

Avaliação de recursos computacionais para o português

Evaluating computational resources for Portuguese

Matilde Gonçalves 
Instituto Superior Técnico
Universidade de Lisboa
INESC-ID Lisboa

matilde.do.carmo.lages.goncalves@tecnico.ulisboa.pt

Jorge Baptista 
Universidade do Algarve
INESC-ID Lisboa
jbaptis@ualg.pt

Luísa Coheur 
Instituto Superior Técnico
Universidade de Lisboa
INESC-ID Lisboa

luisa.coheur@tecnico.ulisboa.pt

Ana Mineiro 
Instituto de Ciências da Saúde
Universidade Católica Portuguesa
Centro de Investigação Interdisciplinar em Saúde
amineiro@ics.lisboa.ucp.pt

Resumo

Têm sido desenvolvidas várias ferramentas para o processamento da língua portuguesa. No entanto, devido a escolhas variadas na base dos comportamentos destas ferramentas (diferentes opções de pré-processamento, diferentes conjuntos de etiquetas morfosintáticas e de dependências, etc.), torna-se difícil ter uma ideia do desempenho comparativo de cada uma. Neste trabalho, avaliamos um conjunto de ferramentas gratuitas e publicamente disponíveis, que realizam as tarefas de Etiquetagem Morfosintática e de Reconhecimento de Entidades Mencionadas, para a língua portuguesa. São tidos em conta doze modelos diferentes para a primeira tarefa e oito para a segunda. Todos os recursos usados nesta avaliação (tabelas de mapeamento de etiquetas, *corpora* de referência, etc.) são disponibilizados, permitindo replicar/afinar os resultados. Apresentamos ainda um estudo qualitativo de dois analisadores de dependências. Não temos conhecimento de nenhum trabalho similar recente, isto é, que tenha em conta as ferramentas atuais disponíveis, realizado para a língua portuguesa.

Palavras chave

processamento da linguagem natural, avaliação de recursos, língua portuguesa, análise morfosintática, reconhecimento de entidades mencionadas, análise de dependências

Abstract

There are several tools for the Portuguese language. However, and due to different choices at the basis of these tools' behaviour (different pre-processing, different labels, etc.), it becomes difficult to have an idea of each one's comparative perfor-

mance. In this work, we propose an evaluation of tools, publicly available and free, that perform the tasks of Part-of-Speech Tagging and Named Entity Recognition, for the Portuguese language. We evaluate twelve different models for the first task and eight for the second. All the resources used in this evaluation (mapping tables between labels, testing *corpora*, etc.) will be made available, allowing to replicate/fine-tune the results here presented. We also present a qualitative analysis of two dependency parsers. To the best of our knowledge, no recent work that considers the recent available tools, was carried out for the Portuguese language.

Keywords

natural language processing, evaluation of resources, portuguese language, part-of-speech tagging, named entity recognition, dependency parsing

1. Introdução

A área do Processamento da Linguagem Natural (PLN) encontra-se em profunda expansão e em Portugal não é excepção. Se há alguns anos os trabalhos computacionais ligados à língua portuguesa eram de pura investigação, hoje em dia várias empresas têm projetos neste campo, desenvolvendo sistemas de pesquisa em dados médicos, agentes virtuais, sistemas de tradução, etc. Do mesmo modo, vários agentes interessados utilizam ferramentas que operam sobre o português e, apesar de a língua inglesa continuar a ser imbatível em termos de recursos disponíveis, existem actualmente várias ferramentas gratuitas que oferecem modelos pré-treinados (ou facilmente treináveis) para a língua portuguesa, em



especial para as variantes do português europeu e do Brasil. Coloca-se, então, a questão de escolher a ferramenta mais adequada para a tarefa em mãos. Vários trabalhos têm-se focado na avaliação de ferramentas (Gamallo & Garcia, 2013), além das que têm sido levadas a cabo no quadro de diferentes *fora* de avaliação conjunta para a língua portuguesa, tendo como objectivo determinar o estado da arte em várias tarefas de PLN. Algumas destas avaliações conjuntas focaram-se na avaliação de ferramentas que realizam tarefas de base, nomeadamente Etiquetação Morfo-sintática (EMS) — Morfo-olimpíadas (Santos et al., 2003)¹ — e Reconhecimento de Entidades Mencionadas (REM) — MiniHAREM, primeiro e segundo HAREM (Santos & Cardoso, 2007; Santos et al., 2008), bem como IberLEF-2019 (Collovini et al., 2019). Outras avaliações visaram tarefas de mais alto nível, tais como determinar a similaridade semântica entre duas frases (Fonseca et al., 2016). Todas estas competições são tipicamente montadas por equipas de peritos em PLN, responsáveis pelos dados de treino, coleções douradas, etc. (Santos et al., 2003); do mesmo modo, em todas estas

competições participam usualmente equipas que fazem a sua investigação em PLN.

Ora, nos dias de hoje, mais do que saber exactamente qual o estado da arte na realização de uma tarefa, os não-especialistas precisam de poder decidir (rapidamente) que ferramentas usar. Dada toda a oferta actual, perdemo-nos facilmente na avaliação de ferramentas. Assim, neste artigo, focamo-nos na avaliação de ferramentas que realizam as tarefas de EMS e REM para a língua portuguesa (português europeu). No entanto, não é nosso objectivo comparar detalhadamente as várias ferramentas e escolher a vencedora com base em sofisticados *corpora* de referência anotados, tal como realizado em avaliações anteriores, nomeadamente nas levadas a cabo pela Linguateca², mas mostrar como, com base numa metodologia simples mas correcta, se pode ter uma ideia da utilidade de uma ferramenta e/ou modelos associados, consoante as necessidades da aplicação final em vista. Também não é nosso objectivo trazer os utilizadores para uma avaliação. No entanto, além da facilidade de instalação e de utilização de cada ferramenta, mostramos como estas podem ser testadas, dando pistas sobre a sua aplicabilidade. Todo o código usado, bem como recursos linguísticos criados, estão disponíveis³, com espe-

cial destaque para as tabelas de mapeamento de etiquetas. Estas permitem avaliar os sistemas sobre a mesma referência, sendo o ponto que maior dificuldades acarreta ao levar a cabo uma tarefa de avaliação com sistemas tão variados, os quais retornam etiquetas diferentes, em especial na tarefa de EMS. De notar, igualmente, que neste artigo vão ser avaliadas apenas ferramentas disponíveis publicamente e gratuitamente para a língua portuguesa. Além disso, apresentamos um estudo qualitativo de duas ferramentas que realizam a tarefa de Análise de Dependências (AD), tendo em conta vários factores, tais como a pontuação, a segmentação e a etiquetagem morfo-sintática, dos quais dependem as dependências obtidas. De notar que se trata de um estudo qualitativo das saídas de dois sistemas, sem qualquer pretensão de generalidade ou de quantificar essas observações.

As contribuições deste trabalho são, pois:

- a construção de dois *corpora* de referência, um para cada uma das tarefas (EMS e REM);
- a adaptação dos *corpora* de referência tendo em conta os diferentes pré-processamentos dos dados, realizados pelas diversas ferramentas;
- os *scripts* de conversão entre as etiquetas de cada ferramenta e as etiquetas dos *corpora* de referência;
- a avaliação de nove ferramentas (doze modelos diferentes) na tarefa de EMS;
- a avaliação de oito modelos distintos na tarefa de REM;
- a avaliação (qualitativa) de dois analisadores na tarefa de AD.

Apesar de, ao longo dos anos, vários trabalhos se focaram na avaliação de ferramentas para a língua portuguesa, não temos conhecimento de um trabalho similar a esta escala.

Este artigo está organizado como se segue: na Secção 2, apresentamos as ferramentas em análise e, na Secção 3, todo o *setup* experimental. A Secção 4 trata os resultados relativos à EMS e a Secção 5, os resultados relativos à tarefa de REM. Finalmente, na Secção 6, apresenta-se o estudo relativo à AD e, na Secção 7, resume-se as principais conclusões e discute-se o trabalho futuro.

¹<https://www.linguateca.pt/Morfolimpiadas/>

²<http://www.linguateca.pt>

³<https://gitlab.hlt.inesc-id.pt/lcoheur/ptools>

2. Ferramentas em análise

Nesta secção, são descritas as ferramentas e modelos pré-treinados, desenvolvidos para o processamento de português, para as tarefas de EMS e REM (e, em dois casos, de AD), e disponibilizados gratuitamente. A maioria destas ferramentas fornece modelos pré-treinados para as tarefas em estudo. As linguagens de programação destas ferramentas alternam entre o Java, o C++ e o Python, e, de um modo geral, apresentam documentação, o que torna relativamente fácil a sua instalação e utilização. Algumas destas ferramentas podem ser usadas em linha de comando, não exigindo muitos conhecimentos de programação.

2.1. FreeLing

O FreeLing (Padró, 2012)⁴ é uma biblioteca *open source* em C++, que disponibiliza modelos pré-treinados para o português, para (entre outras) as tarefas de EMS e REM. Encontra-se disponível um manual de utilizador⁵ completo e bem estruturado, no qual são descritos os procedimentos de instalação, importação para outras linguagens de programação, o sistema de etiquetas, etc. Apesar de a maioria das tarefas de FreeLing estar disponível através de linha de comandos, algumas funcionalidades apenas são acessíveis usando a ferramenta como biblioteca.

2.2. NLTK

O NLTK (Bird et al., 2009)⁶ é uma das plataformas mais utilizadas em PLN. Não dispõe de modelos pré-treinados para o português, mas oferece várias funcionalidades (implementadas em Python), bem como vários *corpora* (incluindo em português europeu), que permitem criar modelos para várias tarefas de PLN. Os *corpora* disponíveis para o processamento de texto em português fazem parte do projeto Floresta Sintática⁷. De notar que os *corpora* do Floresta Sintática permitem treinar modelos capazes de realizar a tarefa de EMS, mas não contêm informação sobre entidades nomeadas, pelo que a realização da tarefa de REM depende da existência de outros *corpora*.

