Extracting Named-Entities and their Relationships

Elias Oliveira[1], Gabriel Dias[2], Jaimel Lima[1], Juliana Pirovani[2]

[1] Laboratório de Computação de Alto Desempenho (LCAD)
Universidade Federal do Espírito Santo
Av Fernando Ferrari, 514, Goiabeiras – Vitória, ES 29075-910
http://www.lcad.inf.ufes.br/
{elias,juliana}@lcad.inf.ufes.br
[2] Departamento de Computação
Universidade Federal do Espírito Santo
Alto Universitário, s/n – Guararema
Alegre, ES 29500-000 – Brasil

Abstract.

Extracting named entities from an unstructured text is a good form to build knowledge for conversational intelligent systems. Named Entity Recognition aims to automatically identify and classify entities like persons, places, organizations, and so forth. In addition, Named Entity Recognition is also a fundamental step for relations extraction. However, both problems are hard to solve, as several categories of named entities are similarly written and appear in cognate contexts. To accomplish it, some hybrid approaches combining machine learning and expert linguistic tailored models are usually used. In this current study, we turn our focus onto the expert linguistic flavor by applying Local Grammar and Cascade of Transducers. Local Grammars are to represent the rules of a particular linguistic structure. They are often built manually to describe the entities and relations we aim to recognize. In our study, we adapted a Local Grammar to improve the Recognition of Named Entities. The results show an improvement of up to 7% on the F-measure metric in relation to the previous Local Grammar. Besides, we built another Local Grammar to recognize binary relationships between person and person linked by parenthood and person and localization linked by a recognized place of birth from the improved Local Grammar. Finally, we present practical applications for a conversational system using Prolog for inferring over the extracted entities and relations.

Categories and Subject Descriptors: H.1.2 [**Information Systems**]: User/Machine Systems; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; D.1.6 [**Programming Techniques**]: Logic Programming

Keywords: Named-Entity Recognition, Information Extraction, Artificial Intelligence

1. INTRODUCTION

A large volume of information these days is available in unstructured free-texts in various sorts of documents, such as a) personals; b) journalistic; c) officials or even d) governmental; and e) in social media. For some automatic applications, in the area of artificial intelligence, it is crucial to extract valuable meta-information from these textual documents. To do this, the Named Entity Recognition (NER) and some other more general Information Extraction (IE) tasks, all of them part of the Natural Language Processing (NLP) area, are important tools. One task within IE is the Relation Extraction (RE) [Lima et al. 2020]. We can find these relations connected to a named entity, or between two different named-entities. NER aims to identify and classify entities automatically in free written texts, such as names of person, places, organizations, dates, and others, depending on the relevance these domains [Pirovani and Oliveira 2021].

However, NER is not a simple task. Various categories of Named Entities (NEs) are written in a similar form and also appear in alike contexts. For example, person and places start with a capital letter, just like temporal expressions and values contain numbers. In addition, the same NE can be classified into different categories depending on the context where it appears. For instance, the NE *Washington* can refer to the name of a person in one context and one location in another. Another example is the name Rosa Margarida when searched by the *Google search engine*[1]. It gets confusing to deal with the name of a person and the name of flowers, returning thus both [Lima et al. 2020].

NER is strongly dependent on the language, the *corpus*, and the domain [Campos and Oliveira 2015]. Considering the domain dependence, the same category of NE can be written in different ways in different textual genres. For example, in e-mail texts, it is common to show names of people following words like *Olá* (*Hello* in English) and *Boa tarde* (*Good afternoon* in English), while in texts of *Portarias* (*Ordinances* in English) and *memorandos* (*Memorandums* in English) it is common to show names of people following words like *servidor* (*Oficial* in English) and *professor* (*teacher* in English). Thus, analyzing texts of different genres contributes to improving the identification of different rules for the NER and consequently for the RE, as well.

NER proves useful in several situations, for instance, the search for persons in proceedings or other types of documents, in disambiguation, as shown in the previous examples, in the extraction of relationships [Lima et al. 2020], like a family tree, *etc.*

To evaluate a NER tool, researchers usually adopt the use of benchmark *corpus* and the previous literature results for comparisons. A *corpus* may be a set of texts of a given context, or even a variety of domains, such as one or more newspapers, books, among others. HAREM [Mota and Santos 2008; Santos and Cardoso 2007] was an event organized by Linguateca [Linguateca 2018] and was an enormous incentive for NER researchers in the Portuguese language. It adopts the classification of 10 NEs categories, namely, for instance, 1) Abstraction, 2) Event, 3) Thing, 4) Location, 5) Work, 6) Organization, 7) Person, 8) Time, 9) Value, and 10) Other.

