API [96], for Wordnet [97] based linguistic metrics; the OWL API for handling ontologies written in OWL, the Alignment API [98] to deal with alignments, the JFlex[2] API to create and use lexical analyzers, and the JavaCC parser generator[3] API, to create and use the syntactic analyzer.

ALIN can work with a large set of metrics, individually or in combination. One of the metrics available for ALIN is the ALIN metric. We created the ALIN metric to leverage the fact that entity names are standardized lexically and syntactically. The ALIN metric does not directly compare entity names, but rather their concept or constituent concepts. ALIN associates entities with concepts by taking advantage of the syntactic standardization of entity names, in addition to utilizing background knowledge such as WordNet and FMA (see Subsection 4.2.1). For a more detailed study of the ALIN metric, see Appendix III.

4.2.1. How ALIN associates the concept of an entity

Before starting the mapping selection, ALIN loads all the entities from the ontologies into memory. It performs lexical and syntactic analysis for each entity name. It stores in the entity structure their lexemes and tokens, their variations, and the nested terms of their entity names.

In ALIN, we associate each entity name, each nested term in the entity name and each word in the entity name with a concept (Definition 3.5). We equate concepts in various scenarios:

- All synonyms in an ontology are associated with the same concept;
- All synonyms in Wordnet are associated with the same concept;
- All synonyms in FMA are associated with the same concept;
- Each word from an entity name in an ontology found in Wordnet has its concept equated to the corresponding Wordnet word concept. If ALIN fails to locate the word, it undergoes a stemming process before ALIN searches for it again;
- Each entity name from an ontology found in FMA has its concept equated to the concept of the corresponding FMA term;
- Every entity name found in another ontology has its concept equated to the concept of the other ontology;
- Every entity name that is a variation (Definition 3.7) of an entity name from the other ontology has its concept equated to the concept of the another ontology;
- Any portion of an entity name that constitutes a nested term (see Definition 3.6) relative to another entity name in the same or the other ontology has its concept equated to the of this entity.

4.3. Evaluation overview

To evaluate ALIN, we followed the same protocol as the OAEI interactive matching track[93] in which ALIN participated [99][100][101][102][103][104][105][106]. The OAEI provides a reference alignment for the evaluation of ontology matching tools' performance. ALIN uses the reference alignment to simulate the expert answering. At each interaction, up to three selected mappings can be submitted to the expert, as long as each selected mapping has one entity in common with another selected mapping in the interaction [107]. The quality of an alignment generated by a matching approach is generally measured by the F-Measure, which is the harmonic mean between Recall and Precision. When the ontology matching process is interactive another quality metric occurs, it is the number of interactions with the expert.

We did not implement lowercase normalization or stemming in the lexical analyzers. ALIN always normalizes entity names to lowercase. It applies stemming only when it cannot find a word in WordNet. In such cases, it applies stemming and searches again in WordNet.

We evaluated our hypothesis by running ALIN five times. Each of the executions had the following characteristics:

- In the first execution, we ran ALIN with three string-based metrics. We utilized Jaccard, Jaro-Winkler, and n-gram. We based the process of choosing the similarity metrics used by ALIN on the results of these metrics

[2]http://jflex.de last accessed on Jan, 24, 2024.
[3]https://javacc.org/ last accessed on Jan, 24, 2024.

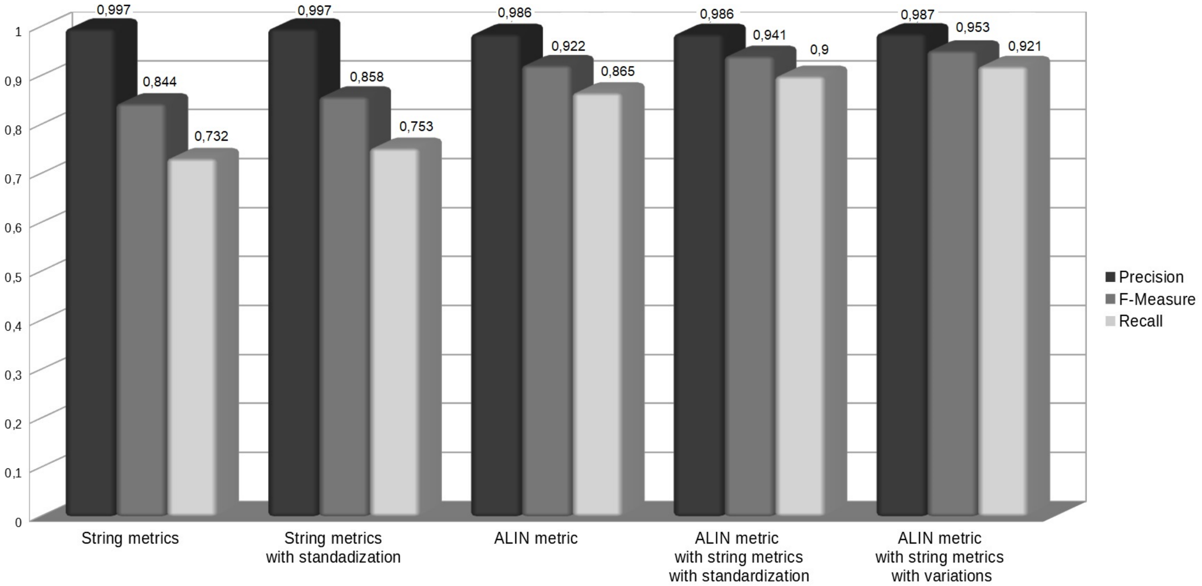

Fig. 2. Comparison between different ALIN versions - Quality

in assessments [108]. In this execution, we employed lexical analyzers to implement standardization techniques without considering the other ontology. We implemented the following standardization techniques: Word separation, Conversion of Roman numerals to Arabic numerals, Removal of stop words, Removal of non-alphanumeric characters, and Removal of punctuation characters;

- In the second execution, we employed the lexical analyzers, but this time, we considered the other ontology. We implemented the same techniques as the first execution. Additionally, we utilized the lexical analyzers for expanding acronyms or matching synonymous words that had acronyms if they existed in the other ontology. All subsequent executions utilized the same lexical analyzers.
- In the third execution, we ran ALIN without the string-based metrics. We solely utilized the ALIN metric;
- In the fourth execution, we included the three string-based metrics along with the ALIN metric. We applied the string-based metrics in the same way as in the second execution, meaning that we considered the other ontology;
- In the fifth execution, we used the same metrics as in the fourth execution, but we made a modification in the utilization of the string-based metrics. We will not use the string metric solely with the entity name returned from the lexical analyzer. Instead, we will utilize the variations and synonyms, for which ALIN has also made variations, to calculate the metric value. For example, consider two entity names. The name of the first entity has two variations and two synonyms. This entity has five names associated with it, including its original name. If we aim to calculate the string metric relative to another entity with three associated names, we will perform fifteen comparisons. The highest value found will be the value assigned to the metric calculation.

In the first two executions, we opted to include directly in the alignment, without requiring expert approval, all names that became identical after preprocessing, i.e., after using the lexical analyzers. In the three last executions, we no longer included directly in the alignment mappings where entities had identical names. Instead, we included mappings where entities shared the same concept (Subsection 4.2.1) or where the value of the ALIN metric was 1. In all ALIN executions, we selected to present to the expert the mappings with the highest sum of the metrics used in the matching process.

4.4. *Analysis of the results*

Our results demonstrate that considering the other ontology during lexical standardization leads to better outcomes compared to disregarding it. We can observe this when comparing the results of the second execution with

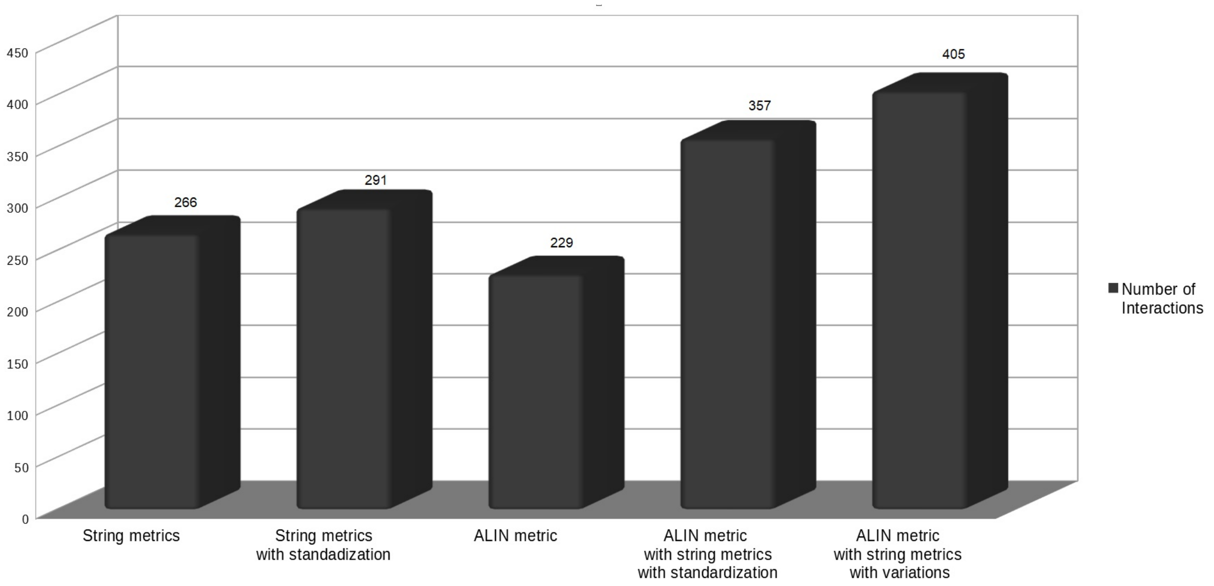

Fig. 3. Comparison between different ALIN versions - Number of interactions

those of the first (Figures 2 and 3). Our hypothesis was confirmed when comparing the fifth execution with the fourth execution, as we achieved better quality in the fifth execution. The difference between the two executions is due to the fact that we also compared variations of entity names. ALIN could calculate these variations because the entity names followed the syntax defined by the target grammar. Furthermore, when comparing the result of the third execution to the second one, we can see that the third execution achieved higher quality with fewer iterations. That shows us that a metric based on characteristics of the target grammar can yield good results. Of course, part of this result is not solely due to the use of the target grammar but also because we use concepts for comparison instead of entity names. ALIN uses not only the target grammar to find the concept of an entity but also background knowledge such as Wordnet and FMA (see Subsection 4.2.1).

ALIN participate in the OAEI interactive matching track since 2016. OAEI provides a comparison between tool performance in the interactive matching track each year, and it uses the Conference and the Anatomy data sets [109][110][93]. Here, we present the results of the Anatomy set, which we used to create the lexical analyzers, as illustrated in Table 2. AML's last participation was in 2021, while XMAP's last participation was in 2018. The ALIN appears two times. For this paper, we automated further and develop better lexical analyzers, resulting in a better outcome for the ALIN than its performance in the OAEI 2023. The first time represents the result achieved with the ALIN using the lexical analyzers described in this paper. The second time displays the outcome of the ALIN in the OAEI 2023. The table demonstrates that in terms of F-Measure, ALIN, using the techniques described in this paper, produced a high-level outcome.

Table 2

Comparison between OAEI interactive matching track tools using Anatomy data set with 100% hit rate [109][110][93]

	Total Requests	Precision	F-Measure	Recall	Year
ALIN	405	0.987	**0.952**	0.921	this paper
ALIN	514	0.987	**0.952**	0.92	2023
LogMap	388	**0.988**	0.912	0.846	2023
AML	189	0.972	**0.952**	**0.933**	2021
XMAP	**35**	0.929	0.897	0.867	2018

5. Conclusion and Future Works

Many tools standardize entity names before calculating the value of linguistic metrics. Generally, this standardization occurs without regard for the ongoing matching process. Additionally, it often overlooks the syntactic standardization of entity names. In this paper, we introduce an approach that creates a customized lexical analyzer tailored to standardizing ontology entity names within ongoing matching. The lexical analyzer will standardize entity names both lexically and syntactically. We evaluate the effectiveness of standardization taking into account the ongoing matching. We also assess the effectiveness of using the proposed grammar to improve the quality of the generated alignment. To conduct the evaluation, we utilized ALIN, an interactive ontology matching system, in conjunction with the human and mouse ontologies from the Anatomy track of the OAEI.

