MTS 1.0: UMA INOVAÇÃO INTERDISCIPLINAR PARA O ENSINO DE TEORIA E ANÁLISE LINGUÍSTICA*

Marcos Moisés Crisóstomo de Oliveira[1]
Laís Lima eSilva [2]
Brendo Nascimento Santos Souza [3]

Resumo
O MTS 1.0 é um parser sintático da Língua Portuguesa, um modelo de sistema de processamento de linguagem natural (PLN) por regras, desenvolvido na linguagem de programação Prolog com módulos de formalização sintagmática e de entradas lexicais elaborados a partir dos trabalhos de Covington (1994), Chierchia (2003) e Othero (2006). Ele é uma ferramenta didática utilizada nas aulas de Teoria e Análise Linguística do ensino médio e superior do IFBA Campus Santo Amaro. Esse trabalho discute essa aplicação, mostrando como a interdisciplinaridade entre ensino de língua e de lógica de programação pode ser alcançada através da implementação de analisadores sintáticos automáticos, promovendo a aproximação entre os pesquisadores das áreas da linguística e da computação. A análise foca os resultados do aprendizado de regência e de concordância através da tarefa de formalização desses fenômenos para a linguagem Prolog. Assim, buscar-se-á considerar como os sistemas de PLN por regras, atualmente suplantados em valor comercial pelos sistemas estatísticos, ainda têm relevância no ensino e desenvolvimento de ferramentas para a pesquisa em Linguística de Corpus.

Palavras-chave : *Desenvolvimento de programas, Linguística de corpus, Processamento de Linguagem Natural, Parser*

Abstract
The MTS 1.0 is a syntactic parser of Portuguese, one PLN (Natural Language Processing) system model by rules developed in Prolog programming language with syntagmatic formalization modules and elaborate lexical entries from the works of Covington (1994), Chierchia (2003) and Othero (2006). The implementation of this system is used as a teaching tool in the theory classes and Analysis of Linguistic in high school and college (IFBA). This paper discusses this application, showing how this interdisciplinary teaching language and programming logic can be exploited positively in the development of automated parsers, promoting closer relations between researchers in the fields of linguistics and computing. The analysis focuses on the results of the regency of learning and agreement through the formalization of these phenomena task for Prolog. Finally, it will be deemed to technological relevance for research in linguistics these NLP systems by rules, which are currently supplanted in commercial value by statistical systems, still have relevance in education and development tools for research in Corpus Linguistics.

Keywords: *Development programs, Corpus Linguistics , Natural Language Processing, Parsing*

1. Introdução

Esse artigo pretende demonstrar os primeiros passos de um trabalho ainda em construção no Instituto Federal de Educação Ciência e Tecnologia da Bahia - IFBA Campus de

* Gostaríamos de agradecer à Capes (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo financiamento dos anais da VII Escola Brasileira de Linguística Computacional e do XIII Encontro de Linguística de Corpus, processo nº 3472/2015-87.

1 Professor Me Marcos Moisés Crisóstomo de Oliveira, Professor e Pesquisador em Linguística computacional do IFBA – Instituto Federal de Educação, Ciência e Tecnologia da Bahia campus de Santo Amaro, marcosoliveira@ifba.edu.br
2 Láis Lima Silva, Estudante do 3°ano do Curso Integrado de Técnico em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia- IFBA Campus Santo Amaro, laiilima@outlook.com
3 Brendo Nascimento Santos Souza, Estudante do 3°ano do Curso Integrado de Técnico em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Bahia- IFBA Campus Santo Amaro, brendo.urie.02@gmail.com

Santo Amaro que busca aproximar linguistas e programadores, possibilitando a formação de uma equipe de pesquisa capaz de atuar com autonomia tanto no desenvolvimento de aplicações em Processamento de Linguagem Natural (PLN), quanto na realização de estudos em Linguística de Corpus. Para tanto, ele pretende abrir as peculiaridades do mundo da programação aos linguistas, ao mesmo tempo em que buscará promover aos programadores conhecimentos linguísticos necessários ao desenvolvimento de autômatos gramaticais.

Muitas barreiras ainda existem para que a habilidade inata dos seres humanos de uso de línguas naturais seja dominada eletronicamente e posta a serviço do bem comum. No entanto, mesmo considerando essas dificuldades, importantes avanços têm sido conquistados nas pesquisas em Inteligência Artificial (IA). Tal fato vem se confirmando, mais precisamente, na divulgação dos trabalhos de uma subárea dessa ciência, que tem como objeto de pesquisa o processamento de línguas naturais por computadores. Trata-se da Linguística Computacional, uma disciplina que alia os conhecimentos da Linguística e da Ciência Computação a fim de realizar o tratamento das línguas naturais de maneira automática. Nesse trabalho, o foco se dá sobre uma das tarefas principais desse tipo de pesquisa, o desenvolvimento de *parsers*, softwares capazes de fazer a classificação e a anotação morfológica, sintática e/ou semântica das palavras de uma determinada sentença de forma automática e estrutural, tomando como base a Gramática Generativa Transformacional (GGT).

Essa modalidade de pesquisa é considerada uma das mais importantes e basilares quando se trata de integrar ainda mais a inteligência humana e a artificial. Isso porque, um sistema dessa natureza, capaz de fazer anotação morfológica, sintática (morfossintática) e também semântica das palavras, pode ser usado como base para a implementação de alguns aplicativos importantes de PLN, como os tradutores automáticos, corretores ortográficos e sumarizadores de informação, bastante conhecidos e utilizados pelo público em geral na atualidade. De fato, dentro desses programas, o analisador automático atua como o interpretador, ou seja, ele é a ferramenta responsável pela determinação de um modelo de língua que servirá de guia na execução de tarefas automáticas dependentes do conhecimento de padrões linguísticos.