The HAREM repository provides annotated *corpora*, that is, with the NEs related to these categories already identified and classified manually. The annotated corpora used in the HAREM have been used as a golden standard reference for NER systems in Portuguese. These *corpora* are known as the Golden Collections (GC), and they are split into First HAREM, Mini HAREM, and Second HAREM.

When developing NER systems using the linguistic approach, rules are constructed manually to identify NEs. One strategy for representing the rules of the linguistic approach is the use of *Local Grammars* (LG), a formalism introduced by Maurice Gross [Gross 1997]. *Local grammars are finite-state grammars or finite-state automata that represent sets of expressions in a natural language* [Gross 1999]. These grammars are constructed manually and are a manner of grouping or capturing expressions that have common characteristics, whether they are syntactic or semantic.

Pirovani [Pirovani 2019] showed the potential of LGs for use in the NER problem. Both individually when there is no training corpus available, and in conjunction with other machine learning techniques such as Conditional Random Fields (CRF) [Lafferty et al. 2001] to improve its performance. Her work pointed out that any improvements in the LG could also bring improvements to the hybrid approaches.

Therefore, this work is an extension of the KDMILe 2021 paper [Oliveira et al. 2021b] and is part of a larger project. It has two folds. Firstly, we sought to explore different textual genres such as emails, ordinances, interviews, and many others to identify rules to be used to adapt an LG[2] for the Portuguese language. That is already an improvement in the work carried out by Pirovani [Pirovani 2019] and

[1] www.google.com
[2] <https://inf.ufes.br/~elias/dataSets/ner/recursosTese-julianaPirovani.zip>

[Pirovani and Oliveira 2021]. She built an LG only analyzing the GC of the First HAREM[3], which contains, in its majority, newspaper texts. In the current work, we will explore a Biblical document genre due to its sophisticated language. Secondly, we proposed a strategy in this work for identifying and annotating the family (fatherhood and motherhood) and place of birth relationships between pairs of NEs. The relationships we are interested in are only those described in Mathew Chapter 1 and 2. In the end, we will have the NEs annotated and a list of parentship and place of birth relationships cited in the worked documents.

As a practical application, we present these lists in the Prolog facts format to allow reasoning about them. Hence, we show that it is possible to generate the Prolog facts database using the LG built to extract the relationships and it is also possible to generate well-structured answers from Prolog facts using LG. The output is an input to another project which aims to provide a robot with an extra logical reasoning capability [Oliveira et al. 2020].

The organization of this article is as follows. In Section 2, we discuss briefly the literature review related to this work. In Section 3, we present the idea of *Local Grammar and Cascade of Transducers* and we discuss the methodology of our experiments to seek better results. The evaluations carried out and the results yielded are presented in Section 4. Our conclusions are in Section 5.

2. LITERATURE REVIEW

The NER task can encompass different approaches: linguistic [Campos and Oliveira 2015; Rocha et al. 2016], machine learning [Castro et al. 2018; Santos and Guimaraes 2015; Yang et al. 2017] or hybrid [Pirovani and Oliveira 2018; Pirovani et al. 2019], which can be understood as a combination of the two models.

In their studies, [Pirovani 2019] proposed a model for NER for Portuguese. The authors used a hybrid approach, CRF+LG. The LGs are built using the tool Unitex[4], a free software package for the construction and application of grammars and cascade. In Unitex, LGs can be represented as one or more graphs. The results of the classification carried out by LG are used as input for a CRF classifier. There is a strong relationship between this work and our study since we propose improvements in the LG proposed by the authors, including using them in cascade.

The approach proposed by [Lima et al. 2018] combines the use of ontologies and logical programming (Prolog) for NER and relation extraction. The approach allows the use of domain ontologies to extract relationships between entities. The model has components for text preprocessing, the generation of a knowledge base; the extraction of new rules; the ontology population. The relationship with our study is in the proposal to create rules in Prolog for relation extraction. The authors worked with binary relations. Among these relations, the *kill* type relates to two entities of the Person type. Our work presents, in addition to improving the LGs, a proposal for application kinship extraction.

In their work, [Parsaeimehr et al. 2020] proposed a deep learning model for NER and RE. They propose a joint model in order to reduce propagation errors in both tasks. The proposal was developed with four components: embedding layer, Bi-LSTM layer, entity type detection module, and relationship extraction module. For the incorporation layer, the authors used the Glove method. The second component is composed of a recurrent neural network (Bi-LSTM). The third component is responsible for the NER task, and the fourth and last component is responsible for the RE task, which uses a dependency tree and looks for the shortest path between two NEs. The relationship between this study and our work is in using the NER for ER, beyond grammatical structures.