The contributions of this paper are:

- taking ongoing matching into consideration to standardize entity names lexically and syntactically. We achieve this through custom lexical analyzers designed for the matching process;
- the availability of the lexical analyzer as separate files. That allows other ontology matching tools reuse them;
- the development of an iterative process for creating the lexical analyzer. Each iteration in the lexical analyzer construction process generates a new version of the lexical analyzer. Each new version builds upon its predecessor;
- the creation of a context-free grammar that allows for the use of techniques that improve alignment;
- the creation of the ALIN metric, which considers the grammar to calculate the similarity value between two entity names;
- The use of string-based metrics in a way that takes grammar into account enhances their ability to find correct mappings.

We defined as our research hypothesis that *the lexical and syntactic standardization, in accordance with the grammar proposed, improves the quality of the resulting alignment*.

To leverage the lexical and syntactically standardization of entity names, we develop a new metric, the ALIN metric. This metric has demonstrated favorable outcomes. We incorporate string metrics into the ALIN, which proved valuable in enhancing the quality of the generated alignment. This improvement was greater with the inclusion of variations and synonyms of these entity names in the calculation of the string-based metrics. As the use of target grammar allows for the generation of variations, this improvement confirms our hypothesis.

An area for future exploration could involve investigating the standardization of entity names not only within ontologies but also across external resources like Wordnet and FMA. Standardizing WordNet and FMA terms could prove valuable in identifying new mappings in cases where a synonym exists in Wordnet or FMA but not within the involved ontologies.

Another area of interest is the automation of lexical analyzer construction. Currently, we have programs that provide information to the NLP Expert. In some situations, these programs directly generate the lines that the NLP Expert should include in the lexical analyzer. We can further automate this process.

6. Appendix I

We followed the process for creating the lexical analyzer and created two lexical analyzers, one for the human ontology and another for the mouse ontology. We have generated eleven versions of each, where each subsequent version incorporates all standardization techniques from the previous version and possibly adds a few more. We have assigned Roman numerals from I to XI to designate the versions of the lexical analyzers. In order to maintain synchronization between the versions of the lexical analyzers, there are versions of the lexical analyzers that remain unchanged from their previous version. For example, in version IX of the lexical analyzer for the human ontology, we reject all terms with hyphens that have no corresponding term in the mouse ontology. We made no modifications to version VIII of the lexical analyzer for the mouse ontology, but we created version IX to maintain the synchronization.

We passed, through the lexical analyzer, all terms related to classes, attributes, and their synonyms, a total of 3298 classes, 0 attributes, and 6104 synonyms from the human ontology, resulting in a total of 9402 terms, and 2738 classes, 0 attributes and 345 synonyms from the mouse ontology, resulting in a total of 3083 terms. We used ALIN to evaluate each version of the lexical analyzer.

We created graphs showing the number of entities with non-standard names, the number of entities with syntactically incorrect names, and the number of rejected items. We denoted each version of the lexical analyzer by a Roman numeral on the graph. Since its first version, the lexical analyzer of the human ontology has not contained any entities with non-syntactically correct names; hence, we omitted this data from the graph. We can see these graphs in Figures 4 and 5. Additionally, we created graphs showing the improvement in the quality of the generated alignment and the number of interactions with the expert; we can see these graphs in Figures 6 and 7. We only display in the graphs the versions that had some modification compared to the previous version.

The graph shows that with each version, the number of entities with non-standard, non-syntactically correct names decreased until it reached zero. With the lexical analyzer of the mouse ontology, this occurred in version VII. That is due to the fact that we implemented the final standardization technique for the lexical analyzer of this ontology in this version. In the ultimate version, the total number of rejected entities in this ontology was 49, accounting for 1.58% of the total terms in the ontology. In the human ontology, the number of rejected entities in the ultimate version was 438, which is 4.66% of the total terms in the ontology.

6.1. Version I of lexical analyzers - Creating the syntactic analyzer and the first lexical analyzer, separating the words

From this subsection, we will outline the process of generating new versions of the lexical analyzers of human and mouse ontologies. We will specialize them in aligning the Anatomy track. We will perform the actions that the NLP Expert must take to create a new version (see subsection 3.2). We ran the ALIN with ALIN metric and the string-based metrics (Jaccard. Jaro-Wringler and n-gram). In ALIN, we will not handle uppercase and lowercase in the lexical analyzer. We will standardize this at the time of comparing terms in ALIN. we will not handle stemming in the lexical analyzers. ALIN applies stemming only when ALIN does not find the word in the Wordnet. The first standardization technique in the lexical analyzer will be word separation. The human and mouse ontologies use either space or underscores to separate words. It's important to note that in the first versions of the lexical analyzer, it may break a single word into multiple lexemes (Definition 3.2). In the ultimate version of the lexical analyzer, our goal is for each lexeme to represent a single word. After the word separation, the lexical analyzer will classify the identified lexemes into one of the three tokens described in Definition 3.3.

Our first version of the lexical analyzer does not modify any term, only classifies the lexemes of the terms into the three tokens, and list entities with non-standard or non-syntactically correct name.

We can observe the first version of the lexical analyzers in the Appendix II. The regex MASK is solely for internal processing within ALIN, we found no term starting with %. The period at the end indicates that any other type of lexeme not fitting into the three tokens should be marked as non-standard for the lexical analyzer and considered invalid for comparison with entities from the other ontology to generate mappings. When the lexical analyzer mark a lexeme as 'invalid for comparison,' it is considered rejected, and ALIN will no longer use it to form a mapping.

However, when the lexical analyzer marks a lexeme as 'invalid for comparison' and 'non-standard', we do not consider it rejected. The lexical analyzer may ocasionally classify it into one of the three tokens in the future.

The OAEI provides reference alignments for its tracks. We will use the reference alignment for the anatomy track to evaluate the performance of alignments generated for each version of the lexical analyzers. That will allow us to assess the change in alignment quality after each modification of the lexical analyzers.

The results of the ALIN executions, with the first versions of the lexical analyzers, we can see in column I of Figures 4, 5, 6 and 7.

6.2. Version II of lexical analyzers - Creating the next lexical analyzer, conversion of roman letters to numeric

In the second version of the lexical analyzer, we added lines to convert Roman numerals to Arabic numerals, as shown in the line below:

"II" associate_token_to_lexeme("Numeral", "2");

In the second version of the lexical analyzer, we also standardized numbers were in a format different from the others:

"05" associate_token_to_lexeme("Numeral", "5");
"#30" associate_token_to_lexeme("Numeral", "30");

Additionally, we standardized ordinal numbers, as illustrated below:

"1th" associate_token_to_lexeme("Noun_or_Modifier", "first");
"VIIIth" associate_token_to_lexeme("Noun_or_Modifier", "eighth");

The second version, as well as all subsequent versions, were uploaded to OSF.

You can find the results of the second version of the lexical analyzer in column II of the Figures 4, 5, 6 and 7.

We observe a reduction in the number of terms that cannot be classified. For example, lexemes such as II and VIII (which were modified to 2 and 8) can now fit into the token 'numeral'. Previously, when they were II and VIII, they could not be classified into any token, as the 'noun or modifier' token specifies that only the first letter of the lexeme can be uppercase. Similarly, lexemes with special characters like #30 can now be categorized. No more terms pass through the lexical analyzer of mouse ontology without undergoing the syntactic analyzer.

6.3. Version III of lexical analyzers - Removing the stop words

We identified the stop words 'the' and 'a' in the human and the mouse ontology. We modified the lexical analyzers of both ontologies to exclude these two stop words from the entity names.

We found the stop words 'and' and 'or' in the human ontology and the stop word 'and' in the mouse ontology. We created a program to identify entity names containing 'and' or 'or' that we could standardize.

The program operates as follows: we provide two parameters to the program: one specifying the ontology and the other allowing for either 'and' or 'or'. As an example, let's assume we pass 'human' and 'and':

- The program searches through all entity names in the human ontology that contain the word 'and'. The program stores in two variables the word that comes immediately before and the one that comes immediately after;
- The program lists all entity names in the human ontology that contain both words;
- The program lists all entity names in the mouse ontology that contain both words.

Based on this information, we wrote lines in the lexical analyzer to standardize the use of 'and', ensuring that entity names indicating the same concept become uniform.

We will consider the following example:

28 - Head_and_Neck
Head
Neck
Ontology human
H and N - Head And Neck

Head and Neck, Buccal Cavity
Head and Neck, Fluids, Secretions
Head and Neck, Larynx
Head and Neck, Parathyroid
Head and Neck, Pharynx
Head and Neck, Saliva
Head and Neck, Salivary Glands
Head and Neck, Thyroid
Head and Neck-Fluids, Secretions
Head_and_Neck_Fluids_and_Secretions
Head_and_Neck_Lymph_Node
Head_and_Neck_Muscle
Head_and_Neck_Part
Lymph Node of Head, Face and Neck
Lymph Nodes of Head, Face and Neck
the_head_or_neck_vein
Vein_of_the_Head_or_Neck
Number of terms : 18
Ontology mouse
head/neck
head/neck blood vessel
head/neck bone
head/neck connective tissue
head/neck muscle
head/neck nerve/ganglion
head/neck organ
head/neck skin
Number of terms : 8

We observed that the program identified the term 'Head_and_Neck' in the human ontology and produced a list of all entity names in the human and mouse ontologies containing the words 'head' and 'neck'. For example, we have the entity name 'Head_and_Neck_Muscle' in the human ontology and 'head/neck muscle' in the mouse ontology. After observing these two entity names and taking other entity names into consideration, we concluded that we should add the following line to the lexical analyzer of the human ontology.

"Head_and_Neck" associate_token_to_lexeme("Noun_or_Modifier", "Head-and-Neck");

And add the following line to the lexical analyzer of the mouse ontology.

"head/neck" associate_token_to_lexeme("Noun_or_Modifier", "Head-and-Neck");

By standardizing the entity names this way, we enable similarity metrics to identify these entity names as equals.

The version III of the lexical analyzer is available on OSF.

The number of entities with non-standard names decreased in the mouse ontology. That is due to the presence of lexemes like 'head/neck' that the lexical analyzer was not previously categorized but is now classified as 'Noun_or_Modifier'. We marked as invalid for comparison all entities whose names contain 'and' or 'or' and for which we couldn't find a corresponding entity name in the other ontology indicating conjunction. in other words, we rejected them. We rejected all entities whose names contain slashes from the mouse ontology for which we did not find equivalents in the human ontology. In the version III of the lexical analyzer, we decreased the need for expert interactions while we elevated the quality of the generated alignment. The results can be observed in the column III of Figures 4, 5, 6 and 7.