O uso desta plataforma requer algum conhecimento em programação. Contudo, existe um

conjunto de programas para a linha de comandos chamado NLTK-Trainer⁸ que permite ao utilizador abstrair-se da programação, facilitando o treino de modelos presentes na ferramenta, a avaliação desses modelos e a análise de *corpora*. A instalação da plataforma NLTK encontra-se documentada para cada sistema operativo⁹. Por outro lado, a utilização das componentes que a ferramenta oferece é facilitada com exemplos de utilização, incluindo a realização de tarefas para a língua portuguesa¹⁰.

2.3. OpenNLP

O OpenNLP (Apache Software Foundation, 2014)¹¹ é uma biblioteca para Java, que fornece modelos pré-treinados, inclusive para português, para a tarefa de EMS. A documentação para a instalação da ferramenta não se encontra referenciada na página da ferramenta¹², o que dificultou a instalação. No entanto, no site da ferramenta, existe um guia de referência bem estruturado, que descreve os modos de utilização da ferramenta, o procedimento para treino de modelos e a execução de cada componente com base em modelos pré-treinados. Estas informações, são acompanhadas por exemplos. O OpenNLP oferece ainda um modo de utilização baseado na execução de programas na linha de comandos. Assim, para treinar, testar e aplicar esta ferramenta com modelos pré-treinados nas diferentes tarefas de PLN, não são exigidos conhecimentos de programação. Por ser uma biblioteca direcionada para a linguagem de programação Java, a sua importação para Python requer uma componente que faça a ligação. Para tal, neste trabalho usou-se a interface NLTK-OPENNLP¹³.

2.4. NLPyPort

O NLPyPort (Ferreira et al., 2019b,a)¹⁴ (Python) realiza as tarefas de EMS e REM, disponibilizando os modelos criados. Os recursos usados pelo NLPyPort, à excepção dos da tarefa de identificação do lema, são baseados em modelos e funções da ferramenta NLTK, previamente apresentada. À função genérica de divisão em tokens

⁴<http://nlp.lsi.upc.edu/freeling/index.php/>

⁵<https://freeling-user-manual.readthedocs.io/en/latest/>

⁶<http://www.nltk.org>

⁷<https://www.linguateca.pt/Floresta/>

⁸Acessível em <https://github.com/japerk/nltk-trainer> e documentado em <https://nltk-trainer.readthedocs.io/en/latest/>

⁹<https://www.nltk.org/install.html>

¹⁰http://www.nltk.org/howto/portuguese_en.html

¹¹<http://opennlp.apache.org/>

¹²O procedimento da instalação encontra-se na página <https://opennlp.apache.org/building.html>

¹³https://github.com/paudan/opennlp_python

¹⁴<https://github.com/jdportugal/NLPyPort>

da ferramenta NLTK (`word_tokenize`), o NLPy-Port adiciona uma função de identificação de pronomes clíticos e contrações. De notar que existe uma versão anterior desta ferramenta compatível com projetos desenvolvidos na linguagem de programação Java (Rodrigues et al., 2018)¹⁵.

2.5. Polyglot

O Polyglot (Al-Rfou, 2015)¹⁶ é uma biblioteca para Python que suporta funcionalidades que permitem realizar várias tarefas de PLN para diversas línguas, incluindo as de EMS e REM. Dispõe ainda de modelos pré-treinados capazes de levar a cabo ambas as tarefas em análise. Por consistir numa biblioteca (Python), a sua utilização pressupõe alguma experiência com programação. A documentação¹⁷ desta ferramenta encontra-se bem estruturada e os procedimentos para a sua instalação e importação estão descritos de forma clara. A documentação inclui também tutoriais e informações sobre cada tarefa, o que simplifica o seu uso.

2.6. SpaCy

O SpaCy (Honnibal et al., 2020)¹⁸ é uma biblioteca que incorpora modelos pré-treinados de várias línguas, inclusive de português, para as tarefas de EMS, REM e AD. Esta ferramenta foi pensada para ser importada como biblioteca em programas Python e não disponibiliza outras opções de uso. A sua documentação contém, entre outras informações, procedimentos para a sua instalação, importação para outras linguagens e realização das diferentes tarefas de PLN, acompanhados de exemplos, o que contribui para a sua usabilidade. No entanto, não existe documentação sobre as etiquetas usadas, o que complicou a sua avaliação no que respeita a tarefa de EMS.

2.7. StanfordNLP

O StanfordNLP (Qi et al., 2018)¹⁹ é uma biblioteca para Python, que permite realizar várias tarefas de PLN, incluindo EMS e REM. Tal como o SpaCy, também realiza AD. O StanfordNLP oferece modelos pré-treinados para 53 línguas, inclusive para português. Na realidade, o Stan-

fordNLP é uma interface para Python da ferramenta Stanford CoreNLP, em Java, que o grupo *Stanford NLP* disponibiliza. A interface para Python requer conhecimentos em programação. No entanto, as mesmas funcionalidades que o StanfordNLP apresenta podem ser executadas por via da linha de comandos, usando a ferramenta principal (Stanford CoreNLP). Existem tutoriais²⁰ que descrevem os passos da instalação e utilização desta biblioteca, assim como exemplos para as diferentes tarefas de PLN, nomeadamente para as de EMS e REM²¹.

2.8. TreeTagger

Das tarefas em estudo, o TreeTagger (Màrquez & Rodríguez, 1998)²² realiza apenas a tarefa de EMS para o português europeu, oferecendo modelos pré-treinados. Os procedimentos relativos à sua instalação e uso são descritos de forma clara no site da ferramenta. O modo de utilização de TreeTagger não implica conhecimentos de programação, pois pode ser realizado através da linha de comandos. A ferramenta pode ser igualmente importada para Python por via de uma componente intermediária, como por exemplo, `treetagger-python`²³. São também disponibilizados tutoriais²⁴.

2.9. LinguaKit

O LinguaKit (Gamallo & Garcia, 2017a) une várias ferramentas de processamento da língua natural permitindo a realização de tarefas como a lematização, análise de sentimentos, análise morfo-sintática, análise sintática, reconhecimento e classificação de entidades nomeadas, entre outras. As diferentes tarefas podem ser executadas através da linha de comandos ou pela versão web²⁵. No repositório²⁶ do LinguaKit encontram-se as instruções de instalação, procedimentos e exemplos de utilização da ferramenta através da linha de comandos.

²⁰https://stanfordnlp.github.io/stanfordnlp/installation_usage.html#getting-started

²¹Por falta de memória RAM foi necessário usar o servidor da Google (Google Colabs) para instalar a biblioteca (seguir estes passos: <https://stanfordnlp.github.io/stanfordnlp/>).

²²<https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

²³<https://github.com/miotto/treetagger-python>

²⁴<https://freeling-tutorial.readthedocs.io/en/latest/>

²⁵<https://www.linguaakit.com/es/analisis-completo>

²⁶<https://github.com/citiususc/Linguaakit>

¹⁵<https://github.com/rikarudo/NLPPORT>.

¹⁶<https://draquet.github.io/PolyGlot/>

¹⁷<https://polyglot.readthedocs.io/en/latest/Installation.html>

¹⁸<http://www.spacy.io>

¹⁹<http://stanfordnlp.github.io/stanfordnlp/>

2.10. Modelos pré-treinados para a tarefa de Reconhecimento de Entidades Mencionadas

O trabalho descrito por Pires (2017) fornece modelos pré-treinados para a tarefa de REM, para o português europeu. Estes modelos – doravante modelos-SIGARRA – foram treinados com as ferramentas OpenNLP, StanfordNLP, SpaCy²⁷ e NLTK, com base no *corpus* SIGARRA NEWS. Neste trabalho, vamos também avaliar estes modelos tendo em conta a tarefa de REM.

3. Setup Experimental

O primeiro passo foi criar um *corpus* de referência (ou *coleção dourada* ou *corpus* de teste) para as duas tarefas em análise. Esse *corpus* é descrito na Secção 3.1. O segundo passo foi atender a especificidades de cada ferramenta, o que levou a modificações do *corpus* de referência (Secção 3.2) no que respeita à segmentação feita pelas diferentes ferramentas. De seguida, foram ainda definidos vários *scripts* que transformam etiquetas do *corpus* criado nas etiquetas usadas pelas diferentes ferramentas (Secções 3.3 e 3.4 para a conversão de etiquetas na tarefa de EMS e REM, respetivamente). Finalmente, apresentam-se as medidas de avaliação, na Secção 3.6.

3.1. Construção do *Corpus* de Referência

O *corpus* de referência é composto por 101 frases retiradas de revistas, jornais e livros portugueses, disponíveis *on-line*. Para efeitos deste estudo considerou-se um *corpus* de referência reduzido, ainda que construído tendo em conta alguma diversidade e proporção das fontes utilizadas, nomeadamente, por ser constituído por diferentes tipos de texto e, dentro de cada tipo, com fontes diversas, para abranger diferentes usos da língua. Dessas frases, 59% pertencem a revistas como a Visão²⁸ e a Exame Informática²⁹, 29% são de jornais (jornal Observador³⁰ e as restantes do Público³¹) e 13% pertencem ao livro “O Príncipe com Orelhas de Burro”, de José Régio (Tabela 1).

Antes da anotação das frases, foram corrigidos pequenos erros ortográficos, pois este não era um

²⁷As instruções para o carregamento do modelo da ferramenta SpaCy não se ajustaram ao ficheiro, pelo que não foi possível usar esse modelo. A razão pode ter ficado a dever-se à incompatibilidade da versão mais recente da ferramenta com o modelo.

²⁸<https://visao.sapo.pt>

²⁹<https://visao.sapo.pt/exameinformatica/>

³⁰<https://observador.pt>

³¹<https://www.publico.pt>

Tipo	Fonte	%
Revistas	Visão	30
	Exame Informática	29
Jornais	Observador	27
	Público	2
Livro		13

Tabela 1: Composição do *corpus* de referência.

ponto de avaliação. Assume-se assim que os textos que chegam às diferentes ferramentas estão limpos de erros ortográficos.