In their studies, [He et al. 2019] proposed a model based on neural networks to extract named entities and kinship relationships. The approach uses a joint neural model with data from online

[3]Corpus of HAREM available at: www.linguateca.pt/HAREM/
[4]https://unitexgramlab.org/en

obituaries. In our studies, we also propose a model for extracting kinship relationships. However, our study differs from the proposed model since LG in our work is built using the structure of language and reflects human knowledge, making it possible to explain how to create the rules.

3. THE METHODOLOGY

These experiments aim to show our strategies to improve the quality of the identification of NEs and recognize the relation concept linking a pair of NEs. For the sake of illustration of our experiments, we will focus on both the annotation of places and the parenthood relations: fatherhood and motherhood.

3.1 Extracting Named Entities

Local Grammar is a linguistic model, which uses a set of rules built manually [Gross 1997]. Another concept used in our study is the Cascade of Transducers, which is a set of LGs that can be applied in a given order, so that an LG can use the results of the previous one in its rules [Paumier 2021]. The use of cascade has a proposal to solve ambiguity problems generated by LG.

The LG adapted in this work [Pirovani et al. 2019] was created in Unitex [Paumier 2021] and consists of 10 graphs, one for each of the NEs categories considered by HAREM. The graphs are to capture some simple heuristics for the recognition of NEs.

An example of rule in the graph created for the *person* category is presented in Figure 1. This graph recognizes words such as *say* (*diz, in Portuguese*) or *said* (*afirmou, in Portuguese*) followed by words with the first letter capitalized, as identified by the code `<FIRST>` in Unitex dictionaries. Among words with the first letter capitalized, prepositions may appear whose recognition has been previously detailed in graph Preposicao.grf included as subgraph. An example of occurrence identified by this graph:

afirmou `<PESSOA>` José SÓCRATES `</PESSOA>`.

Note that identified person will appear between the tags `<PESSOA>` (`<PERSON>`) and `</PESSOA>` in the text with the identified occurrences.

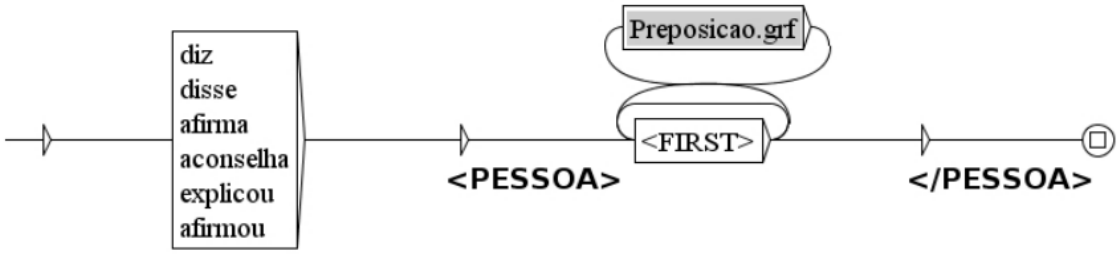

Fig. 1: Example of rule in the graph that recognizes the Person category.

After, these tags are replaced by the tags `<EM>` and `</EM>` (harem pattern) as shown in Figure 2. This is done using shell scripts. The ID *tag* in Figure 2 is a unique identification for the NE. CATEG stands for the category assigned to the recognized NE, among the ten mentioned above. TIPO (or TYPE in English) is a subdivision of the category to specify in depth what the entity represents. This attribute (TIPO) was not used in this work.

Before starting our experiments, we carried out a linguistic study to identify new rules aiming to improve the LG of Pirovani et al. [Pirovani et al. 2019]. We used the HAREM corpora: *PrimeiroHarem*,

```
<EM ID="aa56088-481" CATEG="PESSOA" TIPO="INDIVIDUAL">José SÓCRATES</EM>
```

Fig. 2: Annotated example on HAREM pattern

SegundoHarem, and *MiniHarem*; from a local newspaper *A Tribuna*, named onwards by aTribuna, several articles; from a governmental institution, documents of Ordinances, and some personal emails. Analyzing these documents, we improved the LG to surpass when the previous LG failed.

One of the first observations when working with the HAREM corpora was that we had to implement some rules including some specific words, such as job's positions and kinships, which precede people's names. Some examples of it would be: *the mayor Carlos*, *the deputy Júlia*, among many others.