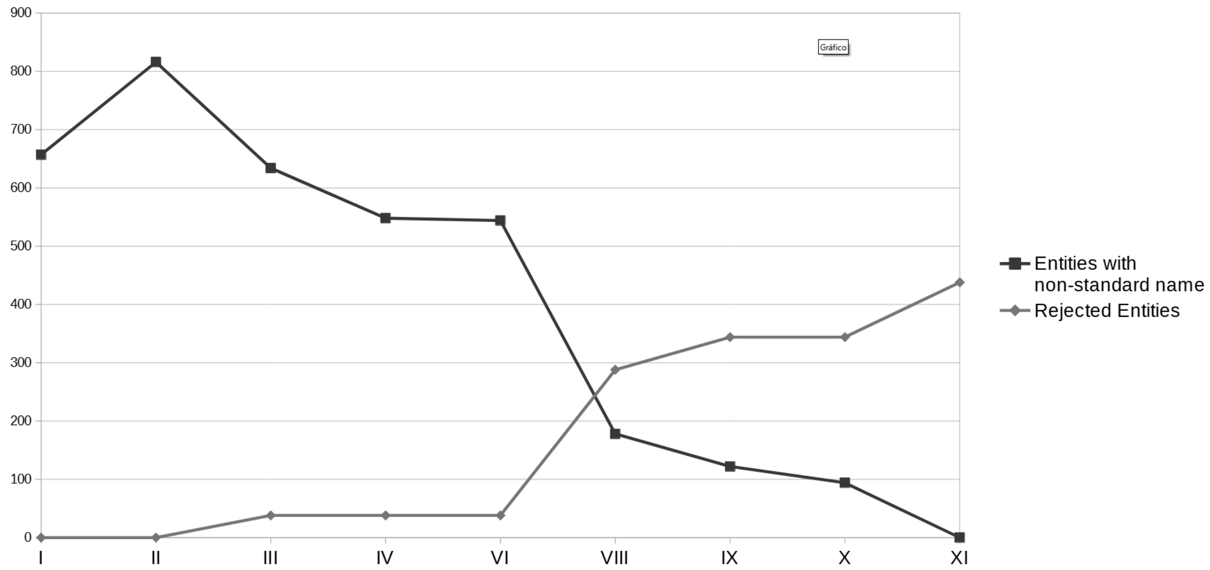

Fig. 4. Entities with non-standard name and rejected entities in each version of the lexical analyzer of the Human ontology

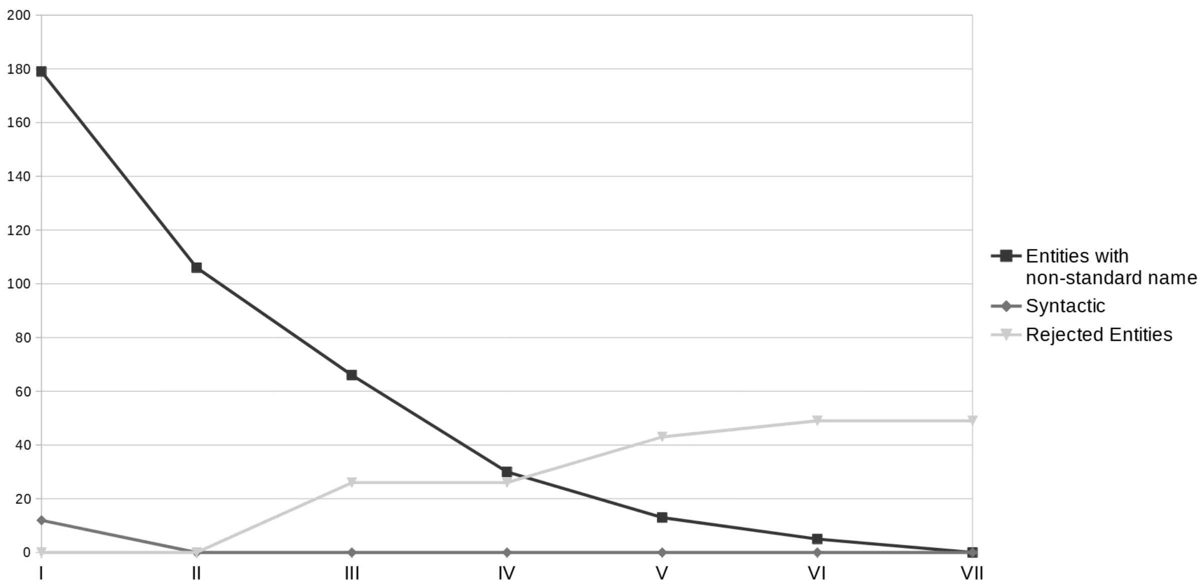

Fig. 5. Entities with non-standard or non-syntactically correct name and rejected entities in each version of the lexical analyzer of the Mouse ontology

6.4. Versions IV and V of lexical analyzers - Removal of non-alphanumeric or punctuation characters

The apostrophe was the first character we found in the mouse ontology and encountered in the human ontology with a different pattern. It represents the possessive case in the mouse ontology. The symbol '_s_' is the principal symbol representing the possessive case in the human ontology.

To aid in standardizing the entity names, we developed a program with the following characteristics:

- We pass an ontology as a parameter, for example, 'human', and another parameter indicating the symbol for the possessive case, for example, '_s_'. The program searches for all entity names that contain the symbol

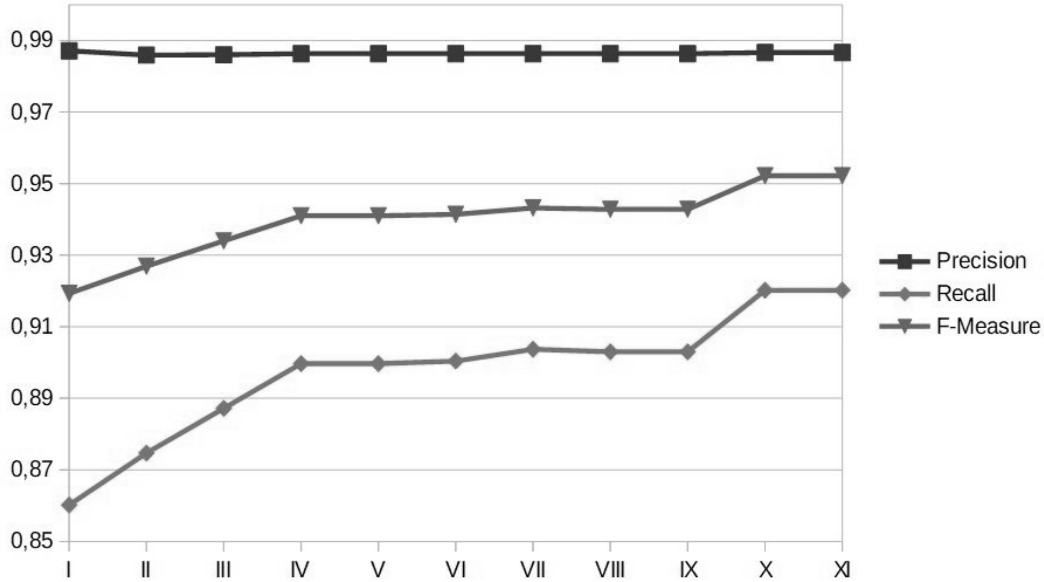

Fig. 6. Performance of ALIN for each lexical analyzer version, with the ALIN metric and with string-based metrics - Quality

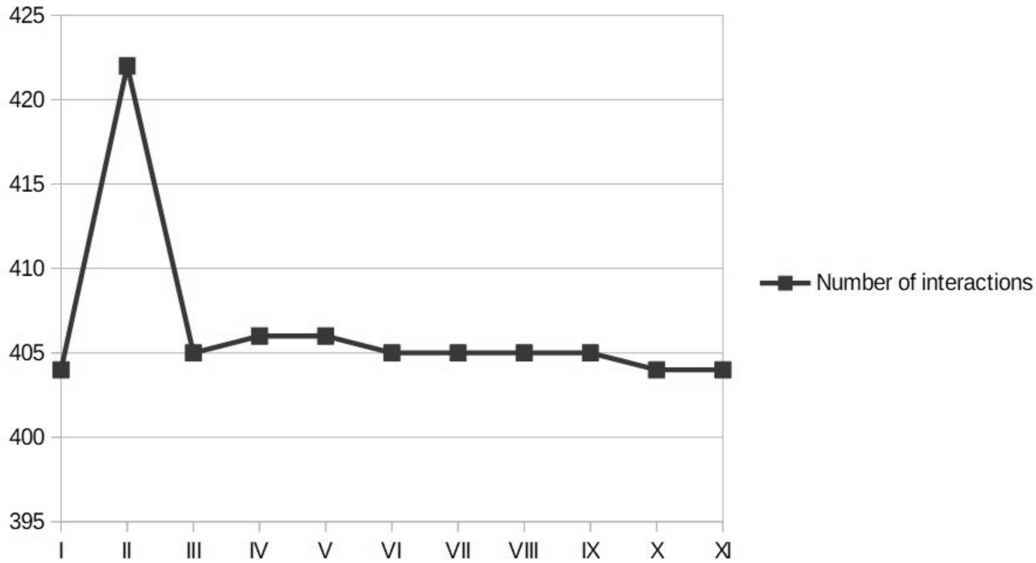

Fig. 7. Performance of ALIN for each lexical analyzer version, with the ALIN metric and with string-based metrics - Number of interactions

passed as a parameter. It records in a variable the word directly preceding the symbol.
- The program lists all entity names in the human ontology that contain the symbol;
- The program lists all entity names in the mouse ontology that contain the symbol.

We ran the program and found the following result, among others:

17 - Visceral_Layer_of_Bowman_s_Capsule - human
Bowman
Ontology human
Bowman's Capsule

Bowman's Membrane
Bowman's Space
bowman_capsule
bowman_capsule_parietal_layer
bowman_capsule_viscera_layer
bowman_capsule_visceral_layer
bowman_membrane
Bowman_s_Capsule
Bowman_s_Membrane
bowman_space
Parietal Layer of Bowman's Capsule
parietal_layer_of_bowman_capsule
Parietal_Layer_of_Bowman_s_Capsule
viscera_layer_of_bowman_capsule
Visceral Layer of Bowman's Capsule
visceral_layer_of_bowman_capsule
Number of terms : 17
Ontology mouse
bowman's capsule
bowman's space
bowman_capsule
bowman_olfactory_gland
bowman_space
olfactory gland of bowman
Number of terms : 6

We observed that the program identified the entity name ’Visceral_Layer_of_Bowman_s_Capsule’ in the human ontology and produced a list of all entity names in the human and mouse ontologies containing the words ’Bowman’. For example, we have the entity name ’Bowman's Capsule’, the entity name ’bowman_capsule’ and the entity name ’Bowman_s_Capsule’ in the human ontology and ’bowman's capsule’ and the entity name ’bowman_capsule’ in the mouse ontology. After observing these entity names and taking other entity names into consideration, we concluded that we should add the following lines to the lexical analyzer of the human ontology:

"'s" { }
"_s_" { }

And add the following line to the lexical analyzer of the mouse ontology:

"'s" { }
"’s" { }

With the lines above, we removed the lexemes "'s" and "_s_" in the human ontology and the lexems "'s" and "’s" in the mouse ontology.

We have identified additional non-alphanumeric characters in entity names, such as hyphens, commas, parentheses, plus signs, periods, and forward slashes. Among these, only the hyphen and the already seen possessive case apostrophe had corresponding entity names in the mouse ontology, necessitating standardization by the lexical analyzer. The lexical analyzer removes the possessive case apostrophe and its various forms in human and mouse ontologies from entity names that do not have counterparts in the other ontology.

We created a program that checks if a hyphenated word can have a variation in another ontology. For instance, ’hindbrain’ is present in the mouse ontology, but ’Hind-Brain’ is in the human ontology. This program was the one that generated the results in a more automated way compared to the other programs we were able to develop. The program generates all the lines of the lexical analyzer for the mouse and human ontologies. After running the program, you only need to copy the corresponding lines to the correct lexical analyzer. The result of running ALIN after modifying the lexical analyzers we can see in column IV of Figures 4, 5, 6 and 7.

We can find this version IV of the lexical analyzer on OSF.

We analyzed the entity names from the mouse and human ontologies, and we found that the hyphenated entity names still outside the standard in the mouse ontology referred to hyphenated words for which we did not encounter an equivalent in the human ontology. In the human ontology, besides hyphenated words, there were other situations, for example, where a hyphen separated a entity name from its abbreviation.

We considered this and exclude all entities whose names contain hyphens from the mouse ontology for which we did not find equivalents in the human ontology. However, we chose not to remove hyphenated entity names from the human ontology at this time. We created version V of the mouse lexical analyzer with this modification. We also created version V of the human lexical analyzer only to maintain consistency in version numbers. We can find these new versions on OSF.

The result of running ALIN after this modification of the lexical analyzers we can see in column V of Figures 4, 5, 6 and 7.

6.5. Version VI of lexical analyzers - Use of synonyms in the mouse ontology

In addition to the current use of synonyms in ALIN, where we associate entity names that have common synonyms, we will utilize synonyms to improve the lexical analyzer.