Em geral, quando se fala em implementação de *parsers*, o assunto parece circunscrever-se exclusivamente no território do desenvolvimento de programas, entretanto, vale salientar que é na pesquisa com corpora que esse tipo de analisador se torna ainda mais pródigo. Isso por que, a Linguística de Corpus, objetivando a pesquisa linguística de uma determinada língua, ou de comparação entre duas ou mais línguas, ou ainda, de uma variedade linguística, atua fazendo coleta, compilação e exploração de *corpora* (conjuntos de textos, ou dados linguísticos). E, para que essa tarefa possa ser processada de forma automática, possibilitando ganhos e avanços à pesquisa, é de suma importância que essa amostra de língua esteja eletronicamente estruturada, ou seja, ela deve estar em um formato processável pelo computador.

Para que um corpus da língua seja estudado, primeiramente, o pesquisador em Linguística de Corpus precisa fazer uma delimitação teórica de cerne linguístico que será a base do seu enfoque de estudo. Somente assim, ele poderá definir qual é a natureza de corpus mais adequada para seu objetivo de pesquisa. Isso porque, independentemente de sua dimensão, esse conjunto de amostras textuais, ou de dados linguísticos deve ser representativo de uma língua ou de uma variedade dela. Em outras palavras, pode-se dizer que o pesquisador em Linguística de Corpus começa seu trabalho pelo estabelecimento de um modelo de língua que deverá guiar a pesquisa, podendo ser ao final do trabalho, confirmado, ou refutado.

Cabe aqui lembrar, antes de seguir a análise da rotina de trabalho atual da Linguística de Corpus no tocante ao uso de analisadores sintáticos automáticos, que essa área da linguística, baseada na análise de corpora, não tem sua origem marcada pelo surgimento dos computadores. Sua prática data de tempos bem anteriores à era cristã. Isso porque, os relatos do uso de corpora para a produção de conhecimento são quase tão antigos quanto a própria escrita. Na Grécia

Antiga, Alexandre, o Grande definiu o *Corpus Helenístico*; o Império Romano legou ao mundo o *Corpus Juris Civiles*, base do direito contemporâneo das sociedades ocidentais; na Antiguidade e na Idade Média, era muito comum a produção de *corpora* de citações da Bíblia. Também, durante boa parte do século XX, houve muitos pesquisadores que se dedicaram à descrição da linguagem por meio da manipulação manual de corpora, entre eles educadores como Thorndike e linguistas de campo como Boas ganham destaque devido a relevância dos seus trabalhos.

Há duas diferenças fundamentais entre os primórdios dessa disciplina e o seu estado de arte atual. A primeira, obviamente, é que os corpora não eram eletrônicos, ou seja, eram coletados, mantidos e analisados manualmente de forma exclusiva pelos linguistas. A segunda, é que a ênfase destes trabalhos era em geral o ensino de línguas. Atualmente, observa-se que prepondera na literatura uma tendência a fazer a descrição de linguagem sem preocupação de ordem pedagógica, visando apenas a aplicação na construção de sistemas de PLN, embora recentemente seja notável o ressurgimento do interesse pelo emprego de corpora na sala de aula e na investigação da linguagem de alunos de língua (Granger, 1998).

Nesse ponto, é importante alertar que esse artigo busca apresentar uma proposta de trabalho que abarca todas essas aplicabilidades supracitadas. Isso porque, esse projeto, que atualmente vem sendo aplicado nos cursos de informática do IFBA Campus de santo Amaro, em todas as suas modalidades, tem interesse no ensino e na descrição da língua, na medida que pretende fazer-se presente na formação desses estudantes, objetivando, com isso, tanto o desenvolvimento das habilidades básicas de uso da língua por cidadãos, quanto a preparação desses discentes para participarem como programadores em pesquisas de PLN, ou de Linguística de Corpus. Nesse sentido último, encontra-se uma preocupação específica em aliar a proficiência da língua e o seu conhecimento teórico, e também aplicado, à indispensável boa utilização das máquinas para o trabalho de pesquisa linguística.

A partir da década de 60, os computadores do tipo *mainframe* começam a equipar centros de pesquisa universitários, e, naturalmente, passaram a ser aproveitados como ferramentas para a pesquisa em linguagem. Posteriormente, a popularização dos computadores veio a incentivar o acesso de mais pesquisadores ao processamento de linguagem natural e, concomitantemente, trouxe consigo a aceleração da sofisticação do equipamento que permitia a realização de tarefas mais complexas, de modo mais eficiente. Isso sem falar no aumento da capacidade de armazenamento e na introdução de novas mídias, as quais facilitaram a criação e a manutenção de *corpora* em maior número.

Com a entrada em cena dos microcomputadores pessoais, nos anos 80, uma nova onda de mudanças aconteceu, como a popularização de *corpora* e de ferramentas de processamento, o que contribuiu decisivamente para o reaparecimento e fortalecimento da pesquisa linguística baseada em *corpus*. Assim, essa área vem experimentando amplo crescimento desde que se tornou necessária a buscar por alternativas tecnológicas capazes de apoiar o estudo da língua em amostras autênticas. Isso porque, a configuração atual dos corpora justamente permite que se pesquise e se baseie em linguagem natural e autêntica, possibilitando, assim, a comprovação de teorias por meio de análise dos dados efetivamente produzidos por usuários da língua. Segundo Berber Sardinha (2000, p. 2),

> A Linguística de Corpus se ocupa da coleta e exploração de corpora, ou conjuntos de dados linguísticos textuais que foram coletados criteriosamente com o propósito de servirem para a pesquisa de uma língua ou variedade linguística. Como tal, dedica-se à exploração da linguagem através de evidências empíricas, extraídas por meio de computador.