Another improvement, due to the study of the used datasets, was the inclusion of a rule for the *enumeration structure* in the written language. The enumeration structure is when NEs of the same category appear in a sequence. The first one is easier to identify from the context where it appears, but its successors do not have a context from which you can classify them. For example, *Today, we will have professors João, Ricardo, and Mário at the meeting.* João is identified by context, but Ricardo and Mário would not be. Therefore, when an NE is followed by a *comma* (,) or an *e* (an *and, in English*), we assume the next NE, also starting with a capital letter, is an NE from the same class as the first NE.

In addition to these new rules, there also were improvements in the graph referring to the *Value* entity. Any number followed by *a.c.* or *d.c.* is now considered a date. Besides, a number followed by a plural noun is also marked as *Value*, for instance, *3 apples*, *2 men*, and so forth.

The latest improvement was the construction of a Cascade of Transducers using the graphs from the improved LG. We performed some tests in order to find dependencies between the graphs. As a consequence of these tests, it was possible to generate a better order to apply the graphs and take advantage of this method.

Having all NEs annotated within the texts, we can now seek for the relations between entities to annotate as well.

3.2 Extracting Relations

The identification of relations between NEs is another goal of this work, although it is still ongoing research. The ultimate goal is to provide search engines [Pirovani et al. 2018] with annotated documents with both NEs and extracted relations for their indexing processes and logical reasoning for a robot [Oliveira et al. 2020].

Our initial focus is on relations that happen only between pairs of NEs. For that, we chose to find person related to a place, fatherhood, and motherhood relationship in the Biblical book, the Gospel of Mathew[5], Chapter 1 and 2. There is plenty of that type of relations in Mathew's book, spanning up to forty-two generations from Abraham to Jesus. The book is challenging because of its formal language, in some versions, and variety of ways of expressing fatherhood. For instance, in Chapter 1, the sentence *Abraham was* **the father of** *Isaac*, in verse 2, and *Jacob* **the father of** *Joseph, the husband of Mary, and Mary was* **the mother of** *Jesus who is called the Messiah*, in verse 16, all these examples from the same a translation version in English. With respect to places, one can find the following example: *Now when Jesus* **was born in** *Bethlehem of Judaea in the days of Herod the king, behold, there came wise men from the east to Jerusalem,* Chapter 2 verse 1.

Figure 3 shows the pipeline of the process. First, the NER is performed by applying the Cascade of Transducers mentioned in the previous section, recognizing all the NEs of interest. Latter, from

[5]https://en.wikipedia.org/wiki/Gospel_of_Matthew

the chunk of sentence between a pair of NEs, we extracted the relation of interest (the parenthood or place of birth in this work). The relationships were extracted from a LG built for this which was added to the Cascade. An example of a rule in this LG is presented in Figure 4. Note that this LG uses the results of NER (`<PERSON>` captures entities of the Person category previously recognized by the Cascade). Another example of a rule in this LG is presented in Figure 5. This LG uses the results of NER (`<PERSON>` and `<PLACE>`) to extract the relationship place of birth. A syntax transformation is needed to uniforms the relational predicate along with the knowledge database. Note thus that the quality of NER plays a crucial role in the whole proposed approach.

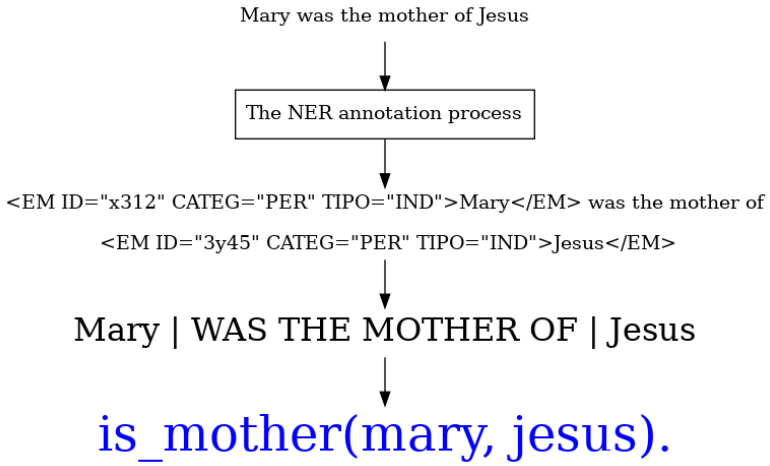

Fig. 3: The pipeline of the process

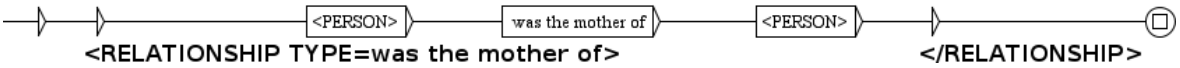

Fig. 4: Example of rule in the LG that extracts relationships between two people

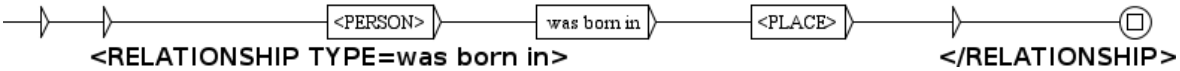

Fig. 5: Example of rule in the LG that extracts relationships between a person and a place