We have listed the entities from the mouse ontology that the lexical analyzer did not reject or whose names contain lexemes it did not categorize:

- adductor group (leg);
- hippocampus CA4;
- hippocampus CA3;
- hippocampus CA2;
- hippocampus CA1.
- TS28;
- hSVZ;
- VZ;
- BALT;
- GALT;
- MALT;
- PALS;
- CSF.

We created a program to search for synonyms in the other ontology, either directly or through the FMA. In the same program, we search within the same ontology to determine if these entity names have a synonym that adheres to the syntax rules.

We then have three possible responses and their corresponding actions:

- The entity name has a synonym in the other ontology - We will include this synonym as the return value for the entity name in the lexical analyzer;
- The entity name has a synonym in the same ontology that adheres to the syntax rules - The lexical analyzer will reject the entity;
- The entity name does not have a synonym in the other ontology, nor does it have a synonym in the same ontology that adheres to the syntax rules - In this case, we will not handle these entities in this version.

We ran the program and obtained the following result:
"adductor group (leg)"
// Present in mouse, this term does not appear elsewhere.

```
"hippocampus CA4"
// Present in mouse, this term does not appear elsewhere.
"hippocampus CA3"
// Present in mouse, this term does not appear elsewhere.
"hippocampus CA2"
// Present in mouse, this term does not appear elsewhere.
"hippocampus CA1"
// Present in mouse, this term does not appear elsewhere.
"TS28" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as adult mouse, this term is in the syntax.
"SVZ" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as brain subventricular zone, this term is in the syntax.
"VZ" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as brain ventricular zone, this term is in the syntax.
"BALT" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as bronchus associated lymphoid tissue, this term is in the syntax.
"CSF" { associate_token_to_lexeme("Noun_or_Modifier", "cerebrospinal");
associate_token_to_lexeme("Noun_or_Modifier", "fluid");
}
// Present in mouse also as cerebrospinal fluid, this term also found in human.
"GALT" { associate_token_to_lexeme("Noun_or_Modifier", "Gut-Associated");
associate_token_to_lexeme("Noun_or_Modifier", "lymphoid");
associate_token_to_lexeme("Noun_or_Modifier", "tissue");
}
// Present in mouse also as gut associated lymphoid tissue, this term also found in human.
"MALT" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as mucosa associated lymphoid tissue, this term is in the syntax.
"PALS" { Mark_Invalid_for_Comparison(); }
// Present in mouse also as spleen periarteriolar lymphatic sheath, this term is in the syntax.
```

Therefore, the entity names in the mouse ontology TS28, SVZ, VZ, BALT, MALT, and PALS have synonyms in the mouse ontology that adhere to syntax. Therefore, the lexical analyzer can reject them without issues, as the search for mapping can be done with their synonym. The entity names CSF and GALT have synonyms in the other ontology. So, the lexical analyzer modified the entity names.

We made the necessary modifications to the lexical analyzers, and we can see the results in column VI of Figures 4, 5, 6 and 7.

The entity names adductor group (leg), hippocampus CA4, hippocampus CA3, hippocampus CA2, and hippocampus CA1 do not have equivalents in the other ontology nor synonyms that adhere to syntax; therefore, this version of the lexical analyzer did not process them.

6.6. *Version VII of lexical analyzers - Expansion of abbreviations and acronyms in the mouse ontology*

In this phase, we will search within the human ontology for the abbreviations and acronyms found in the mouse ontology. We will utilize ChatGPT to expand them or to verify if entity names using the same abbreviation are synonyms.

Four of these entity names have acronyms: CA1, CA2, CA3, and CA4.

We conducted a search for these abbreviations in the human ontology and found entity names that use them. We asked ChatGPT if the entity names obtained as a response were synonyms, and the answer was affirmative.

Additionally, as only one entity name remained, "adductor group (leg)," we searched if 'adductor group' also existed in the human ontology and received a positive response. The found entity name also proved to be a synonym, according to ChatGPT.

Thus, we obtained the following result, with the first entity name from the mouse ontology and the second term from the human ontology:

- adductor group (leg) - Adductor Group of the Leg
- hippocampus CA1 - CA1_Field_of_the_Cornu_Ammonis
- hippocampus CA2 - CA2_Field_of_the_Cornu_Ammonis
- hippocampus CA3 - CA3_Field_of_the_Cornu_Ammonis
- hippocampus CA4 - CA4_Field_of_the_Cornu_Ammonis
- hippocampus CA5 - No equivalent term was found.

We made the necessary modifications to the respective lexical analyzers. We can see the results in column VII of Figures 4, 5, 6 and 7, indicating that there are no longer entities with non-standard or non-syntactically correct name in the mouse ontology. The lexical analyzer rejected a total of 49 entities from the mouse ontology. There was an improvement in alignment quality with the same number of interactions with the expert.

6.7. Versions VIII and IX of lexical analyzers - Use of synonyms in the human ontology

We executed the same program of the subsection 6.5 for the human ontology, rejecting all entities with non-standard or non-syntactically correct name that had synonyms with syntactically correct name. We obtained the result shown in column VIII of Figures 4, 5, 6 and 7.

The result suggests that numerous entities fell into the situation described above, as evidenced by the increase in rejections in the human ontology from 38 to 288.

Upon analyzing the result, we realized that the remaining entity names that use hyphens do so to separate words, so we can now apply the rejection of entities with hyphens, as we did with the mouse ontology in subsection 6.4. Those that may have equivalents in the other ontology the lexical analyzer have already modified in version V.

We can see the results after removing the hyphenated words in the column IX.

6.8. Versions X and XI of lexical analyzers - Expansion of abbreviations and acronyms in the human ontology

We found many acronyms among the entity names still not standardized, such as:

- L1_Vertebra
- LH_Cell
- FSH_Cell
- B-Cell
- T-LGL

We searched for "L1" in the mouse ontology but did not find it. However, we encountered this term in FMA with a synonym like "first lumbar vertebra," a term that does not exist in the mouse ontology. We looked for the term "lumbar vertebra," and one of the results was "lumbar vertebra 1." We asked ChatGPT if "first lumbar vertebra" is a synonym of "lumbar vertebra 1," and the result was positive. We modified the lexical analyzer accordingly. We continued these verifications for each acronym found.

We can see the results of including these expansions in column X of Figures 4, 5, 6 and 7, indicating a decrease in the number of entities with non-standardized names and an improvement in the quality of the generated alignment.

Following this result, we modified the human lexical analyzer again to reject all entities whose names we couldn't expand the acronyms to a entity name that had an equivalent in the mouse ontology. We can see in the column XI of Figures 4, 5, 6 and 7.

All versions of the lexical analyzers can be found on OSF.

7. Appendix II

7.1. Lexical analyzer for mouse ontology

```
package ALIN;
%%

%{

static String Lexema[] = new String[200];
static String Token[] = new String[200];
static int lexema_position;
static boolean Valid_for_comparison;
static boolean term_outside_the_lexical_pattern;

private void associate_token_to_lexeme(String token, String lexema)
{
Lexema[lexema_position]=lexema;
Token[lexema_position]=token;
lexema_position++;
}

private void Mark_Invalid_for_Comparison()
{
Valid_for_comparison=false;
}

private void Mark_term_outside_the_lexical_pattern()

term_outside_the_lexical_pattern=true;


%}

%class regex_mouse
%type void

// Masks - Masks are used in ALIN's internal processing and are not found in the ontologies UNDERSCORE =
[_]
MASK = %([0-9])+

// Common terms
NOUN_OR_MODIFIER = [a-z|A-Z]([a-z])+
SPACE = [ ]
NUMBER = [0-9]+


%%
//——————————————————————————————————————-

// Token association without lexeme modification - Prepositions
```

```
"of" { associate_token_to_lexeme("Preposition", yytext()); }
"for" { associate_token_to_lexeme("Preposition", yytext()); }
"to" { associate_token_to_lexeme("Preposition", yytext()); }

//——————————————————————————————————-

{NUMBER} { associate_token_to_lexeme("Numeral", yytext()); }

//——————————————————————————————————-

// Token association with no lexeme modification - mask
{MASK} { associate_token_to_lexeme("Noun_or_Modifier", yytext()); }

//——————————————————————————————————-

// Token association with no lexeme modification - noun or Modifier
{NOUN_OR_MODIFIER} { associate_token_to_lexeme("Noun_or_Modifier", yytext()); }
// Removal of the underscore
{UNDERSCORE} { }
// Removal of the space
{SPACE} { }

. { Mark_term_outside_the_lexical_pattern();
Mark_Invalid_for_Comparison(); }
```

7.2. Lexical analyzer for human ontology

```
package ALIN;
%%

%{

static String Lexema[] = new String[200];
static String Token[] = new String[200];
static int lexema_position;
static boolean Valid_for_comparison;
static boolean term_outside_the_lexical_pattern;

private void associate_token_to_lexeme(String token, String lexema)

Lexema[lexema_position]=lexema;
Token[lexema_position]=token;
lexema_position++;


private void Mark_Invalid_for_Comparison()

Valid_for_comparison=false;


private void Mark_term_outside_the_lexical_pattern()

term_outside_the_lexical_pattern=true;


%}

%class regex_human
%type void

// Masks - Masks are used in ALIN's internal processing and are not found in the ontologies
MASK = %([0-9])+([a-z])*
HIFEN = [-]
MASK_WITH_HIFEN = {NOUN_OR_MODIFIER}{HIFEN}{MASK}

// Common terms - Each lexema separated by underscore or space, with the first letter capitalized. // The lex-
ema can be a string of numbers NOUN_OR_MODIFIER = [a-z|A-Z]([a-z])+
SPACE = [ ]
UNDERSCORE = [_]
NUMBER = [0-9]+

%%

//————————————————————————————————————————-

// Token association with no lexeme modification - mask
```

```
{MASK} { associate_token_to_lexeme("Noun_or_Modifier", yytext()); }
{MASK_WITH_HIFEN} { associate_token_to_lexeme("Noun_or_Modifier", yytext()); }

//————————————————————————————————-

// Token association with no lexeme modification - Prepositions
"of" { associate_token_to_lexeme("Preposition", yytext()); }
"for" { associate_token_to_lexeme("Preposition", yytext()); }
"at" { associate_token_to_lexeme("Preposition", yytext()); }
"with" { associate_token_to_lexeme("Preposition", yytext()); }
"to" { associate_token_to_lexeme("Preposition", yytext()); }

//————————————————————————————————-

// Token association with no lexeme modification - noun or Modifier
{NOUN_OR_MODIFIER} { associate_token_to_lexeme("Noun_or_Modifier", yytext()); }
// Removal of the underscore
{UNDERSCORE} { }
// Removal of the space
{SPACE} { }

//————————————————————————————————-

{NUMBER} { associate_token_to_lexeme("Numeral", yytext()); }

//————————————————————————————————-

. { Mark_term_outside_the_lexical_pattern();
Mark_Invalid_for_Comparison(); }
```

7.3. Method calling the lexical analyzer for human ontology

```java
    private void Executar_regex_human ()
    {

        InputStream targetStream = new ByteArrayInputStream(this.Termo.getBytes());

        InputStreamReader inputStreamReader = new InputStreamReader(targetStream);

        BufferedReader in = new BufferedReader(inputStreamReader);

        regex_human.lexema_position=0;
        regex_human.Valid_for_comparison=true;
        regex_human.term_outside_the_lexical_pattern=false;

        regex_human a = new regex_human(in);

        try
        {
            a.yylex();

            in.close();
        }
        catch (Exception e)
        {
            System.out.println("Issues with lexical analysis!!!");
        }

        if (regex_human.term_outside_the_lexical_pattern)
        {
            System.out.println
            ("Term outside the standard lexical analysis pattern: "+this.Termo);
            this.term_outside_the_lexical_pattern=true;
        }

        if (!regex_human.Valid_for_comparison)
        {
            if (this.Entidade!=null)
                this.Entidade.Valido_para_comparacao=false;
            this.Valido_para_comparacao=false;
        }
        else
            for (int i=0;i<regex_human.lexema_position;i++)
            {
                this.Lexemas_na_ordem_original[i]=regex_human.Lexema[i];
                this.Tokens_na_ordem_original[i]=regex_human.Token[i];
            }

        return;
    }
```

8. Appendix III

8.1. ALIN *metric*

The lexical analyzer returns the set of lexemes (Definition 3.2) of the entity name. In the ultimate version of the lexical analyzer, each returned lexeme corresponds to a word in the entity name. In addition to returning the words, the lexical analyzers categorize them into one of three tokens: noun or modifier, preposition or numeral. The lexical analyzers return the words and their corresponding tokens (Definition 3.2).