Partindo-se desse viés, é necessário que haja a compreensão de que um *corpus* é um conjunto de textos, ou de dados linguísticos autênticos, coletados e selecionados com uma intenção de pesquisa, de maneira que se parte do enfoque da pesquisa para o corpus e não o

contrário. Assim, deve ficar estabelecido que um *corpus* precisa ser a representação de uma determinada variedade linguística, a qual será utilizada como base para análise e pesquisa sobre padrões da língua investigada. Sanchez e Cantos (1996 apud Sardinha 2004, p. 18) se referem a essa necessidade dizendo que um corpus é:

> Um conjunto de dados linguísticos (pertencentes ao uso oral ou escrito da língua, ou a ambos), sistematizados segundo determinados critérios, suficientemente extensos em amplitude e profundidade, de maneira que sejam representativos da totalidade do uso linguístico ou de algum de seus âmbitos, dispostos de tal modo que possam ser processados por computador, com a finalidade de propiciar resultados vários e úteis para a descrição e análise.

De fato, esse novo papel da máquina no processo de realização da pesquisa com *corpora*, que caracteriza o atual estado de arte da Linguística de Corpus, torna necessário que o conhecimento linguístico seja algo comum entre linguistas e programadores, ou que, pelo menos, estes sejam capazes de comungar com aqueles de um mesmo modelo de língua. Caso contrário, muitas serão as dificuldades de entendimento entre ambos, dificultando, assim, o processo de desenvolvimento de ferramentas que hoje são a base desse trabalho de pesquisa.

Nesse sentido, é importante aqui salientar que a manipulação de ferramentas computacionais faz parte da realidade de pesquisa atual dessa disciplina. Desse modo, entende-se que, atualmente, as tarefas de concepção, e desenvolvimento das mesmas são partes integrantes dessa rotina de trabalho. Então, feita a definição da natureza linguística do corpus, o linguista procederá a tarefa de fazer a coleta e, posteriormente, realizará junto ao programador o trabalho de compilação, aqui entendido como o processo de tradução do material coletado, originalmente em língua natural, para a linguagem de máquina, propiciando, desta maneira, o tratamento automático do material colhido.

Observe-se que este já não é mais um trabalho exclusivo do linguista. Agora, ele ganha a companhia de outro profissional especialista em linguagem artificial, ou linguagem de máquina. Dessa forma, a equipe, em seguida, fará a exploração do *corpus* e é, nessa fase, que a importância de analisadores automáticos, bem como a do estabelecimento de um modelo de língua, crescem para a pesquisa em curso. Isso acontece porque, muitas das informações que o linguista quer buscar no *corpus* não se encontram evidentes. Isso porque, ele, por si só, não é capaz de reconhecer o que seja "verbo" ou "adjetivo". Essas informações precisam ser fornecidas e manipuladas, para que perguntas normalmente formuladas por linguistas venham a ser respondidas, como, por exemplo, quais os adjetivos usados no *corpus*? Qual a frequência que eles aparecem como determinantes do nome, ou como predicativos?

Considerando-se ainda uma pesquisa que focalize a natureza dos verbos, é possível formular questões capazes de revelar peculiaridades no uso da língua que podem ser extraídas de elementos como a frequência de uso da forma composta, no lugar da forma simples para alguns tempos verbais como pretérito mais que perfeito. Essa característica[4], por exemplo, pode permitir uma distinção entre uma variedade linguística do Português do Brasil e do Português de Portugal, tornando-se substância de uma oportuna pesquisa sociolinguística apoiada na metodologia automática de *corpora*.

O fornecimento dessas informações só é possível devido a um processo conhecido como anotação. Dessa forma, além da necessidade de escolher bem o *corpus* para sua pesquisa, os pesquisadores precisam escolher uma ferramenta que lhes possa ser útil para realizar essa análise, dependendo de seu objetivo. Vale aqui, portanto, destacar novamente a importância do computador para tarefas que poderiam ser muito exaustivas, de modo a levar um longo tempo, além de dispensar altos custos caso fossem feitas com as limitações humanas, tais como contar palavras, ou fazer o levantamento da frequência de ocorrência de fenômenos.

---

4 Em geral, no Brasil pouco se usa a forma simples para o pretérito mais que perfeito. Observa-se que é mais comum o usuário dessa variedade usar " eu tinha estado" no lugar de " eu estivera. Esse fato, não acontece no Português de Portugal. Nele, é mais comum a forma simples.

A anotação, ou etiquetagem de *corpus*, é uma operação de extrema importância para as pesquisas com *corpora*, pois é ela quem permitirá ao *corpus* o cumprimento do seu papel como instrumento para investigação. Em outras palavras, pode-se afirmar que a etiquetagem morfossintática permite que uma ferramenta computacional etiquete uma grande quantidade de texto, ou dados linguísticos de maneira eficiente. Segundo Alencar (2012, p. 125), a etiquetagem "é uma tarefa aparentemente simples para o processamento da linguagem, no entanto, o desempenho de outras ferramentas depende diretamente desse processo". Além disso, o mesmo autor assegura que a importância do anotador automático também diz respeito à questão de se entender a linguagem natural. Para ele, "a etiquetagem morfossintática é uma tarefa intermediária que tem como objetivo principal analisar e entender a língua natural" (ALENCAR, 2012, p. 125).

Neste artigo, apresenta-se o processo de implementação de uma ferramenta de etiquetagem que tem se mostrado bastante útil para quem precisa recolher informações morfossintáticas em um corpus de dados linguísticos. Tal tarefa será realizada mediante ao processo etiquetagem que pode ser, basicamente, morfológica ou morfossintática. Nela, são classificadas as unidades lexicais dos dados linguísticos através de *tags* (etiquetas) morfossintáticas, em que se identificam as estruturas da sintaxe presentes. Nesse sentido, pode-se falar de um programa etiquetador morfossintático, que faz especificamente uma etiquetagem automática, marcando antes de cada palavra do corpus qual é sua classe gramatical e sintática. Esse tipo de marcação é também conhecida como Notação entre Colchetes (*bracketed notation*) baseando-se num conjunto de etiquetas morfossintáticas preestabelecido, que por ser oriundo da GGT, é chamado tradicionalmente de *parser*[5].