4. RESULTS

The results of the application of each approach in the corpora *PrimeiroHarem*, *SegundoHarem*, and *MiniHarem* are shown in Table I. In the last columns, we placed the metrics used in these evaluations: Precision, Recall, and F-measure, the average of the previous two metrics. In other corpora used in our linguistic studies, it was unable to obtain the results of the metrics adopted due to the lack of human annotations for comparison or the lack of some of the analyzed entities, those discussed in Section 1. Nevertheless, we managed to observe the behavior of the proposal LG and Cascade in that corpus.

The results of the application of the LG built in [Pirovani et al. 2019] are shown in Table I, *line 1)*. With all the changes presented in the Section 3.1, we yielded the following result shown in Table I, *line 2)*. Applying the Cascade of Transducers, we obtained the results shown in Table I, *line 3)*. Note that in the previous results, a *direct* application was used initially, without observing which order

Table I: The results of each approach

Approach	Corpus	Precision	Recall	F-Measure
1) LG of Pirovani et al. [Pirovani et al. 2019]	PrimeiroHarem	0.70	0.43	0.53
	SegundoHarem	0.70	0.39	0.50
	MiniHarem	0.67	0.41	0.51
2) Adapted LG	PrimeiroHarem	0.71	0.41	0.52
	SegundoHarem	0.71	0.45	0.55
	MiniHarem	0.68	0.42	0.52
3) Cascade of Transducers	PrimeiroHarem	0.72	0.49	0.59
	SegundoHarem	0.71	0.44	0.54
	MiniHarem	0.72	0.48	0.58

would be the best for the application of the graphs. These results with the Cascade show a gain of 7 percentage points in F-measure for the corpus Mini-HAREM in comparison to the original LG. We believe that the Cascade approach performed better because it: a) uses previous results; b) adds more contextual information in some situations; and c) prevents from some rules to generate ambiguities.

Table II shows all the basic relational genealogic facts we extracted from the Chapter of Mathew 1, after the execution of the pipeline shown in Figure 3. Analyzing the results in the cited chapter, the Cascade was able to extract all the mentioned relationships between father and mother and their respective children. In the color blue, we depicted all the motherhood we could find in the text. Just a reminder that it was unusual at that time to register facts related to any woman. The first fatherhood is in boldface, on top left of the table: **is_father(abraham, isaac).** According to the Bible, Abraham is called the father of all the Jews people. The last fatherhood, before Jesus, is given at the bottom right of the table: **is_father(joseph, jesus).** With all this information we can now infer who should be the grandfather of *Jesus*, for example.

Table II: The found list of parenthood in the book of Mathew 1

is_father(abraham,isaac).	
is_father(isaac,jacob).	is_father(jacob,judah).
is_father(judah,perez).	is_mother(tamar,perez).
is_father(judah,zerah).	is_mother(tamar,zerah).
is_father(perez,hezrom).	is_father(hezrom,ram).
is_father(ram,amminadab).	is_father(amminadab,nahshon).
is_father(nahshon,salmon).	is_father(salmon,boaz).
is_mother(rachab,boaz).	
is_father(boaz,obed).	is_mother(ruth,obed).
is_father(obed,jesse).	is_father(jesse,david).
is_father(david,solomon).	is_mother(bathsheba,solomon).
is_father(solomon,rehoboam).	is_father(rehoboam,abijah).
is_father(abijah,asa).	is_father(asa,jehoshaphat).
is_father(jehoshaphat,jehoram).	is_father(jehoram,uzziah).
is_father(uzziah,jotham).	is_father(jotham,ahaz).
is_father(ahaz,hezekian).	is_father(hezekian,manasseh).
is_father(manasseh,amon).	is_father(amon,josiah).
is_father(josiah,jeconiah).	is_father(jeconiah,shealtiel).
is_father(shealtiel,zerubbabel).	is_father(zerubbabel,abiud).
is_father(abiud,eliakim).	is_father(eliakim,azor).
is_father(azor,zadok).	is_father(zadok,akim).
is_father(akim,elihud).	is_father(elihud,eleazar).
is_father(eleazar,matthan).	is_father(matthan,jacob).
is_father(jacob,joseph).	**is_father(joseph,jesus).**
is_father(abraham,david).	is_father(david,jesus_christ).
is_mother(mary,jesus).	