Therefore, the entity name 'Type I Cell of the Epidermis' will have the following lexemes returned by the lexical analyzer, remembering that the stop word 'the' is ignored in our lexical analyzers:

– lexeme 1 - Type
– lexeme 2 - I
– lexeme 3 - Cell
– lexeme 4 - of
– lexeme 5 - Epidermis

and the tokens:

– token 1 - Noun or modifier
– token 2 - Numeral
– token 3 - Noun or modifier
– token 4 - Preposition
– token 5 - Noun or modifier

We pass these tokens, in this order, to the syntactic analyzer, which will check if they obey the grammar and return to us 3, 1, 2, 5. This order indicates that the first number is the number of the lexeme which is the head noun of the noun phrase (Definition 3.4). We will use this output from the syntactic analyzer to search for mappings between the two ontologies.

Table 3
Treatment of the term 'aorta smooth muscle'

Entity name	Concept	Ontology
aorta smooth muscle	7c393a53	mouse
Array of words		
Array element	Concept	
muscle	4b4adf94	
smooth	2396cb65	Word found on Wordnet, thus having gained the same concept given on Wordnet.
aorta	334b7684	
Array of words and nested terms		
Array element	Concept	
smooth muscle	47ef410e	Term found on Mouse ontology
aorta	334b7684	

8.2. ALIN *metric calculation*

After doing this, we search for nested terms within entity names. For each entity name, which we will refer to as a full name, we will search in both ontologies for entity names that are subterms of the full name. We will form two arrays for each entity name: one called the 'array of words' and the other called the 'array of words and nested terms', as shown in Tables 3, 4 and 5.

Table 4
Treatment of the term 'Aorta_Smooth_Muscle_Tissue'

Entity name	Concept	Ontology
Aorta_Smooth_Muscle_Tissue	7c393a53	human
Array of words		
Array element	Concept	
tissue	361a56e3	Word found on Wordnet, thus having gained the same concept given on Wordnet.
muscle	4b4adf94	
smooth	2396cb65	Word found on Wordnet, thus having gained the same concept given on Wordnet.
aorta	334b7684	
Array of words and nested terms		
Array element	Concept	
Smooth_Muscle_Tissue	47ef410e	Term synonym with 'Smooth_Muscle' in Human ontology, thus having its concept unified with that. As there is a term of the same name, 'smooth muscle' in the Mouse ontology, their concepts are unified.
aorta	334b7684	

Table 5
Treatment of the term 'Smooth_Muscle'

Entity name	Concept	Ontology
Smooth_Muscle	47ef410e	human
Array of words		
Array element	Concept	
muscle	4b4adf94	
smooth	2396cb65	Word found on Wordnet, thus having gained the same concept given on Wordnet.
Array of words and nested terms		
Array element	Concept	
muscle	4b4adf94	
smooth	2396cb65	Word found on Wordnet, thus having gained the same concept given on Wordnet.

ALIN initially calculates two values, one based on the array of words and the other based on the array of words and nested terms.

For each array, ALIN calculates the value:

1. 1.0, if the first element of each array is the same concept, and the remaining ones are the same even if they are not in the same order. The first element of the array is the head noun, as returned by the syntactic analyzer;
2. 0.99, if the first element of each array is the same concept, and the remaining ones of one array are a subset of the other array;
3. 0, if there is a different element in the arrays.

The ALIN metric is the higher value of the two calculated.

If we compare the entity names from tables 3 and 4, ALIN finds the value 0 for the array of words metric, because there are different concepts in them. ALIN finds the value 1 for the array of words and nested terms metric. So, as the ALIN metric is 1, ALIN unifies the concepts of the two terms, as can be seen in the tables.

If we compare the entity names from tables 4 and 5, ALIN finds the value 0 for the array of words metric since although the concepts contained in the array of table 5 are a subset of the concepts in table 4, the heads are different. ALIN finds 0 for the array of words and nested terms metric, as there are different concepts in these two arrays. So, the ALIN metric is 0.

References

[1] J. Euzenat and P. Shvaiko, *Ontology Matching - Second Edition*, Springer-Verlag, 2013. ISBN 978-3-642-38720-3.

[2] J. Da Silva, K. Revoredo, F. Baião and J. Euzenat, Alin: improving interactive ontology matching by interactively revising mapping suggestions, *The Knowledge Engineering Review* **35** (2020). doi:10.1017/S0269888919000249.

[3] M. Espinoza Mejía, R. Trillo, J. Gracia and E. Mena, Discovering and Merging Keyword Senses using Ontology Matching., 2006.

[4] W. Hu, G. Cheng, D. Zheng, X. Zhong and Y. Qu, The Results of Falcon-AO in the OAEI 2006 Campaign, in: *Proceedings of the 1st International Conference on Ontology Matching - Volume 225*, OM'06, CEUR-WS.org, Aachen, DEU, 2006, pp. 124–133–.

[5] I.-H. Kuo and T.-T. Wu, ODGOMS - Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 153–160–.

[6] O. Hassanzadeh, M.J. Ward, M. Rodriguez-Muro and K. Srinivas, Understanding a large corpus of web tables through matching with knowledge bases: an empirical study., in: *OM*, 2015, pp. 25–34.

[7] S. Cerón-Figueroa, I. López-Yáñez, W. Alhalabi, O. Camacho-Nieto, Y. Villuendas-Rey, M. Aldape-Pérez and C. Yáñez-Márquez, Instance-based ontology matching for e-learning material using an associative pattern classifier, *Computers in Human Behavior* **69** (2017), 218–225. doi:https://doi.org/10.1016/j.chb.2016.12.039. https://www.sciencedirect.com/science/article/pii/S0747563216308603.

[8] S. Ceron, I. López-Yáñez, Y. Villuendas-Rey, O. Camacho, M. Aldape-Pérez and C. Yáñez-Márquez, Instance-Based Ontology Matching For Open and Distance Learning Materials, *The International Review of Research in Open and Distributed Learning* **18** (2017). doi:10.19173/irrodl.v18i1.2681.

[9] S. Hertling and H. Paulheim, ATBox results for OAEI 2020, in: *Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 2788, CEUR-WS.org, 2020, pp. 168–175. https://ceur-ws.org/Vol-2788/oaei20_paper5.pdf.

[10] S. Hertling, J. Portisch and H. Paulheim, Supervised Ontology and Instance Matching with MELT, 2020.

[11] M. Dhouib, C. Faron-Zucker and A. Tettamanzi, Measuring Clusters of Labels in an Embedding Space to Refine Relations in Ontology Alignment, *Journal on Data Semantics* **10** (2021). doi:10.1007/s13740-021-00137-8.

[12] Y. He, J. Chen, D. Antonyrajah and I. Horrocks, Biomedical ontology alignment with BERT, in: *Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 3063, CEUR-WS.org, 2021, pp. 1–12. https://ceur-ws.org/Vol-3063/om2021_LTpaper1.pdf.

[13] Y. Zhang, X. Wang, S. He, K. Liu, J. Zhao and X. Lv, IAMA Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 123–130–.

[14] A. Doan, J. Madhavan, P. Domingos and A. Halevy, Learning to Map between Ontologies on the Semantic Web, in: *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, Association for Computing Machinery, New York, NY, USA, 2002, pp. 662–673–. ISBN 1581134495. doi:10.1145/511446.511532.

[15] P. Lambrix and H. Tan, SAMBO—A system for aligning and merging biomedical ontologies, *Web Semantics: Science, Services and Agents on the World Wide Web* **4**(3) (2006), 196–206, Semantic Web for Life Sciences. doi:https://doi.org/10.1016/j.websem.2006.05.003. http://www.sciencedirect.com/science/article/pii/S1570826806000151.

[16] M.H. Seddiqui and M. Aono, An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size, *Journal of Web Semantics* **7**(4) (2009), 344–356, Semantic Web challenge 2008. doi:https://doi.org/10.1016/j.websem.2009.09.001. https://www.sciencedirect.com/science/article/pii/S1570826809000432.

[17] D. Faria, C. Martins, A. Nanavaty, A. Taheri, C. Pesquita, E. Santos, I.F. Cruz and F.M. Couto, Agreementmakerlight Results for OAEI 2014, in: *Proceedings of the 9th International Conference on Ontology Matching - Volume 1317*, OM'14, CEUR-WS.org, Aachen, DEU, 2014, pp. 105–112–.

[18] B. Byrne, A. Fokoue, A. Kalyanpur, K. Srinivas and M. Wang, Scalable Matching of Industry Models: A Case Study, in: *Proceedings of the 4th International Conference on Ontology Matching - Volume 551*, OM'09, CEUR-WS.org, Aachen, DEU, 2009, pp. 1–12–.

[19] G. Bouma, Cross-Lingual Dutch to English Alignment Using Eurowordnet and Dutch Wikipedia, in: *Proceedings of the 4th International Conference on Ontology Matching - Volume 551*, OM'09, CEUR-WS.org, Aachen, DEU, 2009, pp. 224–229–.

[20] C. Pesquita, C. Stroe, I.F. Cruz and F.M. Couto, BLOOMS on Agreementmaker: Results for OAEI 2010, in: *Proceedings of the 5th International Conference on Ontology Matching - Volume 689*, OM'10, CEUR-WS.org, Aachen, DEU, 2010, pp. 134–141–.

[21] T. Togia, F. McNeill and A. Bundy, Harnessing the Power of Folksonomies for Formal Ontology Matching On-the-Fly, in: *Proceedings of the 5th International Conference on Ontology Matching - Volume 689*, OM'10, CEUR-WS.org, Aachen, DEU, 2010, pp. 226–227–.

[22] F.C. Schadd and N. Roos, MaasMatch results for OAEI 2011, in: *Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011*, P. Shvaiko, J. Euzenat, T. Heath, C. Quix, M. Mao and I.F. Cruz, eds, CEUR Workshop Proceedings, Vol. 814, CEUR-WS.org, 2011. https://ceur-ws.org/Vol-814/oaei11_paper9.pdf.

[23] Z. Lv, An Effective Approach for Large Ontology Matching Using Multi-Objective Grasshopper Algorithm, in: *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*, ICCAI '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 110–116–. ISBN 9781450396110. doi:10.1145/3532213.3532230.

[24] A. Kammoun and G. Diallo, ServOMap Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 169–176–.

[25] A.N. Tigrine, Z. Bellahsene and K. Todorov, LYAM++ results for OAEI 2016, in: *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham, O. Hassanzadeh and R. Ichise, eds, CEUR Workshop Proceedings, Vol. 1766, CEUR-WS.org, 2016, pp. 196–200. http://ceur-ws.org/Vol-1766/oaei16_paper11.pdf.

[26] M. Cheatham and P. Hitzler, StringsAuto and MapSSS results for OAEI 2013, in: *Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013*, P. Shvaiko, J. Euzenat, K. Srinivas, M. Mao and E. Jiménez-Ruiz, eds, CEUR Workshop Proceedings, Vol. 1111, CEUR-WS.org, 2013, pp. 146–152. https://ceur-ws.org/Vol-1111/oaei13_paper7.pdf.

[27] A. Van Berne and V. Malaisé, Evaluation of String Normalisation Modules for String-Based Biomedical Vocabularies Alignment with AnAGram, in: *Proceedings of the 2014 International Conference on Posters & Demonstrations Track - Volume 1272*, ISWC-PD'14, CEUR-WS.org, Aachen, DEU, 2014, pp. 237–240–.

[28] E. Jiménez-Ruiz, B.C. Grau, Y. Zhou and I. Horrocks, Large-scale Interactive Ontology Matching: Algorithms and Implementation, in: *ECAI 2012 - 20th European Conference on Artificial Intelligence*, Vol. 242, 2012, pp. 444–449.

[29] A. Khiat, M. Benaissa and M. Belfedhal, STRIM Results for OAEI 2015 Instance Matching Evaluation, 2015.