Os *parsers* são ferramentas usadas pela linguística de corpus para fazer a etiquetagem de textos a fim de possibilitar o estudo dos seus padrões sintáticos. A utilização desses programas como ferramenta nos estudos de corpora tem sua origem nos estudos da GGT que começaram com os trabalho de Noam chomsk na década de 50. Entretanto, tornaram-se mais visíveis na década de 80 com a popularização dos computadores domésticos. Antes disso, o trabalho com corpora era realizado de forma manual e não gozava de credibilidade nos meios científicos devido às suas limitações técnicas.

Esse artigo objetiva mostrar como essa atividade vem servindo de ferramenta pedagógica no ensino de línguas, mais precisamente nas aulas de Teoria e Análise Linguística (Língua Portuguesa/Análise Sintática) no IFBA Campus de Santo Amaro. Ele vem sendo usado como metodologia para preparação de pessoal capaz de compor equipes de profissionais de informática habilitados a trabalharem em conjunto com linguistas nas pesquisas em PLN e Linguística de Corpus visando avanços na área de Inteligência Artificial e da Linguística Aplicada.

Ainda a cerca da questão do modelo de língua, esse trabalho pretende demonstrar como a tarefa de implementação de um parser pode atuar como inovação no ensino de língua, principalmente, quando se trata de uma prática executada no âmbito de cursos da área de informática. Tal avanço vislumbra a preparação de pessoal dessa área para atuar em conjunto com linguistas em pesquisas da ordem da Linguística de Corpus ou da Linguística Computacional. Isso acontece porque, essa é uma necessidade que tem sua origem nas dificuldades que se apresentam quando do trabalho de uma equipe composta por linguistas e programadores. É sabido que um dos pontos críticos dessa conjunção está exatamente no desconhecimento da língua por parte destes. Assim sendo, é muito difícil se obter bons resultados quando o modelo de língua que será a base do trabalho não é um conhecimento comum a todos os entes da equipe.

De outro lado, também se deve reconhecer, embora esse não seja o foco desse artigo, já que não se trata de uma experiência pedagógica também verificada nos cursos de Letras, que o exercício de manipulação de uma linguagem de programação como o *Prolog* permite ao

5 caso contrário, a literatura tem nomeado de etiquetadores, ou *Taggers.*

linguista se aproximar de um conhecimento técnico que lhe é necessário, na medida que propicia a este pesquisador, em especial, conhecimentos comuns para um diálogo eficiente com os programadores. Além do mais tal prática ainda é também capaz de legar ao mesmo uma confortável autonomia na escolha da concepção de ferramentas que ele necessitará em suas investigações.

2. Desenvolvimento de um *parser* sintático

O Termo *parser*, empregado nesse trabalho, faz referência a um tipo de aplicação computacional capaz de analisar a estrutura de constituintes de sentenças em línguas naturais de acordo com uma determinada gramática (como dito anteriormente, baseada na GGT). Essa é uma ferramenta básica para o desenvolvimento de muitos sistemas para processamento de linguagem natural, pois o mínimo que se espera desse tipo de máquina é a capacidade de interpretar as estruturas de regras sintagmáticas que formam as sentenças de uma determinada língua.

A expressão *parsing*, que nesse artigo é definida como a atividade executada pelos *parsers*, vem do termo latino *pars orationes* (parte do discurso) e não remete propriamente aos domínios do processamento computacional. Na verdade, esse termo tem sua raiz na tradição clássica greco-romana e significava, já naquela época, o processo de atribuição de uma estrutura formal e de uma interpretação lógica a uma sentença linguística. Aliás, nas práticas de ensino atuais e tradicionais, essa atividade, que é peculiar às aulas da disciplina Língua Portuguesa, encontrada em qualquer nível, ou modalidade de ensino, é denominada de Análise Sintática, ou morfossintática, na medida em que também trata de questões ligadas à forma das palavras.

Um dos objetivos principais dessas aulas de Teoria e Análise Linguística é trabalhar a lógica através da linguagem de modo a propiciar como legado o bom conhecimentos da língua promovendo o domínio de fenômenos como pontuação, concordância e regência. Em outras palavras, é possível dizer que a tarefa de dispor frases de um idioma em sua estrutura de constituintes significa fazer uma formalização lógica para o fenômeno da língua.

Foi por conta dessa questão de natureza lógica que esse projeto escolheu transformar as atividades de *parsing*, tradicionalmente executadas nas aulas de análise sintática, em aulas de desenvolvimento de programas analisadores, beneficiando-se dessa intersecção de conhecimentos que põe em rota de interdisciplinaridade os estudos linguísticos, a lógica formal e o desenvolvimento de programas. Para tanto, escolheu-se uma linguagem de programação baseada na lógica, a fim de se elaborar uma gramática da Língua Portuguesa.

2.1. Prolog

O Prolog é uma linguagem de programação muito utilizada nos trabalhos em IA e em Linguística computacional. Consta que ele foi desenvolvido por Alan Colmerauer, na França, em 1975. Esse nome vem de *PROgramming in LOGic* (programando em lógica) , porque, justamente, essa linguagem é baseada na lógica. Assim, desde o princípio, o seu desenvolvimento visava o processamento de linguagem natural na tarefa de implementação de *parsers* automáticos. Isso porque, diferente das linguagens de programação procedimentais, em que o programador desenvolve algoritmos de comando que dão ordens, na forma de um passo a passo, para que a máquina chegue a algum resultado, o Prolog é uma linguagem declarativa, ou seja, nele, o programador irá declarar proposições que envolvem objetos e as regras de relacionamento lógico entre eles.

Essas proposições são dispostas em forma de cláusulas com um formato conhecido como *lógica de predicados*. Ou seja, por ser uma linguagem declarativa, a máquina decidirá a resposta de um problema através de operações lógicas relativas a esses dados que tem disponível em seu banco de dados.