In the color red, we highlighted a challenging situation that may occur in any other domain but is frequent in the Bible. Someone is said to be the father of another person figuratively. That is the case when *David* figured out as the son of *Abraham*. Someone is said to be the father of another person figuratively. That is the case when *David* is called to be the son of *Abraham*. The fact which is represented by the Prolog tuple as **is_father(abraham, david).** Almost at the end of the table, another expression, in red, saying that *David* is the father of *Jesus Christ*. There are two important things to point out in this example: a) the first is when someone else is said to be the father of *Jesus*, but b) *Jesus* is given another name, for instance, *Jesus Christ*. Would it be another person or the same *Jesus* mentioned previously? That is one of the disambiguation challenges we also need to tackle to extract the correct fact and thus be able to reason correctly over them.

Another important aspect of our approach is that there are both possible to generate the Prolog facts database using the LG built to extract the relationships (Figure 6) and the answers using LG from the Prolog facts (Figure 7). These LGs use input variables that store parts of a text sequence recognized by LG. The red parentheses define the beginning and the end of the piece of information to be stored. The name given to the variable appears above the parentheses. After defining the variable, it can be used in transducer outputs by surrounding its name with $ [Paumier 2021].

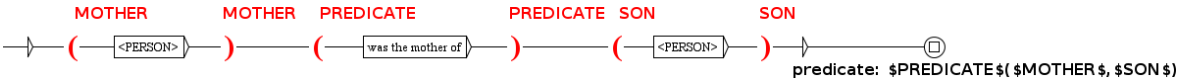

Fig. 6: LG that generates Prolog facts

Note that, in the Figure 6, the generated output (displayed in bold) after **predicate:** has the format **$PREDICATE$($MOTHER$,SON)** in which the variables **$PREDICATE$**, **$MOTHER$**, **SON** will be replaced by the predicates and mother and child names respectively captured by LG. For example, this LG generates the Prolog's predicate **was the mother of(mary,jesus)** from the sentence **mary was the mother of jesus**. Capturing the predicate in a variable allows building a more generic LG capable of generating Prolog facts for any relations recognized a priori in LG.

When applying LGs to extract patterns in a text, the outputs can be: ignored (option *are not taken into account*), appended to the concordance file (option *merge with input text*) or used to replace the recognized sequences in the concordance file (option *REPLACE recognized sequences*). The Prolog facts database can be obtained using this last option. As mentioned before, a syntax transformation is needed to uniforms the relational predicate along with the knowledge database, generating something like **is_mother(mary,jesus)**.

In order for providing any intelligent system with mechanism to answer a query, we need to equally write an additional portion of code to each Prolog predicate presented in the Table II, or any another automatically created.

We consider in our system that a query Q is transformed into many other queries. One of them is that Q' which will be tackled by the relational system [Oliveira et al. 2021a] generating an answer A and sent it to the user. The answer A is built, in a syntax-valid answer, just after the Prolog inference engine processes the transformed query Q'. The construction of this answering-sentence can be done by the use of a LG (Figure 7), tailored to the Question Answering (QA) problem.

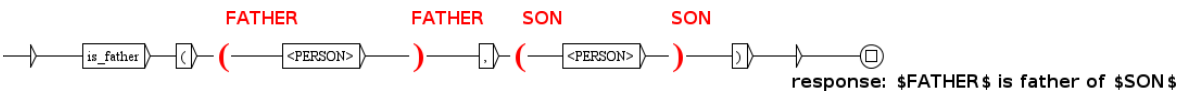

Fig. 7: LG that generates answers

Therefore, to the question Q, *Who is the father of Jesus?*, a possible syntax-valid answer A could be: a) *WP is father of Jesus*, $A+$, whether the Prolog inference engine succeeds to instantiate *WP*, or otherwise, $A-$, b) *I am so sorry, but I have no information about who the father of Jesus is.*

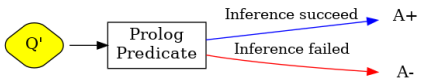

Fig. 8: Building an answer back to the query Q

Figure 7 shows an LG that generates, for example, the answer *Joseph is father of Jesus* for the Prolog's predicate **is_father(joseph,jesus)** whereas the Prolog inference engine succeeds to instantiate WP (WP = Joseph). This LG uses the variables *father* and *son* to generate the answer in the desired format to the output, appending some text to the answer whether necessary. In this particular case, only the **is_father** predicate is recognized, but similar LGs can generate responses to other Prolog facts. It is also possible to build a more generic LG that generates answers for a group of predicates provided that their structure is similar and known a priori.