Seguidamente, foram anotadas as classes gramaticais e informações de flexão associadas a cada palavra e respetiva classe. Por exemplo, para um substantivo, além do seu tipo (comum ou próprio), foram igualmente anotados o seu género e número. Da mesma forma, para os verbos foram anotados o modo, o tempo, a pessoa e o número. Repetiu-se este processo para as outras classes gramaticais. A Tabela 2 apresenta a frequência de cada classe e a Tabela 3 lista a frequência das entidades nomeadas na coleção dourada.

De notar que esta tarefa de anotação não foi realizada por um perito. A ideia não é ter um *corpus* perfeito sob o ponto de vista da anotação, mas um *corpus* que representasse o que um não-perito consideraria interessante/correto anotar.

3.2. Adaptação às diferentes ferramentas

Como foi dito, cada ferramenta apresenta as suas particularidades no que respeita ao modo como processa texto, o que implica que a coleção dourada anterior não possa ser usada sem antes ser pré-processada de acordo com as características de cada ferramenta. Por exemplo, a maior parte destas ferramentas não divide contrações. Assim, a coleção dourada teve de ser adaptada a algumas ferramentas: Polyglot, NLTK, SpaCy e OpenNLP. Por outro lado, as ferramentas FreeLing e LinguaKit permitem unir locuções e nomes compostos, sendo uma opção que decidimos explorar. Por exemplo, a locução “a partir de”, apesar de ter 3 palavras, é vista como uma unidade lexical, o que é representado ligando os seus elementos por *underscore*: “a_partir_de”. Esta particularidade implica que, para estas ferramentas, sejam anotadas como uma unidade as locuções e nomes compostos da coleção dourada, ao invés de serem anotados separadamente os elementos gramaticais que compõem essas expressões.

Classes	Sub-classes	Frequência
Adjetivo (Adj)		179
Advérbio (Adv)	Negação (N)	15
	Normal (G)	73
Conjunção (Conj)	Coordenativa (C)	93
	Subordinativa (S)	58
Determinante (Det)	Artigo (Art)	468
	Indefinido (Ind)	24
	Possessivo (Poss)	14
	Demonstrativo (Dem)	35
	Interrogativo (Int)	2
Nome (Nom)	Comum (C)	699
	Próprio (P)	109
Numeral (Num)		77
Preposição (Prep)		525
Pronome (Pron)	Pessoal (Pes)	23
	Indefinido (Ind)	4
	Relativo (Rel)	43
	Demonstrativo (Dem)	16
Verbo (Verb)	Auxiliar (Aux)	82
	Indicativo (Ind)	142
	Condicional (Cond)	3
	Conjuntivo (Conj)	5
	Gerúndio (Ger)	13
	Particípio Passado (PP)	98
	Infinitivo (Inf)	74

Tabela 2: Frequência de cada classe morfossintática na coleção dourada.

Classes	Frequência
Organização	30
Localização	22
Pessoa	8
Data	21

Tabela 3: Frequência de cada classe de entidades nomeadas na coleção dourada.

A forma como as unidades textuais são segmentadas (*segmentação*) varia igualmente.. Por exemplo, algumas ferramentas consideram “quarta-feira” como um único token e outras separam cada elemento da palavra (“quarta”, “-” e “feira”). Já o OpenNLP considera os verbos ligados a pronomes clíticos como um único token, como sucede em “reuniram-se” e em “escondê-lo”. Esta variedade de características, tornou árdua a compatibilidade da coleção dourada principal com todas as ferramentas, pelo que foram criadas diferentes coleções douradas, específicas para cada ferramenta.

No que respeita à tarefa de Reconhecimento de Entidades Mencionadas, foram consideradas as entidades específicas de cada uma das ferramentas, de modo a que as saídas das diversas ferramentas fossem compatibilizadas com as entidades consideradas na coleção dourada.

3.3. Sobre as Classes Morfossintáticas

Cada ferramenta tem o seu sistema de classes ou etiquetas morfossintáticas para as tarefas de PLN. O acesso a esta informação facilita a conversão automática das anotações da coleção dourada nas etiquetas das diferentes ferramentas. Esta tarefa não é, de todo, trivial devido às diferenças nas etiquetas das ferramentas e, em alguns casos, devido à indisponibilidade de um manual de utilizador que descrevesse o conjunto de símbolos possíveis. É o caso da ferramenta SpaCy, que não possui uma descrição das etiquetas morfossintáticas. No entanto, foi possível, na maioria dos casos, mapear sem problemas de maior, as etiquetas usadas pelo *corpus* de re-

ferência nas etiquetas das diferentes ferramentas. De notar ainda que algumas ferramentas contribuem com etiquetas com uma maior granularidade (por exemplo, o FreeLing, tal como se verá adiante).

3.3.1. NLTK, OpenNLP e NLPyPort

O sistema de etiquetas do NLTK corresponde ao conjunto de etiquetas com a categoria gramatical utilizado na anotação dos *corpora* da Floresta Sintática³². Quanto ao OpenNLP, não se encontrou, na documentação da ferramenta, uma descrição do conjunto de etiquetas para o modelo em português, apenas para a língua inglesa. No entanto, verificou-se que o sistema de etiquetas do OpenNLP é semelhante ao da ferramenta NLTK, existindo uma ligeira diferença entre ambos no tratamento de sinais de pontuação: o OpenNLP agrupa os sinais de pontuação sob uma única etiqueta, ao contrário do NLTK, que considera cada sinal de pontuação como uma classe distinta. Desta forma, para a conversão das anotações, procedeu-se de maneira semelhante à da ferramenta NLTK, à exceção dos sinais de pontuação que precisaram de um outro processamento. Por outro lado, as etiquetas consideradas nestas duas ferramentas não contêm informação de flexão. Além disso, o NLTK e o OpenNLP distinguem apenas verbos no infinitivo, gerúndio e particípio passado; todos as restantes formas verbais são classificadas de forma indiferenciada como “verbos finitos”. Relativamente aos tipos de Pronomes só são identificados 3 tipos (v.g. determinativos, pessoais e independentes). A existência de uma descrição do conjunto de etiquetas facilitou a conversão das anotações da coleção doumada, apesar de não ser totalmente clara a categorização de alguns determinantes e pronomes, como por exemplo a classe *pron-det* (pronomes determinativos). Por esta razão, as etiquetas relativas aos pronomes e determinantes serão tratadas à parte.

Quanto ao NLPyPort, avaliou-se apenas o modelo pré-treinado para a tarefa de EMS. Dado que os recursos da ferramenta NLPyPort se baseiam na ferramenta NLTK e o modelo para esta tarefa foi treinado com os *corpora* Bosque da Floresta Sintática e Mac-Morpho (Fonseca & Rosa, 2013), o conjunto de etiquetas morfossintáticas assemelha-se ao conjunto de etiquetas da ferramenta NLTK. Contudo, tal como no OpenNLP, os sinais de pontuação são agrupados numa mesma etiqueta, *punc*. Existe ainda uma etiqueta adicional (etiqueta por omissão)

³²<https://www.linguateca.pt/>

atribuída a um token quando o sistema não consegue identificar a sua classe gramatical. Essa etiqueta (“N”) corresponde à classe gramatical nome. Assim, existem duas etiquetas diferentes (“N” e “n”) para a classe nome comum e decidimos tratar estas duas etiquetas como sendo a mesma.

3.3.2. Polyglot e StanfordNLP

As ferramentas Polyglot e StanfordNLP baseiam as suas etiquetas nas da CoNLL-U (Buchholz & Marsi, 2006), compostas por vários elementos que descrevem morfossintaticamente uma palavra. Um dos elementos corresponde à classe gramatical universal, UPOS³³; outro dos elementos vem do FEATS³⁴, que descreve informações morfológicas associadas à palavra. Com este conjunto universal de etiquetas, a correspondência entre as anotações da coleção de referência e as classes gramaticais realizou-se de uma forma mais simples. No entanto, o Polyglot só apresenta informações sobre a classe gramatical universal. Em particular, o conjunto de etiquetas morfossintáticas usado nos modelos do Polyglot corresponde às classes gramaticais principais da Tabela 2, como adjetivo, determinante, advérbio, etc.

3.3.3. SpaCy

No que respeita ao SpaCy, e tal como dito anteriormente, apesar de ter uma função que permite obter a descrição de uma determinada etiqueta, o sistema não disponibiliza um glossário completo com a descrição das etiquetas relativas à análise de texto em português, o que dificultou a compreensão dos resultados obtidos. Assim, o seu sistema de etiquetas não é claro, apesar de, aparentemente, se basear igualmente nos mesmos formalismos standard do Polyglot e StanfordNLP. No entanto, partilha as etiquetas relativas aos pronomes e determinantes do NLTK e OpenNLP. A conversão automática tornou-se mais complexa para esta ferramenta.

3.3.4. FreeLing, LinguaKit e TreeTagger

No manual de utilizador da ferramenta FreeLing encontram-se descritos os conjuntos de etiquetas morfossintáticas e de entidades nomeadas. Além das classes gramaticais principais, as etiquetas contêm informações sobre outros aspetos morfológicos como o género, número, modo e tempo

³³<https://universaldependencies.org/u/pos/>.

³⁴<https://universaldependencies.org/u/feat/index.html>

verbal. Por exemplo, para os adjetivos, as classes existentes têm em conta o tipo, o género, grau e número. O sistema de etiquetas desta ferramenta é extenso, tornando a conversão automática complexa. Porém, a descrição clara de cada classe no manual de utilizador evitou dificuldades na correspondência entre as anotações da coleção dourada e as etiquetas. De acordo com os autores³⁵, o *LinguaKit* tem as mesmas etiquetas que o *FreeLing*³⁶. No entanto, foram encontradas pequenas diferenças.