O Prolog ainda tem outra vantagem quando a questão é a elaboração de analisadores

sintáticos automáticos. A maioria das versões recentes dessa linguagem vem equipada com uma DCG, ou gramática de cláusulas definidas (do inglês *Definitive clause grammar*). Trata-se de uma extensão nocional que facilita a implementação de regras formais de *parsing*. Em outras palavras, ela seria um formalismo de representação de gramáticas livres de contexto que torna fácil a implementação de uma gramática para o desenvolvimento de um analisador sintático automático.

2.2.Implementação

Uma gramática é definida formalmente por uma 4-tupla (N,W,S,R) onde:

N → Um conjunto de símbolos não-terminais, ou seja, são as palavras da língua dispostas no módulo de dicionário. No exemplo abaixo, “mário”, “carro”, “o”, “lavou”.

W → Um conjunto de símbolos terminais. Classes morfológicas, ou constituintes sintáticos (sintagmas) que aparecem na forma de rótulos. No exemplo, n (nome), det (determinante), v (verbo), sn (sintagma nominal) e sv (sintagma verbal)

S → Pertence a N. É o símbolo inicial. Significa uma sentença, que é a categoria de análise do programa.

R → Um conjunto de regras de reprodução que ficam no módulo de regras sintagmáticas.

No processo de implementação, primeiro o programador vai abrir um bloco de notas e, nele, irá declarar a gramática como mostra a Figura 1. Nela, tudo que vem depois do símbolo “%” são comentários colocados aí para facilitar o entendimento. Isso porque, esse símbolo serve para fazer declarações que não serão processadas na linguagem Prolog.

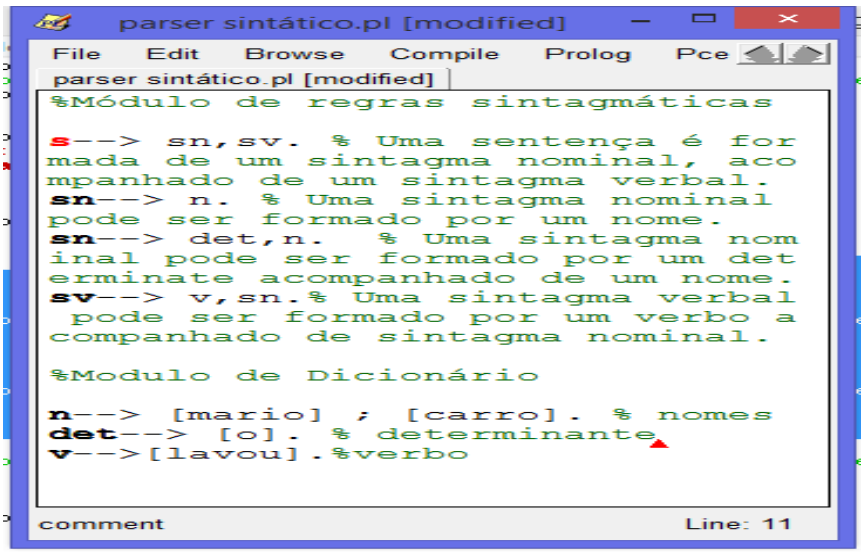

Figura 1 – Programação no bloco de notas

Depois de feita a programação dos módulos de entrada lexical (dicionário) e de regras sintagmáticas no bloco de notas, como se vê na figura acima, faz-se a compilação para linguagem Prolog. Então, abre-se o programa para fazer a consulta. Como é demonstrado na Figura 2.

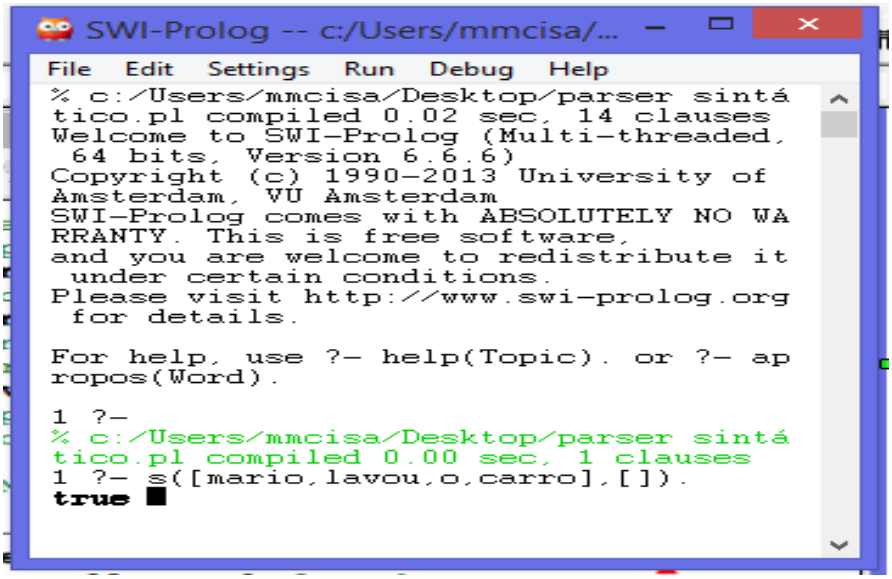

Figura. 2 – Consulta no Prompt de Comando

No caso dessa consulta demonstrada na Figura 2, tem-se uma programação mais simples que trabalha somente com respostas *booleanas (true/false)*. Essa versão de implementação é usada como introdução no treinamento dos discentes. É uma etapa de reconhecimento das regras do sistema e do funcionamento da DCG em que são apresentados e explicados o sistema de *tags* que classificam cada palavra do léxico da gramática. Depois dessa fase, eles passam à implementação do modelo MTS 1.0.