The LG shown in Figure 9 generates answers to binary predicates (in this case, relationships between pairs of NEs of the Person category) composed of two words separated by _ where the answer pattern is **$PERSON1$ $WORD1$ $WORD2$ of $PERSON2$**. Considering the predicates:
is_mother(mary,jesus)
is_father(joseph,jesus)
is_brother(jhon, mary)
The $A+$ answers generated by this LG are respectively:
mary is mother of jesus
joseph is father of jesus
jhon is brother of mary

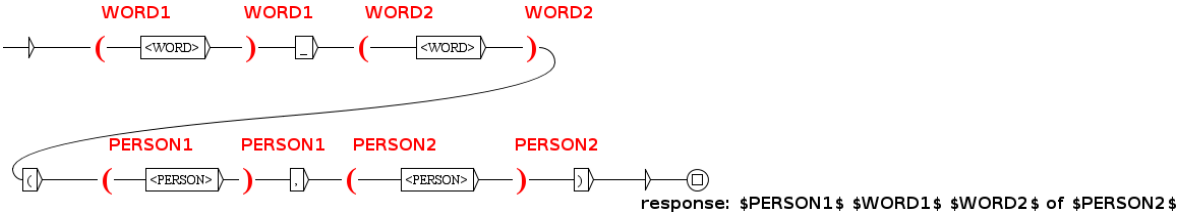

Fig. 9: A more generic LG that generates answers

5. CONCLUSIONS

This work aimed to present a methodology stemming from the linguistic procedures, which are very good for situations where we do not have a reasonable amount of training dataset for the supervised algorithms [Fonseca et al. 2018]. On the other hand, with this methodology, we greatly facilitate the rapid generation of an annotated database for then create a training dataset for the supervised algorithms [Lima et al. 2020], when that is the goal.

We describe in this work some efforts to improve the performance of an LG, on the result generated in [Pirovani et al. 2019]. The grammatical rules, now part of the LG, were built with the features of software Unitex [Paumier 2021]. These improvements also included the creation of a Cascade of Transducers. The Cascade approach performed better. We observed an improvement of up to 5% among the best versions of LG and Cascade, over the result presented in [Pirovani et al. 2019].

Besides the improvements yielded over previous results [Pirovani 2019; Pirovani et al. 2019], we also presented in this paper the use of LG to extract both the birthplace and the parentship between pairs of NEs.

In order to illustrate the execution of the pipeline proposed in this work, we focused only on the Biblical book of Mathew, Chapter 1 and 2. Nevertheless, the same applies to any other document. We generate a list of birthplaces and parentship in the Prolog facts format to allow other applications [Oliveira et al. 2020] of reasoning over these logical facts.

This work is ongoing long-term research on understanding the use of LG as a resort when there are no training datasets available, but also to improve supervised approaches. Another goal, as future work, is to use the annotated NEs both to create more sophisticated logical-knowledge datasets, and to generate exam questions, as proposed in [Pirovani et al. 2017].

REFERENCES

Campos, J. and Oliveira, E. Extração de Nomes de Pessoas em Textos em Português: uma Abordagem Usando Gramáticas Locais. In *Computer on the Beach 2015*. SBC, Florianópolis, SC, 2015.

Castro, P. V. Q., Silva, N. F. F., and Soares, A. S. Portuguese Named Entity Recognition Using LSTM-CRF. In *Villavicencio A. et al. (eds) Computational Processing of the Portuguese Language. PROPOR 2018. Lecture Notes in Computer Science, vol 11122.* Springer, Cham, Canela, RS, pp. 83–92, 2018.

Fonseca, E., Medeiros, I., Kamikawachi, D., and Bokan, A. Automatically Grading Brazilian Student Essays. In *International Conference on Computational Processing of the Portuguese Language*. Springer, pp. 170–179, 2018.

Gross, M. The Construction of Local Grammars. *In ROCHE, E.; SCHABES, Y. (eds.). Finite-State Language Processing, Language, Speech, and Communication, Cambridge, Mass.*, 1997.

Gross, M. A Bootstrap Method for Constructing Local Grammars. In *Proceedings of the Symposium on Contemporary Mathematics*. University of Belgrad, pp. 229–250, 1999.

He, K., Wu, J., Ma, X., Zhang, C., Huang, M., Li, C., and Yao, L. Extracting Kinship from Obituary to Enhance Electronic Health Records for Genetic Research. In *Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task*. pp. 1–10, 2019.

Lafferty, J., McCallum, A., and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001*. Vol. 1. pp. 282–289, 2001.

Lima, J., Colombo, C., Izo, F., Oliveira, E., and Badue, C. Finding Entities and Related Facts in Newspaper. In *20$^{\underline{th}}$ International Conference on Intelligent Systems Design and Applications – (ISDA)*. Springer, Springer International Publishing, On the WWW, pp. 102–113, 2020.