O *TreeTagger* utiliza um sistema de etiquetas semelhante, no qual, além das classes gramaticais principais, são representadas outras informações morfológicas das palavras³⁷. No entanto, as classes não possuem o mesmo nível de especificidade que as do *FreeLing*. Por exemplo, apenas a classe dos nomes incorpora informações sobre flexão em género e número. A descrição do conjunto das classes também é clara. Quanto às etiquetas dos verbos, só apresentam informação sobre o tipo do verbo (auxiliar ou principal) e os modos verbais. Tal como na ferramenta *FreeLing*, as contrações são resolvidas automaticamente e identificadas com o sinal “+”. Por exemplo, a contração “das” é identificada com “SPS + DA” por ser a contração de uma Preposição (SPS) com o Artigo Definido (DA) (“de + as”).

3.4. Sobre as Entidades Mencionadas

Como dito anteriormente, consideraremos os modelos pré-treinados para a tarefa de REM, para o português europeu, tais como descritos em Pires (2017) (os modelos-SIGARRA) e treinados com as ferramentas *OpenNLP*, *StanfordNLP* e *NLTK*, com base no *corpus* SIGARRA NEWS³⁸. Este *corpus* é composto por 1.000 artigos retirados da secção de notícias do sistema de informação da Universidade do Porto (SIGARRA) e, de acordo com Pires (2017), o seu conjunto de etiquetas corresponde a oito classes relacionadas com o domínio dos *corpora*: Hora, Evento, Organização, Curso, Pessoa, Localização, Data e UnidadeOrganica.

No que respeita às restantes ferramentas, o *FreeLing* e o *LinguaKit* apresentam as seguintes classes de entidades e as respetivas etiquetas: Pessoa (NP00SP0), Localização (NP00G00), Organização (NP00O00) e Outros (NP00V00). Esta

última etiqueta corresponde a entidades nomeadas que não se integram em nenhuma das categorias anteriores. No entanto, por ser uma classe ambígua, tokens que não são reconhecidos como entidades na coleção dourada como “Verão” são classificados como Outros. Para facilitar a avaliação automática, foi adicionada mais uma etiqueta (“O”) que identifica os tokens que não são entidades. Finalmente, quanto ao *Polyglot*, este deteta apenas 3 classes de entidades: Pessoa (I-PER), Localização (I-LOC) e Organização (I-ORG). Tal como para a ferramenta anterior foi adicionada a etiqueta “O”, com o mesmo significado.

3.5. Sobre a Análise de Dependências

No que respeita à tarefa de AD, cinco frases foram aleatoriamente retiradas do corpus e processadas pelas ferramentas *SpaCy* e *StanfordNLP*. As frases são as seguintes:

1. Num comunicado enviado às redações, o gabinete do primeiro-ministro fazia saber que considerava que a interpretação da lei que defendia a demissão imediata de um governante por negócios de empresas de familiares com entidades públicas, mesmo que estas nada tivessem a ver com o titular de cargo político, “ultrapassa largamente, no seu âmbito e consequências, o que tem sido a prática corrente ao longo dos anos”.
2. Em 2016, vários membros do governo americano que estavam a prestar serviços em Havana, Cuba, assim como os seus familiares, começaram a queixar-se de uma série de sintomas neurológicos, incluindo dificuldades de concentração e memória, tonturas e problemas visuais e de equilíbrio.
3. Os sintomas foram associados à exposição a sons repentinos e de grande intensidade e volume de uma fonte desconhecida que os pacientes reportam ter ouvido nas suas casas e quartos de hotel.
4. Nem o seu marido, porém, voltou a reparar nesse indício que uma tarde lhe gelara nos lábios palavras de exprobração e cólera.
5. Um homem de 44 anos foi identificado.

Seguidamente, os resultados do processamento foram dados a um especialista, para análise, sem que fossem identificadas as ferramentas. Este analisou os resultados das frases dadas tendo em conta a pontuação, segmentação, etiquetagem morfosintática e, finalmente, as dependências.

³⁵<https://gramatica.usc.es/pln/tools/CitiusTools.html>

³⁶https://github.com/citiususc/Linguakit/blob/master/tagger/tagset_pt-es-gl.html

³⁷<https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/Portuguese-Tagset.html>

³⁸rdm.inesctec.pt/ro/dataset/cs-2017-006

3.6. Medidas de Avaliação

Estando as etiquetas da coleção dourada em conformidade com as etiquetas de cada ferramenta, o próximo passo consistiu em decidir que medidas usar na avaliação. Optámos por usar a Micro- e a Macro-Média da medida F1, tal como descritas por Jurafsky & Martin (2019) e implementadas no SCIKIT-LEARN³⁹, pois também são usadas por outros trabalhos nesta área, tais como o proposto por Garcia & Gamallo (2015).

4. Etiquetação Morfossintática

Começamos por discutir alguns aspectos relativos aos modelos criados pelo NLTK. De seguida, apresentamos e discutimos os resultados obtidos pelos diferentes modelos e ferramentas.

4.1. Sobre o Modelos do NLTK

Todas as ferramentas com a excepção do NLTK oferecem modelos pré-treinados. Assim, para esta ferramenta, foram treinados vários modelos. Um desses modelos é o modelo de Bigramas, tal como apresentado nos manuais do NLTK. Foram ainda implementados modelos baseados na Máxima Entropia e ainda um modelo denominado Perceptrão, pois o OpenNLP fornece modelos baseados nestes dois algoritmos e considerámos interessante a comparação. De notar que poderiam ter sido implementados modelos mais adequados à predição de sequências, como os HMM ou os CRF (Lafferty et al. (2001)) ou ainda modelos baseados em redes neuronais. No entanto, a ideia foi testar o que a ferramenta oferece directamente. Todos estes modelos foram treinados nos *corpora* “Floresta Sintática”, disponíveis com o NLTK⁴⁰. De notar que a versão disponível no NLTK da Floresta Sintática não contém alguns sinais de pontuação como dois-pontos <:>, reticências <...>, parênteses curvos e o símbolo de percentagem <%>, pelo que estes itens não são previstos pelos diferentes modelos. A divisão em tokens das frases foi realizada pela função *word_tokenize*⁴¹. A criação de qualquer um dos modelos não requer um grande esforço na ótica de um informático, pois consiste em importar uma biblioteca própria para

o efeito. No entanto, alguns modelos têm as suas especificidades. Assim, para o modelo de bigramas, cada bigrama é um *token*, tal como identificado pelo *tokenizador*; no caso de aparecer um *token* nunca visto no treino, decidiu-se atribuir, por omissão, a etiqueta “n” (classe gramatical Nome), por ser a classe mais comum. Nos casos em que os modelos requeriam a recolha de características (*features*) dos dados, foram usadas características muito simples, como a própria palavra, a palavra anterior e a seguinte, se a palavra corrente começava com maiúscula, etc. A seleção das *features* dita o desempenho da ferramenta. No entanto, está fora do âmbito deste trabalho desenvolver um estudo exaustivo sobre as *features* a utilizar.

4.2. Resultados Globais

A Tabela 4 apresenta os resultados globais dos diferentes modelos, tendo em conta a totalidade das etiquetas (de cada ferramenta) e considerando a Micro- e a Macro-média relativas à medida F1, tal como definidas anteriormente. Devemos realçar que estes valores não permitem uma comparação totalmente justa dos diferentes modelos, pois, como dito anteriormente, os valores de Micro- e Macro-Média da F1 são calculados com base no conjunto de etiquetas de cada ferramenta, que varia entre estas. Assim, estes valores devem dar apenas uma ideia geral. De modo a comparar de forma justa estes modelos, a secção seguinte detalha os valores para estas medidas sem ter em conta as informações de flexão.

4.3. Resultados por Classe Gramatical

A Tabela 5 permite comparar os diferentes modelos tendo em conta as categorias que têm em comum e sem ter em conta as informações de flexão. De notar que são apenas mostradas as classes gramaticais mais relevantes (‘Bg’ representa o modelo baseado em Bigramas, ‘P’ o Perceptrão, ‘ME’ a Máxima Entropia, ‘NLPy’ o NLPyPort, ‘PG’ o Polyglot, ‘TT’ o TreeTagger, ‘FL’ o FreeLing, ‘StfNLP’ o StanfordNLP e ‘ONLP’ o OpenNLP). As Tabelas 6 e 7 apresentam os resultados para os dois grupos de modelos que partilham algumas etiquetas específicas.

4.4. Discussão

Os melhores resultados globais são obtidos pelo OpenNLP, Máxima Entropia, 91% de Macro-Média e aos 94% de Micro-Média, porém, é importante realçar que o nível de detalhe do conjunto de etiquetas da ferramenta não é tão fino

³⁹<https://scikit-learn.org>

⁴⁰Os modelos disponibilizados pelo OpenNLP, SpaCy e StanfordNLP também utilizam este *corpus*.

⁴¹Esta função tem a particularidade de mudar as aspas iniciais <“> de uma frase para dois acentos graves <``> e as finais <”> para duas plicas <“”>. Por isso, é necessária uma reconversão para aspas, antes do treino e teste dos dados.

Ferramenta	Micro-Média	Macro-Média
NLTK (Bigramas)	0.87	0.69
NLTK (Perceptrão)	0.89	0.71
NLTK (Máxima Entropia)	0.93	0.74
NLPyPort	0.86	0.83
Polyglot	0.79	0.68
Spacy	0.90	0.47
FreeLing	0.90	0.58
LinguaKit	0.83	0.48
TreeTagger	0.91	0.73
StanfordNLP	0.91	0.61
OpenNLP (Perceptrão)	0.93	0.77
OpenNLP (Máxima Entropia)	0.94	0.91

Tabela 4: Micro- e Macro-Média relativos a F1 para os diferentes modelos.