3. MTS 1.0

O MTS 1.0 é um parser sintático da Língua Portuguesa, um modelo de sistema de processamento de linguagem natural (PLN) por regras, desenvolvido na linguagem de programação Prolog SWI versão 6.6.6 com módulos de formalização sintagmática e de entradas lexicais elaborados a partir dos trabalhos de Covington (1994), Chierchia (2003) e Othero (2006). Ele pode ser concebido tanto como um programa, ou seja, um produto sempre em desenvolvimento, quanto como um modelo de programa, na medida que serve para ser usado como um simulacro da tarefa de implementação de parsers nas aulas das disciplinas Língua Portuguesa.

3.1.Módulo de entradas lexicais

O léxico utilizado na construção da gramática do MTS 1.0 vem sendo construído pelos discentes das turmas de 3º anos dos cursos integrados e da licenciatura em Computação do IFBA campus de Santo Amaro. Eles passam por um treinamento na linguagem Prolog no início do curso de Língua Portuguesa e, durante o período restante do ano letivo, eles efetuam a catalogação das palavras. Essa tarefa constitui parte dos exercícios de aprendizagem da disciplina de modo que acompanha o cumprimento do programa de Língua Portuguesa. Assim, cada turma é dividida em duplas que se responsabilizam pelo cadastramento das classes gramaticais a partir de uma distribuição de palavras organizada por ordem alfabética. Ao todo, até o presente, trabalharam no projeto 50 duplas que compuseram um módulo de entradas lexicais com um corpus de cerca de 67 mil palavras simples da Língua Portuguesa, listadas em sua forma canônica e separadas em diferentes categorias (substantivos próprios, substantivos comuns, determinantes, advérbios, adjetivos e verbos).

Como o objetivo no desenvolvimento do MTS 1.0 é o treinamento constante dos discentes para que esses se tornem desenvolvedores dessa modalidade de sistemas, a cada semestre, esse trabalho é retomado pelos novos alunos que irão aumentar o dicionário com novas palavras (atualmente, a meta é atingir um milhão de palavras até o fim de 2016). Dessa

forma, eles são levados a se familiarizar com as *tags* e com os operadores, ganhando com isso uma perspectiva lógica acerca do fenômeno linguístico. A Tabela 1 demonstra como ficam as entradas lexicais etiquetadas conforme a sintaxe exigida pelo Prolog.

Tabela 1 – Entradas lexicais

```
%Substantivos
n([fem, sing], amiga).
n([fem, plur], amigas).
n([masc, sing], amigo).
n([masc, plur], amigos).
%Adjetivos
adj([fem, sing], alta).
adj([fem, plur], altas).
adj([masc, sing], alto).
adj([masc, plur], altos).
%Determinantes
det([masc, sing], o).
det([fem, sing], a).
det([masc, plur], os).
det([fem, plur], as).
```

3.2. Módulo de regras sintagmáticas

Para proceder à implementação do módulo de regras sintagmáticas em Prolog na gramática do MTS 1.0, esse projeto, na fase atual, tomou como base a descrição sintagmática da sentença simples em PB descritas por Othero (2006). Além do mais, para implementação, fez-se uso do recurso das DCGs, um formalismo de representação de gramáticas livres de contexto. As Tabelas 2 e 3, apresentam como foram implementadas algumas regras linguísticas na linguagem Prolog para o desenvolvimento do MTS 1.0.

Tabela 2 – Regras sintagmáticas versão linguística

- (1) s--> sn,sv. (Uma sentença é formada de um sintagma nominal, acompanhado de um sintagma verbal)
- (2) sn--> n. (Uma sintagma nominal pode ser formado por um nome)
- (3) sn--> det,n. (Uma sintagma nominal pode ser formado por um determinante acompanhado de um nome)
- (4) sv--> v,sn. (Um sintagma verbal pode ser formado por um verbo acompanhado de sintagma nominal)

Tabela 2 – Implementação de regras sintagmáticas em Prolog

```
(1') s([sn, SN, sv, SV]) --> sn([_,Num], SN), sv(Num, SV).
(2') sn(Conc, [n_bar, N_Barra]) --> n_barra(Conc, N_Barra).
(3') sn(Conc, [det, [Det], n_bar, N_Barra]) --> det(Conc, Det), n_barra(Conc, N_Barra).
(4') sv(Num, [v_bar, V_Barra, sadv, SAdv ]) --> v_barra(Num, i, V_Barra), sadv(SAdv).
```

3.3. Funcionamento do programa

O resultado final é uma ferramenta didática utilizada nas aulas de teoria e análise linguística do ensino médio e superior do IFBA Campus Santo Amaro. Nele, ao digitar-se uma frase, se ela pertencer à Língua Portuguesa, o programa mostrará a notação entre colchetes (*bracketed notation*) que representa a regra sintagmática que rege aquela sentença, ou melhor,

ele apresentará a frase interpretada na sua estrutura sintática, desde que estejam contidos no banco de dados o léxico da sentença, devidamente etiquetado, e a regra sintagmática que rege a sua formação. Caso contrário, se a frase não for da língua portuguesa, o programa responderá que não pertence a esta, ou devido à falta de léxico da frase no banco de dados, ou porque falta nele a regra sintagmática que corresponda àquela frase, ou ainda por ser a frase, de fato, agramatical.

A finalidade da construção desse sistema é propor essa tarefa de programação como ferramenta de aprendizagem da língua em suas estruturas linguísticas, por isso nesse trabalho o objetivo é demonstrar em cada aula como, na montagem do programa, são elaborados os códigos capazes de processar fenômenos linguísticos tais como a regência nominal e verbal, a concordância nominal e verbal.

3.4. Formalização do fenômeno de concordância nominal e verbal

Entende-se por concordância nominal e verbal a relação obrigatória de simetria nas flexões de gênero, e/ou número entre um núcleo sintagmático e seus complementos, ou determinantes. Em termos práticos se o núcleo estiver no singular, seus determinantes e complementos deverão estar também no singular, o mesmo acontecendo no caso do plural, ou ainda nos casos de determinação de masculino ou feminino.