Lima, J., Colombo, C., Izo, F., Pirovani, J. C. P., and Oliveira, E. Using CRF+LG for Automated Classification of Named Entities in Newspaper Texts. In *Computing Conference (CLEI), 2020 Latin American*. IEEE, Loja, Ecuador, pp. 27–32, 2020.

Lima, R., Espinasse, B., and Freitas, F. OntoILPER: an Ontology and Inductive Logic Programming-Based System to Extract Entities and Relations from Text. *Knowledge and Information Systems* 56 (1): 223–255, 2018.

Linguateca., 2018. Acesso em: 17/06/2021.

Mota, C. and Santos, D. *Desafios na Avaliação Conjunta do Reconhecimento de Entidades Mencionadas: O Segundo HAREM*. Linguateca, 2008.

Oliveira, E., Dias, G., Lima, J., and Pirovani, J. Using Relational Inference Engine to Answer Questions. In *IV Latin American Conference on Learning Analytics – (LALA)*. SBC, On the WWW, 2021a.

Oliveira, E., Dias, G., Lima, J., and Pirovani, J. C. Using Named Entities for Recognizing Family Relationships. In *8$^{\underline{th}}$ Symposium on Knowledge Discovery, Mining and Learning – KDMILE*. SBC, Rio de Janeiro, RJ, 2021b. PDF:https://doi.org/10.5753/kdmile.2021.17457 Vídeo:⟨https://youtu.be/7YSreRTEz5k⟩.

Oliveira, E., Spalenza, M., and Pirovani, J. *r*AVA: A Robot for Virtual Support of Learning. In *20$^{\underline{th}}$ International Conference on Intelligent Systems Design and Applications – (ISDA)*. Springer, Springer International Publishing, On the WWW, pp. 102–113, 2020.

Parsaeimehr, E., Fartash, M., and Torkestani, J. A. An Enhanced Deep Neural Network-Based Architecture for Joint Extraction of Entity Mentions and Relations. *International Journal of Fuzzy Logic and Intelligent Systems* 20 (1): 69–76, 2020.

Paumier, S. *Unitex 3.2 User Manual*, 2021. Acesso em: 24/06/2021.

Pirovani, J., Alves, J., Spalenza, M., Silva, W., Silveira Colombo, C., and Oliveira, E. Adapting NER (CRF+LG) for Many Textual Genres. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*. CEUR Workshop Proceedings, vol. 2421. CEUR-WS.org, Bilbao, Spain, pp. 421–433, 2019.

Pirovani, J., Nogueira, M., and Oliveira, E. Indexing Names of Persons in a Newspaper Large Dataset. In *13th International Conference on the Computational Processing of Portuguese (PROPOR)*. Vol. 11122. Springer, Canela, RS, 2018.

Pirovani, J. and Oliveira, E. Portuguese Named Entity Recognition Using Conditional Random Fields and Local Grammars. In *LREC*. European Language Resources Association (ELRA), Miyazaki, Japan, pp. 4453–4456, 2018.

Pirovani, J. and Oliveira, E. Studying the Adaptation of Portuguese NER for Different Textual Genres. *The Journal of Supercomputing*, 2021.

Pirovani, J., Spalenza, M., and Oliveira, E. Geração Automática de Questões a Partir do Reconhecimento de Entidades Nomeadas em Textos Didáticos. In *XXVIII Simpósio Brasileiro de Informática na Educação (SBIE)*. SBC, Ceará, CE, pp. 1147–1156, 2017.

Pirovani, J. P. C. *CRF+LG: Uma Abordagem Híbrida para o Reconhecimento de Entidades Nomeadas em Português*. Ph.D. thesis, Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, ES, 2019.

Rocha, C., Jorge, A., Sionara, R., Brito, P., Pimenta, C., and Rezende, S. PAMPO: Using Pattern Matching and Pos-tagging for Effective Named Entities Recognition in Portuguese, 2016.

Santos, C. N. and Guimaraes, V. Boosting Named Entity Recognition with Neural Character Embeddings. In *Proceedings of the Fifth Named Entities Workshop, ACL 2015*. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 25–33, 2015.

Santos, D. and Cardoso, N. *Reconhecimento de Entidades Mencionadas em Português: Documentação e Actas do HAREM, a Primeira Avaliação Conjunta na Área*. Linguateca, 2007.

Yang, J., Zhang, Y., and Dong, F. Neural Reranking for Named Entity Recognition. *arXiv preprint arXiv:1707.05127*, 2017.