[30] A. Khiat, CroLOM: cross-lingual ontology matching system results for OAEI 2016, in: *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham, O. Hassanzadeh and R. Ichise, eds, CEUR Workshop Proceedings, Vol. 1766, CEUR-WS.org, 2016, pp. 146–152. https://ceur-ws.org/Vol-1766/oaei16_paper3.pdf.

[31] A. Khiat, E.A. Ouhiba, M.A. Belfedhal and C.E. Zoua, SimCat results for OAEI 2016, in: *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham, O. Hassanzadeh and R. Ichise, eds, CEUR Workshop Proceedings, Vol. 1766, CEUR-WS.org, 2016, pp. 217–221. https://ceur-ws.org/Vol-1766/oaei16_paper14.pdf.

[32] L. Amir, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste and F. Gargouri, POMap: An Effective Pairwise Ontology Matching System, 2017, pp. 161–168. doi:10.5220/0006492201610168.

[33] A. Khiat and M. Mackeprang, I-Match and OntoIdea results for OAEI 2017., in: *OM@ ISWC*, 2017, pp. 135–137.

[34] E. Jiménez-Ruiz, A. Agibetov, M. Samwald and V.V. Cross, We divide, you conquer: from large-scale ontology alignment to manageable subtasks with a lexical index and neural embeddings, in: *OM@ISWC*, 2018. https://api.semanticscholar.org/CorpusID:52961297.

[35] M. Mohammadi, A.A. Atashin, W.J. Hofman and Y.-H. Tan, SANOM Results for OAEI 2019, in: *OM@ISWC*, 2020. https://api.semanticscholar.org/CorpusID:34628695.

[36] S. Qassimi and E.H. Abdelwahed, The Role of Collaborative Tagging and Ontologies in Emerging Semantic of Web Resources, *Computing* **101**(10) (2019), 1489–1511–. doi:10.1007/s00607-019-00704-9.

[37] I. Ouali, F. Ghozzi, R. Taktak and M.S. Hadj Sassi, Ontology Alignment using Stable Matching, *Procedia Computer Science* **159** (2019), 746–755, Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019. doi:https://doi.org/10.1016/j.procs.2019.09.230. https://www.sciencedirect.com/science/article/pii/S1877050919314164.

[38] E. Jiménez-Ruiz, A. Agibetov, J. Chen, M. Samwald and V. Cross, Dividing the Ontology Alignment Task with Semantic Embeddings and Logic-Based Modules, in: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, G.D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín and J. Lang, eds, Frontiers in Artificial Intelligence and Applications, Vol. 325, IOS Press, 2020, pp. 784–791. doi:10.3233/FAIA200167.

[39] J. Madhavan, P.A. Bernstein and E. Rahm, Generic Schema Matching with Cupid, in: *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 49–58–. ISBN 1558608044.

[40] P. Roussille and O. Teste, TOMATO: results of the 2022 OAEI evaluation campaign, in: *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, held as a virtual conference, October 23, 2022*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 3324, CEUR-WS.org, 2022, pp. 210–215. https://ceur-ws.org/Vol-3324/oaei22_paper13.pdf.

[41] W.W.K. Chua and J.-J. Kim, Eff2Match Results for OAEI 2010, in: *Proceedings of the 5th International Conference on Ontology Matching - Volume 689*, OM'10, CEUR-WS.org, Aachen, DEU, 2010, pp. 150–157–.

[42] M. Cheatham, MapSSS Results for OAEI 2011, in: *Proceedings of the 6th International Conference on Ontology Matching - Volume 814*, OM'11, CEUR-WS.org, Aachen, DEU, 2011, pp. 184–189–.

[43] S. Hertling, Hertuda results for OEAI 2012, 2012. https://api.semanticscholar.org/CorpusID:61879831.

[44] H. Paulheim and S. Hertling, WeSeE-Match Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 197–202–.

[45] S. Schwichtenberg and G. Engels, RSDL workbench results for OAEI 2015, 2015.

[46] P. Roussille, I.M. Bousarsar, O. Teste and C.T. dos Santos, Holontology: results of the 2018 OAEI evaluation campaign, in: *13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference (OM@ISWC 2018)*, CEUR-WS : Workshop proceedings, Monterey, CA, US, 2018, pp. 167–172, Thanks to CEUR editor, a free open-access publication service of Sun SITE Central Europe operated under the umbrella of RWTH Aachen University. This paper appears in vol. 2288 Ceur Workshop Proceedings ISSN : 1613-0073 The definitive version is available at: http://ceur-ws.org/Vol-2288/oaei18_paper8.pdf. https://oatao.univ-toulouse.fr/22564/.

[47] S. Hertling and H. Paulheim, DOME Results for OAEI 2019, in: *OM 2019 : Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019) Auckland, New Zealand, October 26, 2019*, Vol. 2536, P. Shvaiko, ed., RWTH Aachen, Aachen, Germany, 2019, pp. 123–130. https://madoc.bib.uni-mannheim.de/53427/.

[48] O. Fallatah, Z. Zhang and F. Hopfgartner, A hybrid approach for large knowledge graphs matching, 2021.

[49] J. Chakraborty, B. Yaman, L. Virgili, K. Konar and S. Bansal, OntoConnect: Results for OAEI 2020, *CEUR Workshop Proceedings* **2788** (2020), 204–210, Funding Information: Beyza Yaman has been supported by the European Union's Horizon 2020 research and innovation programme under Marie Sklodowska-Curie grant agreement No. 801522, by Science Foundation Ireland and co-funded by the European Regional Development Fund through the ADAPT Centre for Digital Content Technology [grant number 13/RC/2106] and Ordnance Survey Ireland. Funding Information: The authors gratefully acknowledge the Google Summer Code program and DBpedia organization for guidance and support. We also thank the Google Cloud Platform (GCP) research credits program for providing an environment to run the experiments using their Cloud Computing services. Beyza Yaman has been supported by the European Union's Horizon 2020 research and innovation programme under Marie Sklodowska-Curie grant agreement No. 801522, by Science Foundation Ireland and co-funded by the European Regional Development Fund through the ADAPT Centre for Digital Content Technology [grant number 13/RC/2106] and Ordnance Survey Ireland. Publisher Copyright: Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).; 15th International Workshop on Ontology Matching, OM 2020 ; Conference date: 02-11-2020.

[50] W.L. Bethea, C. Fink and J. Beecher-Deighan, JHU/APL Onto-Mapology Results for OAEI 2006, in: *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006*, P. Shvaiko, J. Euzenat, N.F. Noy, H. Stuckenschmidt, V.R. Benjamins and M. Uschold, eds, CEUR Workshop Proceedings, Vol. 225, CEUR-WS.org, 2006. https://ceur-ws.org/Vol-225/paper13.pdf.

[51] C.-C. Kiu and C.-S. Lee, OntoDNA: Ontology Alignment Results for OAEI 2007, in: *Proceedings of the 2nd International Conference on Ontology Matching - Volume 304*, OM'07, CEUR-WS.org, Aachen, DEU, 2007, pp. 196–205–.

[52] F. Hamdi, H. Zargayouna, B. Safar and C. Reynaud, TaxoMap in the OAEI 2008 Alignment Contest, in: *Organizational Memories*, 2008. https://api.semanticscholar.org/CorpusID:2833949.

[53] I.F. Cruz, F.P. Antonelli and C. Stroe, Efficient Selection of Mappings and Automatic Quality-Driven Combination of Matching Methods, in: *Proceedings of the 4th International Conference on Ontology Matching - Volume 551*, OM'09, CEUR-WS.org, Aachen, DEU, 2009, pp. 49–60–.

[54] J. Huber, T. Sztyler, J. Noessner and C. Meilicke, CODI: Combinatorial optimization for data integration - Results for OAEI 2011, 2011.

[55] S. Massmann, S. Raunich, D. Aumüller, P. Arnold and E. Rahm, Evolution of the COMA Match System, in: *Proceedings of the 6th International Conference on Ontology Matching - Volume 814*, OM'11, CEUR-WS.org, Aachen, DEU, 2011, pp. 49–60–.

[56] F. Schadd and N. Roos, Coupling of WordNet Entries for Ontology Mapping using Virtual Documents, 2012.

[57] A. Groß, M. Hartung, T. Kirsten and E. Rahm, GOMMA Results for OAEI 2012, in: *Proceedings of the 7th International Conference on Ontology Matching - Volume 946*, OM'12, CEUR-WS.org, Aachen, DEU, 2012, pp. 133–140–.

[58] S. Hertling, Hertuda results for OEAI 2012, in: *OM 2012 : Proceedings of the 7th International Workshop on Ontology Matching (OM-2012), Boston, MA, USA, November 11, 2012*, Vol. 946, P. Shvaiko, ed., RWTH Aachen, Aachen, Germany, 2012, pp. 141–144.

[59] H. Paulheim, WeSeE-Match Results for OEAI 2012, in: *Proceedings of the 7th International Conference on Ontology Matching - Volume 946*, OM'12, CEUR-WS.org, Aachen, DEU, 2012, pp. 213–219–.

[60] S. Hertling and H. Paulheim, WikiMatch: Using Wikipedia for Ontology Matching, in: *Proceedings of the 7th International Conference on Ontology Matching - Volume 946*, OM'12, CEUR-WS.org, Aachen, DEU, 2012, pp. 37–48–.

[61] M. Gulic and B. Vrdoljak, CroMatcher - Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 117–122–.

[62] W.A. Khan, M.B. Amin, A.M. Khattak, M. Hussain and S. Lee, System for Parallel Heterogeneity Resolution (SPHeRe) Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, DEU, 2013, pp. 184–189–.

[63] M. Karpathiotaki, K. Dogani and M. Koubarakis, Enabling semantic search for EO products: an ontology matching approach, in: *Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014*, P. Shvaiko, J. Euzenat, M. Mao, E. Jiménez-Ruiz, J. Li and A. Ngonga, eds, CEUR Workshop Proceedings, Vol. 1317, CEUR-WS.org, 2014, pp. 186–187. https://ceur-ws.org/Vol-1317/om2014_poster9.pdf.

[64] S. Damak, H. Souid, M. Kachroudi and S. Zghal, EXONA results for OAEI 2015, in: *Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 1545, CEUR-WS.org, 2015, pp. 145–149. https://ceur-ws.org/Vol-1545/oaei15_paper5.pdf.

[65] A. Khiat and M. Benaissa, InsMT+ results for OAEI 2015 instance matching, in: *Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 1545, CEUR-WS.org, 2015, pp. 158–161. https://ceur-ws.org/Vol-1545/oaei15_paper7.pdf.

[66] Z. Wang, X. Zhang, L. Hou, Y. Zhao, J.-Z. Li, Y. Qi and J. Tang, RiMOM results for OAEI 2016, in: *OM@ISWC*, 2010. https://api.semanticscholar.org/CorpusID:4341309.

[67] M. Achichi, Z. Bellahsene and K. Todorov, Legato results for OAEI 2017, in: *Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 2032, CEUR-WS.org, 2017, pp. 146–152. https://ceur-ws.org/Vol-2032/oaei17_paper6.pdf.

[68] C. Quix, S. Geisler, R. Hai and S. Alekh, Ontology matching for patent classification, in: *Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 2032, CEUR-WS.org, 2017, pp. 37–48. https://ceur-ws.org/Vol-2032/om2017_Tpaper4.pdf.

[69] M. Zhao, S. Zhang, W. Li and G. Chen, Matching biomedical ontologies based on formal concept analysis, *Journal of biomedical semantics* **9** (2018), 11. doi:10.1186/s13326-018-0178-9.

[70] F.J.Q. Real, G. Bella, F. McNeill and A. Bundy, Using Domain Lexicon and Grammar for Ontology Matching, 2020.

[71] K. Georgala, M. Röder, M.A. Sherif and A.-C. Ngonga Ngomo, Applying edge-counting semantic similarities to Link Discovery: Scalability and Accuracy, in: *Proceedings of Ontology Matching Workshop 2020*, 2020. https://papers.dice-research.org/2020/OM_hECATE/public.pdf.