Por ser uma linguagem baseada na lógica, o funcionamento do analisador sintático, em Prolog, em termos de concordância na língua portuguesa, sofre com a flexão de número (singular e plural) e de gênero (feminino e masculino). Assim, para que a frase seja aceita pelo MTS 1.0 como uma frase gramatical e tenha seu léxico descrito através do que já foi acumulado no banco de dados, uma das exigências é que a sentença apresente corretamente os valores de número e gênero. Se isso acontecer, o programa responderá positivamente de forma booleana (True), dando também a estrutura sintática da sentença. Caso contrário, a frase será classificada como agramatical, ou de léxico desconhecido. Como é demonstrada na Figura 3.

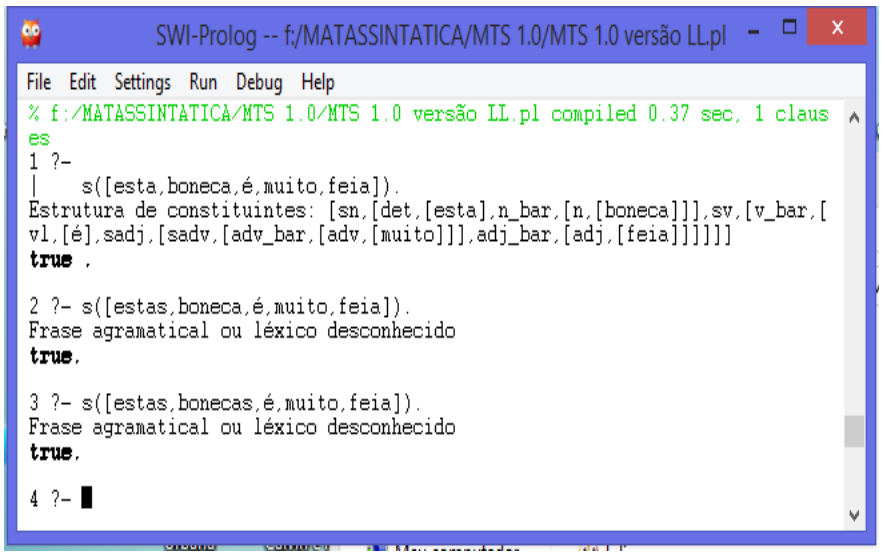

Figura 3 – teste de senteça para concordância verbal e nominal

Para que essa operação seja possível é necessário que os símbolos não terminais ganhem *tags* identificadores dos valores para gênero e número e que haja, nas regras sintagmáticas, operadores capazes de exigir essas simetrias. Nas duas primeiras cláusulas a seguir, que declaram a estrutura sintagmática de uma sentença em ordem direta (SVO) e a de uma sentença em ordem inversa (VSO), respectivamente, pode-se observar a presença do operador "Num" que é quem garante que, se o sintagma nominal (sn) que ocupa a posição de sujeito, estiver no plural, obrigatoriamente o sintagma verbal (sv) deverá concordar com ele estando também no plural. Tem-se aí a solução dada para o processamento da concordância

verbal como pode ser visto na Tabela 4.

Tabela 4 – Operador "NUM" - Concordância Verbal

• s([sn, SN, sv, SV]) --> sn([_,Num], SN), sv(Num, SV).
• s([sv, SV, sn, SN]) --> sv(Num, SV), sn([_,Num], SN).

Nas cláusulas da Tabela 5 a seguir, que declaram a forma de um sintagma nominal, pode-se perceber como a concordância nominal é garantida pelo operador "conc" que exige a simetria entre o núcleo e os seus determinantes.

Tabela 5 – Operador "CONC" - Concordância nominal

• sn(Conc, [pre_det, [Pre_Det], sn, SN]) --> pre_det(Conc, Pre_Det), sn(Conc, SN).

3.5 Formalização do fenômeno de regência nominal e verbal

O fenômeno de regência verbal e nominal ocorre quando há uma relação de submissão entre um verbo, ou um nome, que ocupam a função de núcleo de um sintagma e seus complementos, de modo que estes sejam expressos com clareza, evitando ambiguidades e garantindo-lhes a aplicação correta. De maneira prática, refere-se à natureza dessa relação de ligação mediante o uso, ou não, de preposição entre um núcleo e os seus complementos. Nesse caso, distingue-se esse núcleo como "termo regido" e a preposição como "termo regente". Diz-se então que o verbo "precisar" é regido pela preposição "de", ou que o verbo "comprar" não é regido por preposição. Isso define o primeiro verbo como transitivo indireto, ou seja, aqueles verbos que fazem ligação com os complementos obrigatoriamente sendo mediados, ou regidos, por uma preposição, no caso aqui "de", e o segundo, como transitivo direto, aqueles que não precisam de termo regente para mediar a ligação com os sintagmas que ocupam a posição de objeto.

O MTS 1.0, que tem sua formalização baseado no *Parser Grammar Play*, implementado em Othero (2006) e baseado na teoria X barra elaborada pelo linguista Noam Chomsk, traz no módulo de entradas lexicais a seguinte formalização para verbos, dando classificação sintática e identificando o termo regente quando necessário:

Tabela 6 – Verbos declarados com termo regente

• v(sing, ti(de), precisa). % verbo transitivo indireto regido pela preposição até.
• v(plur, td, compram).% verbo transitivo direto.
• v(sing, tdi (a), oferece).% verbo transitivo direto e indireto regido pela preposição.

Observando a TABELA 6, é possível perceber como o fenômeno de regência verbal é formalizado nesse programa. No caso da figura, vê-se o verbo sendo tratado conforme a sua classificação sintática (ti - transitivo indireto) levando em conta a definição da preposição adequada, ou seja, a preposição que rege o mesmo verbo.