Classes	Bg	NLTK		NLPy	PG	SpaCy	TT	FL	LinguaKit	StfNLP	ONLP		
		P	ME								ME	P	
Adj	0.76	0.82	0.83	0.75	0.63	0.89	0.92	0.96	0.81	0.94	0.88	0.85	
Adv	G	0.81	0.77	0.81	0.80	0.46	0.85	0.91	0.84	0.85	0.89	0.82	0.81
	N							0.97	0.97	0.93	0.97		
Conj	C	0.97	0.95	0.96	0.97	0.94	0.95	0.97	0.94	0.97	0.97	0.97	0.98
	S	0.56	0.72	0.60	0.60	0.56	0.68	0.87	0.88	0.75	0.74	0.70	0.70
Det	Art	0.95	0.93	0.96	0.97	-	0.97	0.94	0.99	0.95	0.98	0.96	0.97
Nom	C	0.82	0.91	0.93	0.81	0.93	0.95	0.94	0.98	0.92	0.98	0.94	0.94
	P	0.33	0.33	0.81	0.33	0.68	0.85	0.51	0.96	0.97	0.85	0.77	0.75
Num		0.84	0.95	0.98	0.83	0.72	0.96	0.98	0.95	0.80	0.97	0.94	0.97
Prep		0.95	0.95	0.96	0.97	0.85	0.96	0.97	0.99	0.95	0.96	0.96	0.96
Pron	Pes	0.97	0.90	0.90	1.00	-	0.83	0.98	0.84	0.70	1.00	0.91	0.94
Verb	Aux					0.00	0.84	0.00	0.00	0.00	0.73		
	Ind						0.88	0.82	0.83	0.81	0.85	0.96	0.97
	Cond	0.88	0.94	0.97	0.88		0.50	*	*	*	0.80		
	Conj					0.62	0.36	0.74	0.71	0.78	0.92		
	Ger	0.91	1.00	0.91	0.91		0.95	0.91	0.91	0.76	0.95	1.00	0.96
	PP	0.82	0.95	0.92	0.82		0.94	0.96	0.95	0.79	0.93	0.96	0.96
Inf	0.89	0.93	0.95	0.89		0.91	0.92	0.94	0.91	0.95	0.96	0.95	

Tabela 5: Valores de F1 para as categorias comuns. O * indica que o Condicional é visto como Indicativo por estes sistemas

Classes	Polyglot	TreeTagger	FreeLing	LinguaKit	StanfordNLP	
Det	Ind		0.96	1.00	0.96	0.91
	Poss		1.00	1.00	0.70	1.00
	Dem	0.79	0.89	1.00	0.89	0.97
	Int		0.00	0.00	0.00	1.00
Pron	Ind		0.46	0.89	0.43	0.75
	Rel	0.70	0.93	0.94	0.88	0.94
	Dem		0.79	0.86	0.76	0.90

Tabela 6: Valores de F1 para as ferramentas que têm as categorias **Det** e **Pron** mais finas

Classes	NLTK		ME	NLPy	SpaCy	OpenNLP	
	Bigramas	Perc.				ME	ME
Pron-det	0.83	0.79	0.86	0.88	0.81	0.89	0.88
Pron-indp	0.80	0.88	0.82	0.79	0.91	0.86	0.88

Tabela 7: Valores de F1 para as etiquetas **Pron-det** e **Pron-indp**. A primeira contém os determinantes, pronomes demonstrativos, pronomes interrogativos, pronomes possessivos e pronomes relativos; a segunda os pronomes indefinidos e outros pronomes de outras categorias que expressam imprecisão.

quanto nos modelos anteriores, pois, tal como referido anteriormente, o conjunto de etiquetas do OpenNLP não contém informações de flexão de número, género e tempos verbais (ao contrário de outras ferramentas como o FreeLing, o StanfordNLP e o TreeTagger, que apresentam etiquetas mais finas).

Quanto à avaliação por classe, a ferramenta StanfordNLP é aquela que apresenta maiores valores na previsão de verbos no conjuntivo e no condicional. O FreeLing apresenta melhor desempenho na classificação de adjetivos e conjunções subordinativas. O LinguaKit ultrapassa FreeLing na previsão de nomes próprios, alcançando 97% de F1-measure. Por fim, o TreeTagger destaca-se na previsão de advérbios (de notar que nesta ferramenta o condicional não é considerado modo mas sim um tempo verbal do modo indicativo).

Existem ainda algumas particularidades das ferramentas, que merecem ser discutidas. Por exemplo, como dito anteriormente, o FreeLing e o LinguaKit reconhecem nomes compostos como um único *token* (por exemplo, “Eduardo Cabrita” é tratado como um *token* único “Eduardo_Cabrita”). Esta particularidade auxilia a classificação de nomes próprios compostos, evitando assim que preposições e nomes próprios sejam incorretamente classificados, o que é constante nas outras ferramentas. Desta forma, estas duas ferramentas obtiveram o melhor desempenho na previsão de nomes próprios. Infelizmente estas ferramentas não classificam corretamente verbos auxiliares (VA). Neste aspecto, a melhor ferramenta é o SpaCy (84%), apesar de o StanfordNLP também conseguir identificar verbos auxiliares (73%). O FreeLing e a LinguaKit têm também outras características que as tornam interessantes: palavras relacionadas com datas são unidas e consideradas como um só *token*, à semelhança dos nomes próprios, como “Novembro de 2018”, que é tratado como “Novembro_de_2018”. Para a ferramenta FreeLing, o critério de atribuição desta classe é, contudo, pouco claro, por não haver uma des-

crição desta etiqueta. Estas ferramentas também unem numerais, por exemplo, consideram “10 mil milhões” como um único *token*. No que diz respeito apenas ao FreeLing, estas uniões não são totalmente precisas, levando a erros de classificação de alguns *tokens*, como acontece na expressão “15 e 35”, que é considerado como um só *token* “15_e_35”. Esta situação leva, posteriormente, a uma incorreta classificação, neste caso atribuindo a classe Interjeição.

5. Reconhecimento de Entidades

Nesta subsecção serão apresentados os resultados e conclusões sobre o desempenho das ferramentas na tarefa de REM.

5.1. Sobre os modelos-SIGARRA

No que diz respeito ao NLTK, tal como previamente indicado, estudaram-se três modelos-SIGARRA, pré-treinados no corpus SIGARRA NEWS: modelo baseado em Árvores de Decisão, no Naïve Bayes e na Máxima Entropia. Os modelos avaliados associados ao OpenNLP e StanfordNLP são também os referidos modelos-SIGARRA, treinados no mesmo corpus, mas com estas ferramentas.

5.2. Resultados Globais

A Tabela 8 mostra os resultados globais dos diferentes modelos, tendo em conta a Micro- e a Macro-Média relativas à medida F1. O FreeLing e o LinguaKit são os melhores sistemas quanto à Micro-Média e o StanfordNLP quanto à Macro-Média. De notar que, sendo os modelos do NLTK, StanfordNLP e OpenNLP, modelos-SIGARRA, há uma diferença substancial de valores quanto à Macro-Média (o modelo Naïve Bayes com 0.18 e o StanfordNLP 0.78), assunto que se discutirá mais à frente.

Ferramenta	Micro-Média F1	Macro-Média F1
NLTK (Árvore de Decisão)	0.97	0.47
NLTK (Naïve Bayes)	0.92	0.18
NLTK (Máxima Entropia)	0.97	0.35
Polyglot	0.98	0.76
FreeLing	0.99	0.77
LinguaKit	0.99	0.69
StanfordNLP	0.98	0.78
OpenNLP	0.97	0.46

Tabela 8: Micro- e Macro-Média relativos a F1 para os diferentes modelos.

5.3. Resultados por Tipo de Entidade

A Tabela 9 apresenta a comparação entre os valores de F1-measure de cada classe e para cada ferramenta na tarefa de REM. O StanfordNLP é a ferramenta com bons (ou os melhores) em praticamente todas as classes, à exceção das classes Localização e Organização, nas quais é a ferramenta FreeLing que se destaca.

5.4. Discussão

Como dito anteriormente, há uma grande diferença de valores quanto à Micro- e Macro-Média em alguns modelos. Na verdade, os resultados apresentados mostram quão enganadora pode ser a Micro-Média (F1). Esta medida tem em conta a soma de todos os Verdadeiros/Falsos Positivos/Negativos de todas as classes, sendo calculada posteriormente a Precisão e a Cobertura, e, finalmente, a F1. Ora, todos os casos que não são considerados como entidades mencionadas e não são de facto entidades mencionadas (isto é, o grosso das palavras, pois a maioria das palavras de um texto não são entidades mencionadas) contam como Verdadeiros Positivos, indicando que esta métrica não é nada informativa neste cenário em que as classes não são balanceadas.

Por outro lado, e como referido, os algoritmos usados têm um papel extremamente relevante na tarefa de NER. Os resultados dos diferentes algoritmos na base dos modelos-SIGARRA treinados com o NLTK ilustram bem essa situação.

É também interessante verificar como a estratégia de segmentação usada pelas ferramentas pode fazer a diferença. No caso do FreeLing e do LinguaKit, a divisão em tokens destas ferramentas foi um alicerce na classificação correta de entidades nomeadas, pelo simples facto de preservar nomes compostos, tornando possível a identificação de entidades compostas como, “Estados Unidos da América” e “Diário de Notícias”.

Outra característica que se realça corresponde à sua capacidade para identificar entidades nomeadas estrangeiras como “Sujoy Ghosh”, “Einstein” e “Xuekun Fang”. Em contrapartida, verificou-se que a ferramenta FreeLing considerou incorretamente alguns *tokens* presentes no início das frases como entidades, talvez por começarem com letra maiúscula. Segue-se um exemplo desse caso: “*Contactada pela Lusa...*”. Em comparação ao FreeLing, o LinguaKit apresenta piores resultados, principalmente na classificação de localizações e organizações. Estes são alguns exemplos de entidades classificadas erroneamente: “ANA”(Organização) e “América do Sul”(Localização) foram considerados como “Pessoa”, “Xuekun Fang” (Pessoa) e “Ásia do Sul” (Localização) como “Outros”.