[72] M. Dhouib, C. Faron-Zucker and A. Tettamanzi, Measuring Clusters of Labels in an Embedding Space to Refine Relations in Ontology Alignment, *Journal on Data Semantics* **10** (2021). doi:10.1007/s13740-021-00137-8.

[73] Z. Lv and R. Peng, A novel periodic learning ontology matching model based on interactive grasshopper optimization algorithm, *Knowledge-Based Systems* **228** (2021), 107239. doi:https://doi.org/10.1016/j.knosys.2021.107239. https://www.sciencedirect.com/science/article/pii/S0950705121005013.

[74] O. Fallatah, Z. Zhang and F. Hopfgartner, KGMatcher+ results for OAEI 2022, in: *OM@ISWC*, 2021. https://api.semanticscholar.org/CorpusID:240427751.

[75] J.R. Gomes, A.L. Gançarski and P.R. Henriques, OMT, a Web-Based Tool for Ontology Matching, in: *Slate*, 2022. https://api.semanticscholar.org/CorpusID:251107396.

[76] Z. Zheng, Y.-C. Zhou, X.-Z. Lu and J.-R. Lin, Knowledge-informed semantic alignment and rule interpretation for automated compliance checking, *Automation in Construction* **142** (2022), 104524. doi:https://doi.org/10.1016/j.autcon.2022.104524. https://www.sciencedirect.com/science/article/pii/S0926580522003971.

[77] A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste and F. Gargouri, The Impact of Imbalanced Training Data on Local Matching Learning of Ontologies, in: *Business Information Systems*, W. Abramowicz and R. Corchuelo, eds, Springer International Publishing, Cham, 2019, pp. 162–175. ISBN 978-3-030-20485-3.

[78] S. Efeoglu, GraphMatcher: a graph representation learning approach for ontology matching, in: *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, held as a virtual conference, October 23, 2022*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 3324, CEUR-WS.org, 2022, pp. 174–180. http://ceur-ws.org/Vol-3324/oaei22_paper7.pdf.

[79] Y.R. Jean-Mary, E.P. Shironoshita and M.R. Kabuka, Ontology Matching with Semantic Verification, *Web Semant.* **7**(3) (2009), 235–251–. doi:10.1016/j.websem.2009.04.001.

[80] G. Diallo and M. Ba, Effective method for large scale ontology matching, *CEUR Workshop Proceedings* **952** (2012).

[81] A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste and F. Gargouri, POMap: An Effective Pairwise Ontology Matching System, in: *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - (Volume 2), Funchal, Madeira, Portugal, November 1-3, 2017*, D. Aveiro, J.L.G. Dietz and J. Filipe, eds, SciTePress, 2017, pp. 161–168. ISBN 978-989-758-272-1. doi:10.5220/0006492201610168.

[82] M. Zhao and S. Zhang, Identifying and validating ontology mappings by formal concept analysis., in: *OM@ ISWC*, 2016, pp. 61–72.

[83] M. Zhao and S. Zhang, FCA-Map results for OAEI 2016., in: *OM@ ISWC*, 2016, pp. 172–177.

[84] Q. Cheng, O. Ursu, T.I. Oprea and S. Schurer, Learning reference-enriched approach towards large scale active ontology alignment and integration, in: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017, pp. 1658–1663. doi:10.1109/BIBM.2017.8217908.

[85] X. Lyu, Q. Zhang, W. Hu, Z. Sun and Y. Qu, njuLink: results for instance matching at OAEI 2017, in: *Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 2032, CEUR-WS.org, 2017, pp. 158–165. https://ceur-ws.org/Vol-2032/oaei17_paper8.pdf.

[86] F. Giunchiglia, P. Shvaiko and M. Yatskevich, S-Match: an Algorithm and an Implementation of Semantic Matching, in: *The Semantic Web: Research and Applications*, C.J. Bussler, J. Davies, D. Fensel and R. Studer, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 61–75. ISBN 978-3-540-25956-5.

[87] F. Gosselin and A. Zouaq, SEBMatcher results for OAEI 2022, in: *Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, held as a virtual conference, October 23, 2022*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh and C. Trojahn, eds, CEUR Workshop Proceedings, Vol. 3324, CEUR-WS.org, 2022, pp. 202–209. https://ceur-ws.org/Vol-3324/oaei22_paper12.pdf.

[88] W.E. Djeddi and T. Khadir, A Dynamic Multistrategy Ontology Alignment Framework Based on Semantic Relationship using WordNet, in: *Conférence Internationale sur l'Informatique et ses Applications*, 2011. https://api.semanticscholar.org/CorpusID:7006468.

[89] Y. Jean-Mary, E. Shironoshita and M. Kabuka, Ontology Matching with Semantic Verification, *SSRN Electronic Journal* (2009). doi:10.2139/ssrn.3199431.

[90] A. Annane and Z. Bellahsene, GBKOM: A generic framework for BK-based ontology matching, *Journal of Web Semantics* **63** (2020), 100563. doi:https://doi.org/10.1016/j.websem.2020.100563. https://www.sciencedirect.com/science/article/pii/S1570826820300111.

[91] A.V. Aho, M.S. Lam, R. Sethi and J.D. Ullman, *Compilers: Principles, Techniques, and Tools (2nd Edition)*, Addison-Wesley Longman Publishing Co., Inc., USA, 2006. ISBN 0321486811.

[92] I. Spasić, P. Corcoran, A. Gagarin and A. Buerki, Head to Head: Semantic Similarity of Multi–Word Terms, *IEEE Access* **6** (2018), 20545–20557. doi:10.1109/ACCESS.2018.2826224.

[93] M.A.N. Pour, A. Algergawy, P. Buche, L.J. Castro, J. Chen, A. Coulet, J. Cufi, H. Dong, O. Fallatah, D. Faria, I. Fundulaki, S. Hertling, Y. He, I. Horrocks, M. Huschka, L. Ibanescu, D.S. Jain, E. Jiménez-Ruiz, N. Karam, P. Lambrix, H. Li, Y. Li, P. Monnin, E. Nasr, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, G. Sousa, C. Trojahn, J. Vataščinová, M. Wu, B. Yaman, O. Zamazal and L. Zhou, Results of the Ontology Alignment Evaluation Initiative 2023, in: *18th International Workshop on Ontology Matching (OM 2023)*, CEUR workshop proceedings, Vol. 3591, CEUR-WS.org, Athens, Greece, 2023, pp. 97–139, collocated with the 22nd International Semantic Web Conference ISWC-2023 November 7th, 2023, Athens, Greece. https://inria.hal.science/hal-04366893.

[94] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard and D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, in: *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

[95] L.M. Surhone, M.T. Timpledon and S.F. Marseken, *SimMetrics*, VDM Publishing, 2010, p. 100. ISBN 6132113266, 9786132113269.

[96] J.J. Lastra-Díaz, A. García-Serrano, M. Batet, M. Fernández and F. Chirigati, HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset, *Inf. Syst.* **66**(C) (2017), 97–118. doi:10.1016/j.is.2017.02.002.

[97] C. Fellbaum (ed.), *WordNet: An electronic lexical database*, MIT Press, 1998, p. 423.

[98] J. David, J. Euzenat, F. Scharffe and C. Trojahn dos Santos, The Alignment API 4.0, *Semant. web* **2**(1) (2011), 3–10. http://dl.acm.org/citation.cfm?id=2019470.2019474.

[99] J. Silva, F. Baião and K. Revoredo, ALIN Results for OAEI 2016, in: *OM-2016: Proceedings of the Eleventh International Workshop on Ontology Matching*, OM'16, 2016, pp. 130–137.

[100] J. Silva, F. Baião and K. Revoredo, ALIN Results for OAEI 2017, in: *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, OM'17, 2017, pp. 114–121.

[101] J. Silva, F. Baião and K. Revoredo, ALIN Results for OAEI 2018, in: *Ontology Matching: OM-2018: Proceedings of the ISWC Workshop*, OM'18, 2018, pp. 117–124.

[102] J. Silva, C. Delgado, K. Revoredo and F. Baião, ALIN Results for OAEI 2019, in: *Proceedings of the 14th International Workshop on Ontology Matching*, OM'19, 2019, pp. 94–100.

[103] J. Silva, C. Delgado, K. Revoredo and F. Baião, ALIN Results for OAEI 2020, in: *Proceedings of the 15th International Workshop on Ontology Matching*, OM'20, 2020, pp. 139–146.

[104] J. Silva, , K. Revoredo, F. Baião and C. Lima, ALIN Results for OAEI 2021, in: *Proceedings of the 16th International Workshop on Ontology Matching*, OM'21, 2021, pp. 109–116.

[105] J. Silva, , K. Revoredo, F. Baião and C. Lima, ALIN Results for OAEI 2022, in: *Proceedings of the 17th International Workshop on Ontology Matching*, OM'22, 2022, pp. 129–136.

[106] J. Silva, , K. Revoredo, F. Baião and C. Lima, ALIN Results for OAEI 2023, in: *Proceedings of the 18th International Workshop on Ontology Matching*, OM'23, 2023, pp. 140–145.

[107] D. Faria, Using the SEALS Client's Oracle in Interactive Matching, 2016, pp. 1–3. https://github.com/DanFaria/OAEI{_}SealsClient/blob/master/OracleTutorial.pdf.

[108] M. Cheatham and P. Hitzler, String Similarity Metrics for Ontology Alignment, in: *Proceedings of the 12th International Semantic Web Conference - Part II*, ISWC '13, Springer-Verlag New York, Inc., New York, NY, USA, 2013, pp. 294–309. ISBN 978-3-642-41337-7.

[109] A. Algergawy, M. Cheatham, D. Faria, A. Ferrara, I. Fundulaki, I. Harrow, S. Hertling, E. Jiménez-Ruiz, N. Karam, A. Khiat, P. Lambrix, H. Li, S. Montanelli, H. Paulheim, C. Pesquita, T. Saveta, D. Schmidt, P. Shvaiko, A. Splendiani, E. Thiéblin, C. Trojahn, J. Vataščinová, O. Zamazal and L. Zhou, Results of the ontology alignment evaluation initiative 2018, *CEUR Workshop Proceedings* **2288** (2018), 76–116. http://www.dit.unitn.it/~pavel/om2018/papers/oaei18_paper0.pdf.

[110] M.A.N. Pour, A. Algergawy, F. Amardeilh, R. Amini, O. Fallatah, D. Faria, I. Fundulaki, I. Harrow, S. Hertling, P. Hitzler, M. Huschka, L. Ibanescu, E. Jiménez-Ruiz, N. Karam, A. Laadhar, P. Lambrix, H. Li, Y. Li, F. Michel, E. Nasr, H. Paulheim, C. Pesquita, J. Portisch, C. Roussey, T. Saveta, P. Shvaiko, A. Splendiani, C. Trojahn, J. Vataščinová, B. Yaman, O. Zamazal and L. Zhou, Results of the ontology alignment evaluation initiative 2021, *CEUR Workshop Proceedings* **3063** (2021), 62–108. https://ceur-ws.org/Vol-3063/oaei21_paper0.pdf.

[111] H. Paulheim, S. Hertling and D. Ritze, Towards Evaluating Interactive Ontology Matching Tools, *Lecture Notes in Computer Science* **7882** (2013), 31–45. doi:10.1007/978-3-642-38288-8_3.

[112] L. Chunhua, C. Zhiming, Z. Pengpeng, W. Jian, X. Jie and H. Tianxu, Improving ontology matching with propagation strategy and user feedback, in: *Seventh International Conference on Digital Image Processing (ICDIP 2015)*, Vol. 9631, 2015, p. 6. doi:10.1117/12.2197167.

[113] O. Šváb-Zamazal and V. Svátek, Analysing Ontological Structures through Name Pattern Tracking, in: *Knowledge Engineering: Practice and Patterns*, A. Gangemi and J. Euzenat, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 213–228. ISBN 978-3-540-87696-0.

[114] Y.R. Jean-Mary, E.P. Shironoshita and M.R. Kabuka, Ontology matching with semantic verification, *Web Semantics: Science, Services and Agents on the World Wide Web* **7**(3) (2009), 235–251, The Web of Data. doi:https://doi.org/10.1016/j.websem.2009.04.001. http://www.sciencedirect.com/science/article/pii/S1570826809000146.