4. Considerações finais

A realização dessa tarefa de implementação do *parser* MTS 1.0 nas aulas de Teoria e Análise Linguística apresentou um grande avanço no aprendizado de análise sintática no tocante ao tratamento dos fenômenos de regência e concordância. Observou-se que a escolha por essa nova metodologia veio a sanar um problema tradicional nesse ensino. Acontece que, geralmente, o professor de Língua Portuguesa ensina a morfossintaxe da língua explicando a sua estrutura de constituintes num momento diferente da exposição sobre os fenômenos de regência e concordância. Em geral, os dois assuntos são tratados de forma independente e trabalhados em momentos distintos do ano letivo. Dessa forma, resulta que o aprendizado

desses últimos conteúdos aparenta ser uma atividade de mera memorização de regras e de casos de exceções a essas regras que, consequentemente, torna a atividade enfadonha e sem sentido, já que o discente não é levado a perceber que esses fenômenos são, basicamente, lógicos, ainda que haja muitas exceções advindas da cultura.

Esse problema de progressão de conteúdos não acontece na atividade de implementação, pois nela, o estudante é levado a observar os impactos lógicos que esses fenômenos de regência e concordância têm na hora de declarar as regras de relação dos constituintes. Assim, a concomitância de conteúdos, que é peculiar à tarefa de desenvolvimento do analisador, figura como uma solução capaz de fazer o estudante perceber o fenômeno linguístico como uma ocorrência plausível e genuinamente lógica.

Outro avanço para o ensino diz respeito ao uso da tarefa de implementação de *parsers* como atividade avaliativa. Nessa modalidade, como o trabalho de desenvolvimento consiste em estar sempre buscando corrigir o programa a fim de que ele funcione sem falhas, o docente pode focalizar a avaliação nos acertos ("consertos") feitos pelo estudante. Na atividade tradicional, meramente quantitativa, com exames escritos, o avaliador dá a nota a partir da quantidade de erros e acertos. Nessa atividade, como o objetivo é entregar um produto pronto, o discente é levado a corrigir seus erros, até que o programa não acuse mais nenhum deles. Assim, ele é obrigado a evoluir sempre até que se complete o aprendizado com a implementação do parser concluída.

Essa tarefa interdisciplinar de implementação de um *parser* sintático, além da vantagem de representar uma prática pedagógica, que une as disciplinas de Linguística, Matemática e Programação, se constitui numa excelente inovação para ensino de Língua Portuguesa, mais precisamente na tarefa de análise linguística e, no futuro, poderá amparar a pesquisa no conhecimento de outras línguas, inclusive aquelas que, por ventura, já tenham desaparecido, ou que estão em curso de desaparecimento, como, por exemplo, algumas línguas de povos indígenas do Brasil. Ademais, essa é uma metodologia que cumpre o objetivo de familiarizar os estudantes dos cursos de informática da modalidade integrada (ensino médio) e os do ensino superior em Licenciatura da Computação com a nomenclatura e a teoria linguística, possibilitando, com isso, a superação de uma barreira que tradicionalmente tem servido de estorvo à comunicação e, consequentemente, ao trabalho realizado entre linguístas e programadoras tanto no desenvolvimento de aplicações em PLN, quanto no uso de ferramentas computacionais para as tarefas de pesquisa com *corpora.*

Por fim, vale destacar que para linguistas, principalmente aqueles que atuam dentro do âmbito da Linguista de corpus, conhecer o processo de desenvolvimento de um parser é a oportunidade de ganhar autonomia na manipulação dessas ferramentas oriundas da tecnologia computacional. Dessa forma, esse trabalho trouxe também como proposta legar a esses pesquisadores uma profundidade de conhecimento técnico capaz de dar-lhes mais do que somente o status de usuários dessa ferramenta, na medida que lhes possibilita o poder de serem também participantes desenvolvedores e responsáveis pela sua concepção. Além disso, espera-se que tal prática, aqui descrita, sirva de convite a outros professores e que, num futuro próximo, possa haver, de fato, um real intercâmbio de conhecimento entre as áreas de Linguística e de Computação também nos cursos de Letras à semelhança do que se prática hoje no curso de Licenciatura da Computação do IFBA - Campus de Santo Amaro na disciplina Comunicação e Informação.

AGRADECIMENTOS

Agradeço ao Grupo de Informática Aplicada (GIA) do IFBA Campus Santo Amaro pelo espaço concedido para realização do trabalho e ao Instituto Federal de Educação, Ciência e Tecnologia da Bahia Campus Santo Amaro pelo Plano de Amparo e Assistência Estudantil (PAAE/PINNA) que concedeu de bolsas de iniciação científica para os coautores desse trabalho, Laís Lima e Brendo Santos, meus alunos e orientandos. A eles também meu "muito

obrigado!”

Referências

ALENCAR, Leonel Figueiredo de. *Superando o estado da arte na etiquetagem morfossintática por meio de regras de pós-etiquetagem*. In: Anais do X Encontro de Linguística de Corpus – Aspectos metodológicos dos estudos de corpora. Belo Horizonte: UFMG, 2012.

ALENCAR, Leonel Figueiredo de; OTHERO, Gabriel de Ávila (Orgs.). *Abordagens computacionais da teoria da gramática*. Campinas: Mercado de Letras, 2011.

BERBER SARDINHA, Tony. *O que é um corpus representativo?* DIRECT Papers 44. São Paulo / Liverpool: LAEL & AELSU, 2000.

_____. *Linguística de Corpus*. Barueri, SP: Manole, 2004.

COVINGTON, M. A. *Natural language processing for Prolog programmers.* New Jersey, Prentice Hall. 1994.

CHIERCHIA, G. *Semântica*. São Paulo: Eduel, 2003.

GRANGER, S. (Org.) (1998) *Learner English on Computer*. New York: Longman.

OTHERO, G. A. *Grammar Play: um parser sintático em Prolog para a língua Portuguesa*. Dissertação de Mestrado *PUCRS*, Porto Alegre, 2006.

THORNDIKE, E. L. (1921) *Teacher's Wordbook*. New York: Columbia Teachers College.