A análise da classificação do Polyglot revelou que esta considera nacionalidades tais como “mexicanos”, “dinamarquesa” e “americanos” sempre como Localização. No entanto, esta ferramenta consegue identificar entidades nomeadas estrangeiras como “Einstein”, “Kaiserslautern” e “Sujoy Ghosh” e algumas entidades nomeadas compostas como “Estados Unidos”, “Universidade de Aveiro” e “Physical Review”. Já o OpenNLP tem dificuldades a identificar entidades nomeadas estrangeiras, como por exemplo, “Netflix”, “Maximilian Gunther”, “Einstein”; por outro lado, não é totalmente capaz de classificar nomes compostos. Por exemplo, “Associação de Proteção e Socorro” e “Universidade de Londres” são corretamente identificadas. Contudo, outras entidades como “Universidade da Califórnia”, “Diário de Notícias”, “Estados Unidos” e “Eduardo Cabrita” já não o são. Outra conclusão retirada é que não identifica siglas como “EUA”, “MIT”, “PSML”, “EHT”, etc.

O StanfordNLP, por seu turno, consegue identificar entidades nomeadas noutras línguas, incluindo siglas. No entanto, verificou-se que, quando uma sigla está entre parênteses, a fer-

Classes	NLTK							
	FreeLing	LinguaKit	Polyglot	AD	NB	EM	OpenNLP	StanfordNLP
Data	–	–	–	0.78	0.11	0.74	0.76	0.92
Localizacao	0.92	0.82	0.73	0.51	0.06	0.17	0.00	0.34
Organizacao	0.84	0.63	0.61	0.62	0.34	0.58	0.70	0.75
Pessoa	0.67	0.63	0.70	0.42	0.00	0.35	0.32	0.88

Tabela 9: Valores de F1 de cada ferramenta, tendo em conta as entidades nomeadas da coleção dourada.

ramenta identifica os parênteses como fazendo parte da entidade nomeada. Por exemplo, dada a sigla MIT neste formato “(MIT)”, além de MIT ser considerada como Organização, o parêntese “()” também o é. De entre as ferramentas analisadas para esta tarefa, o StanfordNLP apresenta um melhor desempenho em praticamente todas as classes, à excepção da classe Localização e Organização, onde o vencedor é o FreeLing, como anteriormente referido.

6. Análise de Dependências

Nesta secção, apresentamos o estudo qualitativo relativo à tarefa de AD. Vários parâmetros são analisados: pontuação, segmentação, etiquetagem morfosintática e dependências. Como dito anteriormente, este trabalho foi executado sem conhecimento dos sistemas que produziram as anotações. Assim, doravante, o Sistema A corresponde ao SpaCy e Sistema B ao StanfordNLP.

6.1. Pontuação

Os dois sistemas adotam diferentes estratégias de segmentação relativamente aos sinais de pontuação e à sua análise.

No Sistema A, os sinais de pontuação (vírgulas, aspas e pontos finais) aparecem ligados ao token anterior, e.g. (Frase 1) *redacções*, ; *públicas*, ; *político*, “ ; *anos*”. (negritos nossos), e não parecem receber qualquer análise, não havendo nenhuma dependência associada. Note-se, além disso, que as aspas junto a *político* emparelham com as aspas ligadas a *anos*, mas este emparelhamento não parece ser feito pelo sistema.

Por outro lado, no Sistema B, os sinais de pontuação são tratados como tokens independentes e sobre eles recai (sobretudo) uma dependência específica: *punct*. No entanto, não é evidente o significado que a dependência exprime, quando se considera o elemento que opera sobre o sinal de pontuação, a sua função na frase ou outros sinais com que se encontra articulado. Assim, por exemplo, na Frase 1 (Figura 1), o adjunto

deslocado para o início de frase (v.g. *Em um comunicado enviado a as redacções*,) apresenta a dependência *punct* ligando o particípio *enviado* à vírgula; este particípio depende de/modifica (dependência *acl*, *adjectival clause*, *clausal modifier of noun*) o nome *comunicado*, que é a cabeça ou núcleo deste constituinte, pelo que todos os outros elementos flecham direta ou indiretamente sobre este nome. Por sua vez, este nome depende (*obl*; complemento oblíquo) do verbo principal da oração seguinte (*fazia*). Ora, se a função da vírgula neste exemplo é assinalar a deslocação (anteposição) do adjunto complemento de *fazer*, a partir da sua posição básica, para o início da frase; esta forma de representação, não dá conta de forma clara da função sintática que a vírgula exerce, já que não faz depender a vírgula do núcleo sintático do complemento (*comunicado*), mas sim de um dos seus modificadores (*enviado*).

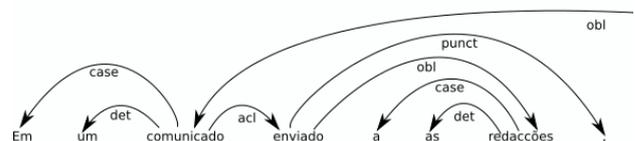


Figura 1: “Num comunicado enviado às redacções, ...”, Sistema B (Frase 1).

Mais adiante, e ainda no Sistema B e na mesma frase, duas vírgulas delimitam a oração subordinada concessiva (, *mesmo que estas nada tivessem a ver com...*) e encontram-se (e bem!) “emparelhadas”, estando ambas dependentes do verbo finito desta oração (*tivessem*). O mesmo sucede (Figura 2), no complemento adjunto (... *político* , “ *ultrapassa largamente* , *no seu âmbito e consequências* ,), também delimitado por vírgulas, e que são emparelhadas e postas na dependência do verbo anterior (*ultrapassa*). Contudo, no caso das aspas, as primeiras (de abertura), estas são feitas depender do verbo *ultrapassa* que está à sua direita e (estranhamente) com a dependência *nsubj*, que corresponde à função de sujeito; enquanto as segundas (de fecho), são colocadas na dependência de *anos*

(no fim da frase, fora da Figura 2), mas agora com a etiqueta **amod**, que corresponde à função de modificador (adjetival) de nome.

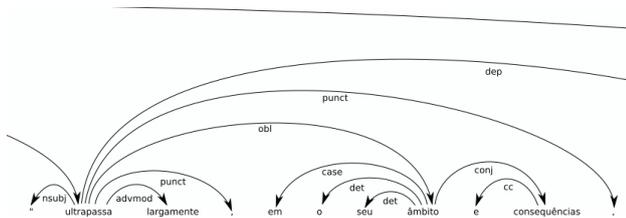


Figura 2: “ “ultrapassa largamente, em o seu âmbito e consequências,” , Sistema B (Frase 1).

É, assim, óbvia a inconsistência da anotação das dependências sobre os sinais de pontuação, quer pelo uso de etiquetas que não podem ser aplicadas a este tipo de tokens (**nsubj** e **amod**), quer pelos elementos textuais de que se faz depender os sinais de pontuação, quer ainda pelo incorreto emparelhamento de sinais que funcionam em conjunto (aspas e vírgulas).

6.2. Segmentação e etiquetagem morfosintática

Ambos os sistemas apresentam diversos problemas de segmentação de texto (delimitação de tokens) e de marcação de categorias morfosintáticas. Várias unidades lexicais multipalavras (ou palavras compostas/locuções) são analisadas como tokens distintos. No Sistema A, por exemplo, encontramos *primeiro*/**adj** e *ministro*/**noun** e não o nome composto *primeiro-ministro*/**noun**; *mesmo*/**adv** e *que*/**conj** e não a locução conjuntiva (ou conjunção composta *mesmo que*/**conj**; *cargo*/**noun** e *político*/**adj** e não o composto *cargo político*/**noun**; e a sequência *a*/**prep** *o*/**art** *longo*/**adj** *de*/**prep** não é tratado como uma locução preposicional (ou preposição composta), *ao longo de*/**prep**. Refira-se, ainda, a misteriosa etiqueta **x** (Figura 3) atribuída ao pronome da (assim chamada) oração relativa sem antecedente (Veloso (2013)), v.g. *o/X que tem sido*.

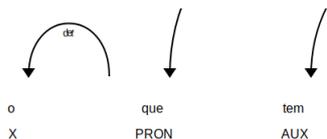


Figura 3: Etiqueta **x**, Sistema A (Frase 1).

Já o Sistema B trata como um só token o composto *primeiro-ministro* e liga por uma dependência **fixed** o elemento *mesmo* a *que* na

conjunção composta *mesmo que*. As restantes locuções desta frase também não são identificadas por este sistema. Na frase 2, a locução conjuncional *assim como*/**conj** também não foi reconhecida pelos sistemas. As restantes frases não apresentam expressões multipalavras, exceto, talvez, o composto *quarto de hotel*, na frase 3, que nenhum dos sistemas analisa como um único *token*.

Alguns problemas de etiquetagem morfosintática podem ainda ser observados na saída do Sistema A e parecem difíceis de explicar. Na frase 2, o nome *membros* é marcado como **propn** (nome próprio); também o elemento *como* (da locução *assim como*) é curiosamente etiquetado como **noun**. O nome *anos*, no fim da frase 1 e o o adjetivo-nome *familiares*, na frase 2, a que se encontram ligados um ponto final e uma vírgula, respetivamente, aparecem etiquetados como **sym** (symbol), talvez pelo facto de os sinais de pontuação não terem sido tratados como *tokens*. Contudo, tal só ocorre nestes dois nomes, havendo várias outras situações idênticas, com estes sinais de pontuação, em que tal não sucede.