[115] E.G.M. Petrakis, G. Varelas, A. Hliaoutakis and P. Raftopoulou, Design and Evaluation of Semantic Similarity Measures for Concepts Stemming from the Same or Different Ontologies object instrumentality, *Proceedings of the 4th Workshop on Multimedia Semantics (WMS)* **4** (2006), 233–237.

[116] P. Lambrix and R. Kaliyaperumal, A Session-based Ontology Alignment Approach enabling User Involvement, *Semantic Web* **1** (2016), 1–28.

[117] E. Jiménez-Ruiz and B.C. Grau, LogMap: Logic-Based and Scalable Ontology Matching, in: *The Semantic Web – ISWC 2011*, L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. Noy and E. Blomqvist, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 273–288. ISBN 978-3-642-25073-6.

[118] V. Lopes, F. Baião and K. Revoredo, Alinhamento Interativo de Ontologias Uma Abordagem Baseada em Query-by-Committee, Master's thesis, UNIRIO, 2015.

[119] H. Paulheim and S. Hertling, WeSeE-match Results for OAEI 2013, in: *Proceedings of the 8th International Conference on Ontology Matching - Volume 1111*, OM'13, CEUR-WS.org, Aachen, Germany, Germany, 2013, pp. 197–202. http://dl.acm.org/citation.cfm?id=2874493.2874513.

[120] S. Hertling, Hertuda Results for OAEI 2012, in: *Proceedings of the 7th International Conference on Ontology Matching - Volume 946*, OM'12, CEUR-WS.org, Aachen, Germany, Germany, 2012, pp. 141–144. http://dl.acm.org/citation.cfm?id=2887596.2887607.

[121] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I.F. Cruz and F.M. Couto, The AgreementMakerLight Ontology Matching System, in: *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. De Leenheer and D. Dou, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 527–541. ISBN 978-3-642-41030-7.

[122] S. Duan, A. Fokoue and K. Srinivas, One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback, in: *The Semantic Web – ISWC 2010*, P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks and B. Glimm, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 177–192. ISBN 978-3-642-17746-0.

[123] F. Shi, J. Li, J. Tang, G. Xie and H. Li, Actively Learning Ontology Matching via User Interaction, in: *The Semantic Web - ISWC 2009*, A. Bernstein, D.R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta and K. Thirunarayan, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 585–600. ISBN 978-3-642-04930-9.

[124] To H., R. Ichise and H. Le, An Adaptive Machine Learning Framework with User Interaction for Ontology Matching, in: *Proc. Twenty-first International Joint Conference on Artificial Intelligence*, 2009.

[125] I.F. Cruz, C. Stroe and M. Palmonari, Interactive User Feedback in Ontology Matching Using Signature Vectors, in: *28th IEEE International Conference on Data Engineering*, 2012, pp. 1321–1324. ISSN 1063-6382. doi:10.1109/ICDE.2012.137.

[126] W.E. Djeddi and M.T. Khadir, A Novel Approach Using Context-Based Measure for Matching Large Scale Ontologies, in: *Data Warehousing and Knowledge Discovery*, L. Bellatreche and M.K. Mohania, eds, Springer International Publishing, Cham, 2014, pp. 320–331. ISBN 978-3-319-10160-6.

[127] I.F. Cruz, F. Loprete, M. Palmonari, C. Stroe and A. Taheri, Pay-As-You-Go Multi-user Feedback Model for Ontology Matching, in: *Knowledge Engineering and Knowledge Management*, K. Janowicz, S. Schlobach, P. Lambrix and E. Hyvönen, eds, Springer International Publishing, Cham, 2014, pp. 80–96. ISBN 978-3-319-13704-9.

[128] B. Balasubramani, A. Taheri and I. Cruz, User involvement in ontology matching using an online active learning approach, in: *CEUR Workshop Proceedings*, Vol. 1545, CEUR-WS, 2015, pp. 45–49.

[129] D. Ngo, Z. Bellahsene and K. Todorov, Opening the Black Box of Ontology Matching, in: *The Semantic Web: Semantics and Big Data*, P. Cimiano, O. Corcho, V. Presutti, L. Hollink and S. Rudolph, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 16–30. ISBN 978-3-642-38288-8.

[130] J. David, J. Euzenat, F. Scharffe and C. Trojahn dos Santos, The Alignment API 4.0, *Semant. web* **2**(1) (2011), 3–10. http://dl.acm.org/citation.cfm?id=2019470.2019474.

[131] D. Gale and L.S. Shapley, College Admissions and the Stability of Marriage, *The American Mathematical Monthly* **69**(1) (1962), 9–15. doi:10.1080/00029890.1962.11989827.

[132] R.W. Irving, D.F. Manlove and G. O'Malley, Stable marriage with ties and bounded length preference lists, *Journal of Discrete Algorithms* **7**(2) (2009), 213–219, Selected papers from the 2nd Algorithms and Complexity in Durham Workshop ACiD 2006. doi:https://doi.org/10.1016/j.jda.2008.09.003. http://www.sciencedirect.com/science/article/pii/S1570866708000683.

[133] J. Da Silva, K. Revoredo, F.A. Baião and J. Euzenat, Semantic interactive ontology matching: Synergistic combination of techniques to improve the set of candidate correspondences, in: *OM 2017 - 12th ISWC workshop on ontology matching*, Vol. 2032, 2017, pp. 13–24. ISSN 16130073. http://disi.unitn.it/~pavel/om2017/papers/om2017_Tpaper2.pdf.

[134] J. Da Silva, K. Revoredo, F.A. Baião and J. Euzenat, Interactive Ontology Matching: Using Expert Feedback to Select Attribute Mappings, 2018. http://disi.unitn.it/~pavel/om2018/papers/om2018_LTpaper3.pdf.

[135] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, E. Jimenez-Ruiz, K. Kolthoff, E. Kuss, P. Lambrix, H. Leopold, H. Li, C. Meilicke, M. Mohammadi, S. Montanelli, C. Pesquita, T. Saveta, P. Shvaiko, A. Splendiani, H. Stuckenschmidt, E. Thieblin, K. Todorov, C. Trojahn and O. Zamazal, Results of the Ontology Alignment Evaluation Initiative 2017, in: *Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017) Vienna, Austria, October 21st, 2017.*, 2017.

[136] A. Groß, M. Hartung, T. Kirsten and E. Rahm, GOMMA Results for OAEI 2012, in: *Proceedings of the 7th International Conference on Ontology Matching - Volume 946*, OM'12, CEUR-WS.org, Aachen, Germany, Germany, 2012, pp. 133–140. http://dl.acm.org/citation.cfm?id=2887596.2887606.

[137] W.E. Djeddi, M.T. Khadir and S.B. Yahia, XMap : Results for OAEI 2017, in: *CEUR Workshop Proceedings*, Vol. 2032, 2017, pp. 196–200. ISSN 16130073.

[138] E. Jimènez-Ruiz, B.C. Grau and V. Cross, LogMap family participation in the OAEI 2017, in: *CEUR Workshop Proceedings*, Vol. 2032, 2017, pp. 153–157. ISSN 16130073.

[139] D. Faria, B.S. Balasubramani, V.R. Shivaprabhu, I. Mott, C. Pesquita, F.M. Couto and I.F. Cruz, Results of AML in OAEI 2017, in: *CEUR Workshop Proceedings*, Vol. 2032, 2017, pp. 122–128. ISSN 16130073.

[140] A. Guedes, F. Baião and K. Revoredo, On the Identification and Representation of Ontology Correspondence Antipatterns, in: *Proceedings of the 8th International Workshop on Modular Ontologies co-located with the 8th International Conference on Formal Ontology in Information Systems (FOIS 2014), CEUR Workshop Proceedings*, Vol. 1248, 2014. ISSN 1613-0073.

[141] A. Guedes, F. Baião and K. Revoredo, Digging Ontology Correspondence Antipatterns, in: *Proceedings of the 5th Workshop on Ontology and Semantic Web Patterns (WOP2014) co-located with the 13th International Semantic Web Conference (ISWC 2014)*, Vol. 1032, 2014, pp. 38—48.

[142] Y.R. Jean-Mary, E.P. Shironoshita and M.R. Kabuka, Ontology matching with semantic verification, 2009.

[143] N.J. Belkin, R. Bierig, G. Buscher, J. Jose and J. Teevan, Understanding the user - logging and interpreting user interactions in information search and retrieval, *Information Retrieval* (2010), 2009–2010.

[144] M.A.N. Pour, A. Algergawy, P. Buche, L.J. Castro, J. Chen, H. Dong, O. Fallatah, D. Faria, I. Fundulaki, S. Hertling, Y. He, I. Horrocks, M. Huschka, L. Ibanescu, E. Jiménez-Ruiz, N. Karam, A. Laadhar, P. Lambrix, H. Li, Y. Li, F. Michel, E. Nasr, H. Paulheim, C. Pesquita, T. Saveta, P. Shvaiko, C. Trojahn, C. Verhey, M. Wu, B. Yaman, O. Zamazal and L. Zhou, Results of the ontology alignment evaluation initiative 2022, *CEUR Workshop Proceedings* **3324** (2022), 84–128. http://disi.unitn.it/~pavel/om2022/papers/oaei22_paper0.pdf.

[145] M.T. Cabré Castellví, *Terminology: Theory, methods and applications*, John Benjamins, 1999. https://www.jbe-platform.com/content/books/9789027298652.

[146] ISO, ISO 704. Terminology Work - Principles and Methods, 2000.

[147] C. Roche, Ontoterminology: How to unify terminology and ontology into a single paradigm, in: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 2626–2630. http://www.lrec-conf.org/proceedings/lrec2012/pdf/567_Paper.pdf.

[148] D.G. Maynard, TERM RECOGNITION USING COMBINED KNOWLEDGE SOURCES, 2000.

[149] C. Jacquemin and J. Royauté, Retrieving Terms and their Variants in a Lexicalized Unification-Based Framework, 1994, pp. 132–141. ISBN 978-3-540-19889-5. doi:10.1007/978-1-4471-2099-$5_1 4$.

[150] A. Lauriston, Criteria for Measuring Term Recognition, in: *EACL*, 1995.

[151] P. Jackson, Natural Language Processing for Online Applications : Text Retrieval, Extraction and Categorization / P. Jackson, I. Moulinier., *Computational Linguistics* **29** (2003), 510–511. doi:10.1075/nlp.5.

[152] D. Gasevic, D. Djuric and V. Devedzic, *Model Driven Architecture and Ontology Development*, 2006. ISBN 978-3-540-32180-4. doi:10.1007/3-540-32182-9.

[153] J.F. Sowa, *Knowledge representation : logical, philosophical, and computational foundations*, 4th prin edn, Brooks Cole, 2000. ISBN 9780534949655,0534949657. http://gen.lib.rus.ec/book/index.php?md5=CD48FB624BE49BAA0059562FAA53F6EC.

[154] A. Olivé, *Conceptual Modeling of Information Systems*, 2007. ISBN 978-3-540-39389-4. doi:10.1007/978-3-540-39390-0.

[155] M.J.J. Rosse C, A reference ontology for biomedical informatics: the Foundational Model of Anatomy, *J Biomed Inform.;36(6)* (2003), 478–500. doi:10.1016/j.jbi.2003.11.007.

[156] M.Z. Kurdi, *Natural Language Processing and Computational Linguistics 1*, 2016, p. 277. ISBN 978-1-84821-848-2.

[157] A. Laadhar, F. Ghozzi, I. Megdiche, F. Ravat, O. Teste and F. Gargouri, POMap results for OAEI 2017, in: *Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017*, P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, M. Cheatham and O. Hassanzadeh, eds, CEUR Workshop Proceedings, Vol. 2032, CEUR-WS.org, 2017, pp. 171–177. https://ceur-ws.org/Vol-2032/oaei17_paper10.pdf.

[158] P. Kolyvakis, A. Kalousis, B. Smith and D. Kiritsis, Biomedical ontology alignment: an approach based on representation learning, *Journal of Biomedical Semantics* **9** (2018). https://api.semanticscholar.org/CorpusID:52012682.