6.3. Dependências

Os sistemas apresentam análises muito semelhantes, pelo que analisaremos os problemas com base na saída do Sistema A, indicando as diferenças mais relevantes. Nesta forma de representação, as dependências são marcadas sobre os arcos do grafo de dependências e estabelecem-se sempre entre palavras/*tokens*, isto é, entre os elementos que são núcleos (cabeça) dos constituintes sintáticos e os elementos que deles dependem, e.g. *a prática*: *a* ←(**def**) *prática*. Os constituintes da frase (e.g. grupos nominais, preposicionais, etc.) não são diretamente delimitados, mas podem ser deduzidos a partir do grafo de dependências, já que todos os elementos se ligam direta ou indiretamente ao seu núcleo. Repare-se que se adota uma representação *top-down* da dependência, em que um elemento hierarquicamente superior (*governor*) flecha sobre o elemento diretamente abaixo (*dependent*), independentemente da sua disposição linear na frase, e.g. *enviado às redações*: *enviado* (**obl**) → *redações*, *redações* (**det**) → *as*.

Para maior facilidade de comparação, em ambas as saídas, foram utilizadas as já referidas dependências universais (de Marneffe et al., 2014), em cuja descrição nos baseámos para a análise destas saídas. Assim na Frase 1 (Figura 1), o complemento indireto (dativo) *às redações* encontra-se (corretamente!) ligado ao

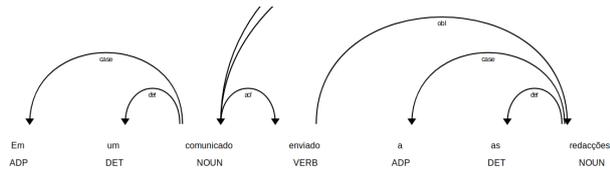


Figura 4: “Num comunicado...”, Sistema A

particípio *enviado*, mas pela dependência *obl* (*oblique*, oblíquo) e não, como se esperaria, por *iobj* (*indirect object*).

Ambos os sistemas fazem depender o complemento preposicional *Em comunicado enviado às redações*, deslocado para o início da frase, do verbo *fazia* por meio da dependência *obl*. Contudo, o sistema A extrai também uma dependência *appos*, que consideramos espúria, entre *comunicado* e *gabinete*. Tal pode dever-se ao facto de *primeiro-ministro* não ter sido tratado como um único *token*. Efetivamente, o Sistema B reconhece o composto e analisa-o corretamente como complemento de nome (*nmod*) de *gabinete*, enquanto o Sistema A apenas estabelece essa dependência com o elemento *primeiro*. Por essa razão também, o sujeito (*nsubj*) de *fazia*, no sistema B, é (corretamente) *gabinete*, ao passo que no Sistema A, que não considera o composto, o sujeito deste verbo é *ministro*.

Outro aspeto relacionado com a incorreta segmentação é o facto de a preposição composta *ao longo de*, que forma a expressão temporal *ao longo dos anos*, não ter sido identificada, pelo que o seu elemento *longo/noun* é analisado, por ambos os sistemas, como um mero complemento de nome (*ncomp*) de *prática*. O facto de, aparentemente, não se ter tratado a expressão como uma entidade mencionada de tempo (Hagège et al., 2010) leva a supor que, se este nome não estivesse construído com um verbo copulativo (*vido*), a expressão temporal iria ficar a depender de qualquer constituinte anterior e não do verbo (pleno) de tal frase.

Ainda neste sentido, repare-se que a locução *tivessem a ver* (praticamente sinónima de *estar relacionado com*) não deverá ter sido reconhecida por nenhum dos sistemas, que a analisam como uma mera construção de auxiliar (Baptista et al., 2010), a qual, aliás, aparentemente só existe em português com um valor modal: “Só tenho a/posso/devo dizer que...”, e que, claramente não é a construção aqui empregue. Vemos, assim, a importância do reconhecimento das expressões multpalavra e do impacto que têm na análise sintática (dependências) das frases.

Outro aspeto interessante é a análise da oração relativa sem antecedente, introduzida por *o* (marcado em A com a categoria *x*). Este elemento é tratado como determinante do pronome *que* (não se indica a subclasse de pronome, nomeadamente, se é relativo ou de outro subtipo). Esta análise poderia ser considerada correta no caso das *verdadeiras* relativas sem antecedente, resultantes da redução de um nome elidido que fosse cabeça desse constituinte e antecedente do pronome relativo (e.g. *Havia vários livros. Comprei o [livro] que tinha capa azul*). O determinante poderia, nesse caso variar em género e número consoante o nome que determina (e.g. *Havia várias camisas. Comprei as [camisas] que não tinham colarinho*). Ora, neste caso, trata-se de uma construção diferente, em que *o* é invariável (**as que têm sido prática corrente*), e cuja análise pode ser aproximada do pronome relativo *o qual*.

Um outro problema detetado, envolvendo também determinantes, é a dependência *det* entre *nada* e *estas* (v.g. *mesmo que estas nada tivessem a ver com*), o que faz considerar que as duas palavras foram analisadas como um só constituinte. Trata-se aqui do chamado “emprego pronominal” de *estas* (= *estas entidades públicas*) e não do “emprego determinativo”, que forma um constituinte autónomo, sujeito de *tivessem*; e do pronome indefinido *nada* que é aqui complemento deste verbo. Provavelmente, por essa razão, a dependência de sujeito (*nsubj*) estabelece-se entre *nada* e o verbo, apesar de a flexão deste último não autorizar essa relação, já que viola a concordância gramatical sujeito-verbo.

Um dos problemas mais difíceis de tratar é a imbricação de elementos coordenados. Na formalização em grafo aqui adotada, a dependência *conj* liga os elementos coordenados, sendo sempre orientada da esquerda para a direita, enquanto uma dependência *cc* liga o segundo membro da coordenação à conjunção coordenativa, sendo igualmente sempre orientada, mas da direita para a esquerda. Podemos ver um caso complexo deste problema na Frase 2, cuja hierarquia representamos por parênteses numerados: (...) *incluindo* [₁*dificuldades de* [₃*concentração*]₃ e [₄*memória*]₄]₂, [₅*tonturas*]₅ e [₆*problemas* [₇*visuais*]₇ e [₈*de equilíbrio*]₈]₆]₁. Esta estrutura leva a considerar os seguintes pares de elementos coordenados (cuja numeração mantemos, para maior clareza): *concentração*₃ (*conj*) → *memória*₄, *visuais*₃ (*conj*) → *de equilíbrio*₄; e a tripla coordenação, assinalada com uma vírgula e a conjunção *e*, que se repre-

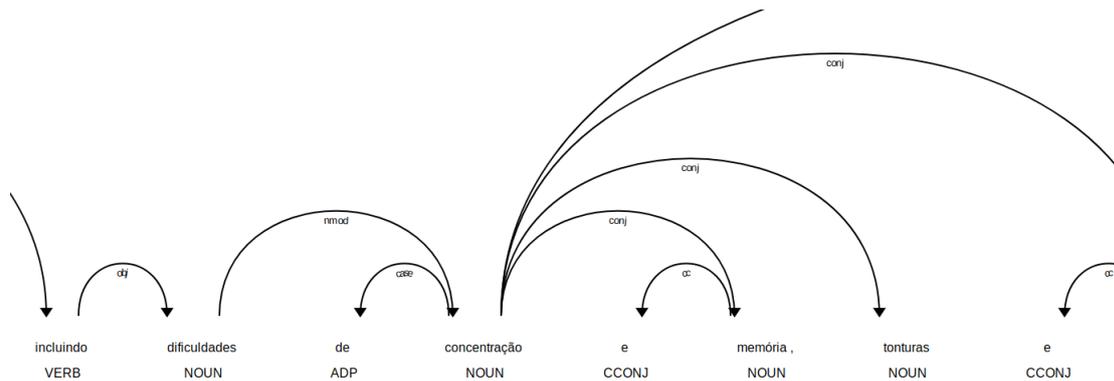


Figura 5: "... incluindo dificuldades..."

senta por duas dependências com foco no primeiro elemento: $dificuldades_2$ (conj) \rightarrow $tonturas_5$ e $dificuldades_2$ (conj) \rightarrow $problemas_6$.

Ora, os sistemas apresentam uma saída igual (Figura 5), onde apenas o primeiro par é corretamente capturado. A Frase 3 apresenta um problema similar. Aparentemente, o facto de apenas um dos sistemas tratar explicitamente a pontuação (neste caso, a vírgula) não parece fazer diferença nos resultados.

Noutras situações, os erros parecem resultar de dificuldades de análise de dependência a longa distância entre os elementos relacionados. Assim, na oração relativa explicativa da Frase 3 (Figura 6), v.g. *sons (...), que os pacientes reportam ter ouvido*, o pronome relativo *que*, referente a *sons*, deveria ter sido analisado como complemento direto (obj) de *ouvido* e não de *reportam*.

Ao seguir a definição e exemplos das dependências universais (de Marneffe et al., 2014), a descrição dos advérbios parece particularmente problemática, já que ignora distinções substanciais, algumas bem conhecidas pelas gramáticas mais tradicionais (para uma síntese moderna, v. (Molinier & Levrier, 2000)). Assim, não se distinguem: (i) o valor determinativo de *nem* no grupo nominal *nem o seu marido* (frase 4); (ii) o advérbio (de quantificação) *largamente*, que funciona como mero modificador verbal de *ultrapassa*, em *ultrapassa largamente* (frase 4); e o advérbio conjuntivo (com valor adversativo) *porém* (frase 4), que deve ser considerado como modificador de toda a frase, ligando-a à frase anterior. Ainda que os argumentos da dependência possam ser considerados corretos, todos estes advérbios são representados pela mesma dependência (advmod), sem os diferenciar. Por outro lado, nenhum dos sistemas reconhece o valor adverbial da expressão temporal *uma tarde* (frase 5), o que dá origem a um conjunto de dependências incorretas.

6.4. Discussão

Em jeito de síntese, é possível dizer que: (i) os problemas de segmentação e de etiquetagem morfosintática estão na origem não só das diferenças entre os sistemas comparados, mas também dos principais erros de análise sintática (dependências) encontrados. (ii) Neste sentido, reveste-se de especial importância a identificação das unidades lexicais compostas e expressões multipalavras, praticamente ignoradas por um dos sistemas e muito incompleta no outro. (iii) Nos restantes casos, os sistemas têm um comportamento muito semelhante ou mesmo idêntico. (iv) A pontuação é um elemento fundamental na análise sintática, mas um dos sistemas aparentemente ignora-a, enquanto o outro parece não a usar, do que resultam diversos erros.

Entre os diversos problemas de análise sintática detetados, salientamos (v) o tratamento da coordenação e da imbricação de elementos ordenados, bem como (vi) a ausência de distinção entre funções sintáticas muito diferentes, desempenhadas por vários tipos sintático-semânticos de advérbios, e todas colapsadas sob a mesma dependência "universal". Trata-se de fenómenos linguísticos bem conhecidos, por vezes complexos e dificilmente tratáveis, mas que o desenvolvimento de sistemas de processamento computacional de português terá de enfrentar. O levantamento destas e de outras situações problemáticas é fundamental para construir um *roadmap* dos desafios a vencer.

7. Conclusões e Trabalho Futuro

Neste trabalho fizemos a avaliação de várias ferramentas (a maioria através de modelos pré-treinados) para as tarefas de EMS e REM, para a língua portuguesa. Apresentámos ainda um estudo qualitativo sobre duas ferramentas que re-

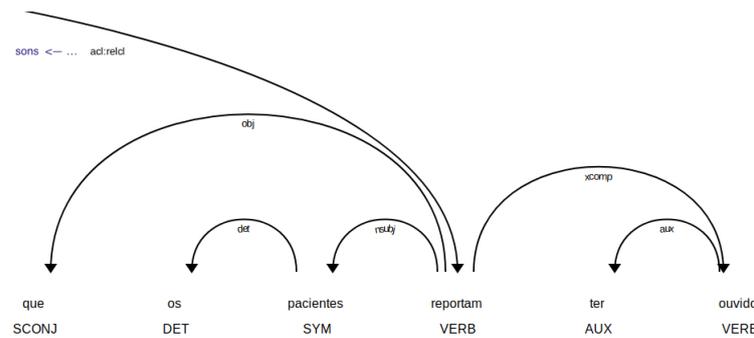


Figura 6: “sons ... que os pacientes reportam ter ouvido...”

alizam a tarefa de AD. Não há uma ferramenta claramente vencedora, pois, dependendo das necessidades de cada um, uma ferramenta pode ser mais ou menos apropriada, de acordo com vários factores. Fica, no entanto, claro, que, para além da utilização de colecções de etiquetas variadas, há uma grande diferença entre as ferramentas se considerarmos o modo como a segmentação é feita. Por outro lado, há ferramentas que são mais finas em algumas classes de etiquetas, o que torna mais difícil a obtenção de bons resultados. No entanto, mais uma vez, dependendo das necessidades dos seus utilizadores, pode fazer sentido ou não usar essas etiquetas mais finas em detrimento de melhores resultados. De notar também que as decisões tomadas a nível da segmentação, bem como os resultados atingidos em termos de EMS (e REM), vão afectar a AD. Todos os *corpora* usados, bem como o mapeamento de etiquetas, serão tornados públicos⁴², permitindo a replicação das experiências e facilitando futuras avaliações na mesma linha. Como trabalho futuro, o *LinguaKit* (Gamallo & Garcia, 2017b; Gamallo et al., 2018) deverá ser avaliado na tarefa de AD. Por outro lado, podem também ser explorados mais modelos, por exemplo, com o NLTK. Uma outra experiência a realizar, seria usar o mesmo corpus de treino para treinar vários modelos de maneira a poder comparar diretamente e apenas os diferentes algoritmos.

Agradecimentos

Este trabalho foi parcialmente suportado pela Fundação para a Ciência e a Tecnologia através dos projectos UIDB/50021/2020 e PTDC/LLT-LIN/29887/2017, financiando este último a bolsa de Matilde Gonçalves.

⁴²<https://gitlab.hlt.inesc-id.pt/lcoheur/ptools>

Referências

- Al-Rfou, Rami. 2015. *Polyglot: A massive multilingual natural language processing pipeline*: State University of New York at Stony Brook. Tese de Doutoramento.
- Apache Software Foundation. 2014. OpenNLP natural language processing library. <http://opennlp.apache.org/>.
- Baptista, Jorge, Nuno Mamede & Fernando Gomes. 2010. Auxiliary verbs and verbal chains in European Portuguese. Em *Computational Processing of the Portuguese Language (PROPOR)*, 110–119. doi:10.1007/978-3-642-12320-7_14.
- Bird, Steven, Ewan Klein & Edward Loper. 2009. *Natural language processing with Python*. O’Reilly Media, Inc.
- Buchholz, Sabine & Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. Em *Conference on Computational Natural Language Learning*, 149–164.
- Collovini, Sandra, Joaquim Francisco Santos Neto, Bernardo Scapini Consoli, Juliano Terra, Renata Vieira, Paulo Quaresma, Marlo Souza, Daniela Barreiro Claro & Rafael Glauber. 2019. IberLEF 2019 Portuguese named entity recognition and relation extraction tasks. Em *Proceedings of the Iberian Languages Evaluation Forum (IberLEF)*, 390–410.
- Ferreira, João, Hugo Gonçalo Oliveira & Ricardo Rodrigues. 2019a. Improving NLTK for processing Portuguese. Em *Symposium on Languages, Applications and Technologies (SLATE)*, 18:1–18:9. doi:10.4230/OASIcs.SLATE.2019.18.
- Ferreira, João, Hugo Gonçalo Oliveira & Ricardo Rodrigues. 2019b. NLPyPort: Named entity recognition with CRF and rule-based relation

- extraction. Em *Iberian Languages Evaluation Forum (IberLEF)*, 468–477.
- Fonseca, Erick, Leandro Borges dos Santos, Marcelo Criscuolo & Sandra Aluísio. 2016. Visão geral da avaliação de similaridade semântica e inferência textual. *Linguamática* 8(2). 3–13.
- Fonseca, Erick Rocha & João Luís G. Rosa. 2013. Mac-Morpho revisited: Towards robust part-of-speech tagging. Em *Brazilian Symposium in Information and Human Language Technology (STIL)*, s/pp.
- Gamallo, P., M. Garcia, C. Piñeiro, R. Martínez-Castaño & J. C. Pichel. 2018. LinguaKit: A big data-based multilingual tool for linguistic analysis and information extraction. Em *Conference on Social Networks Analysis, Management and Security (SNAMS)*, 239–244. doi 10.1109/SNAMS.2018.8554689.
- Gamallo, Pablo & Marcos Garcia. 2013. FreeLing e treetagger: um estudo comparativo no âmbito do Português. Relatório técnico. Universidade de Santiago de Compostela.
- Gamallo, Pablo & Marcos Garcia. 2017a. LinguaKit: A multilingual tool for linguistic analysis and information extraction. *Linguamática* 9. 19–28. doi 10.21814/lm.9.1.243.
- Gamallo, Pablo & Marcos Garcia. 2017b. LinguaKit: uma ferramenta multilingue para a análise linguística e a extração de informação. *Linguamática* 9(1). 19–28. doi 10.21814/lm.9.1.243.
- Garcia, Marcos & Pablo Gamallo. 2015. Yet another suite of multilingual NLP tools. Em *Languages, Applications and Technologies*, 65–75. doi 10.1007/978-3-319-27653-3_7.
- Hagège, Caroline, Jorge Baptista & Nuno Mamede. 2010. Caracterização e processamento de expressões temporais em português. *Linguamática* 2(1). 63–76.
- Honnibal, Matthew, Ines Montani, Sofie Van Landeghem & Adriane Boyd. 2020. spaCy: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. doi 10.5281/zenodo.1212303.
- Jurafsky, Dan & James H. Martin. 2019. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson.
- Lafferty, John D., Andrew McCallum & Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Em *International Conference on Machine Learning*, 282–289.
- de Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre & Christopher D. Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. Em *Conference on Language Resources and Evaluation (LREC)*, 4585–4592.
- Màrquez, Lluís & Horacio Rodríguez. 1998. Part-of-speech tagging using decision trees. Em *Machine Learning*, 25–36. doi 10.1007/BFb0026668.
- Molinier, Christian & Françoise Levrier. 2000. *Grammaire des adverbes: description des formes en -ment*. Genève: Droz.
- Padró, Lluís. 2012. Analizadores multilingües en FreeLing. *Linguamática* 3(2). 13–20.
- Pires, André Ricardo Oliveira. 2017. *Named Entity Extraction from Portuguese web text*: Faculty of Engineering of University of Porto. Tese de Mestrado.
- Qi, Peng, Timothy Dozat, Yuhao Zhang & Christopher D. Manning. 2018. Universal dependency parsing from scratch. Em *CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies CoNLL Shared Task*, 160–170. doi 10.18653/v1/K18-2016.
- Rodrigues, Ricardo, Hugo Gonçalo Oliveira & Paulo Gomes. 2018. NLPPort: A pipeline for Portuguese NLP. Em *Symposium on Languages, Applications and Technologies (SLATE)*, 18:1–18:9. doi 10.4230/OASICS.SLATE.2018.18.
- Santos, Diana & Nuno Cardoso. 2007. Balanço do primeiro HAREM e perspectivas de trabalho futuro. Em *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*, 87–94. Linguateca.
- Santos, Diana, Luís Costa & Paulo Rocha. 2003. Cooperatively evaluating portuguese morphology. Em *Computational Processing of the Portuguese Language (PROPOR)*, 259–266. doi 10.1007/3-540-45011-4_40.
- Santos, Diana, Hugo Gonçalo Oliveira, Cláudia Freitas, Cristina Mota & Paula Carvalho. 2008. Segundo HAREM: Balanço e perspectivas de futuro. Apresentação no Encontro do Segundo HAREM.
- Veloso, Rita. 2013. *Gramática do Português*, vol. 2 chap. Subordinação relativa, 2061–2134. Fundação Calouste Gulbenkian.