Proposta recebida em Março 2017 e aceite para publicação em Junho 2017.

LinguaKit: uma ferramenta multilingue para a análise linguística e a extração de informação

LinguaKit: a multilingual tool for linguistic analysis and information extraction

Pablo Gamallo
Centro Singular de Investigação de Tecnologias da Informação (CiTIUS)
Universidade de Santiago de Compostela
pablo.gamallo@usc.es

Marcos Garcia
Grupo LyS, Departamento de Letras
Faculdade de Filologia, Universidade da Corunha
marcos.garcia.gonzalez@udc.gal

Resumo

Este artigo apresenta LinguaKit, uma *suite* multilingue de ferramentas de análise, extração, anotação e correção linguísticas. LinguaKit permite realizar tarefas tão diversas como a lematização, a etiquetagem morfossintática ou a análise sintática (entre outras), incluindo também aplicações para a análise de sentimentos (ou minaria de opiniões), a extração de termos multipalavra, ou a anotação concetual e ligação a recursos enciclopédicos tais como a DBpedia. A maior parte dos módulos funcionam para quatro variedades linguísticas: português, espanhol, inglês e galego. A linguagem de programação de LinguaKit é Perl, e o código está disponível sob a licença livre GPLv3.

Palavras chave

extração de informação, tecnologia linguística

Abstract

This paper presents LinguaKit, a multilingual *suite* of tools for analysis, extraction, annotation and linguistic correction. LinguaKit allows the user to perform different tasks such as lemmatization, PoS-tagging or syntactic parsing (among others), including applications for sentiment analysis (or opinion mining), extraction of multiword expressions or conceptual annotation and entity linking to DBpedia. Most part of the developed modules work in four linguistic varieties: Portuguese, Spanish, English, and Galician. The system is programmed in Perl, and it is freely available under a GPLv3 license.

Keywords

information extraction, linguistic technology

1 Introdução

Neste artigo apresentamos LinguaKit, um pacote de ferramentas multilingues para o Processamento da Linguagem Natural (PLN), que contém módulos de análise, extração, anotação e correção linguística. Os diferentes módulos que compõem LinguaKit são interdependentes entre si, e estão organizados mediante uma arquitectura de *pipeline*. Permite realizar um vasto conjunto de tarefas de PLN, entre as quais: (i) identificação de orações e tokenização, (ii) lematização, (iii) etiquetagem morfossintática, (iv) identificação e (v) reconhecimento de entidades mencionadas, (vi) análise sintática de dependências, (vii) resolução de correferência a nível de entidade, (viii) extração de termos e (ix) de relações semânticas, (x) análise de sentimentos (minaria de opiniões), (xi) anotação conceitual com ligação a recursos enciclopédicos, (xii) correção e avaliação de léxico e sintaxe, (xiii) conjugação verbal automática, (xiv) resumo automático (sumarização), (xv) identificação de língua, ou (xvi) visualização de concordâncias (palavras chave em contexto).

As ferramentas foram desenhadas e desenvolvidas utilizando diferentes estratégias de PLN, tanto de base simbólica como estatística, com aprendizagem supervisionada, não supervisionada e semi-supervisionada. A maior parte dos módulos de LinguaKit funcionam em português, galego,[1] espanhol e inglês.[2]

[1] Neste trabalho consideramos *português* a variedade escrita utilizando as diferentes ortografias da Academia Brasileira de Letras e da Academia das Ciências de Lisboa, e *galego* a que segue (com maior ou menor fidelidade) as normas publicadas em Real Academia Galega e Instituto da Lingua Galega (2004).

[2] Exceto o sistema de correção e avaliação linguística —

DOI: 10.21814/lm.9.1.243

LinguaKit foi programado em Perl. Está disponível como um serviço web[3] e é acessível via RESTful API.[4] O código fonte está publicado sob uma licença GPL.[5]

A tabela 1 mostra os módulos da *suite* organizados em quatro categorias: análise básica, análise profunda, sistemas de extração, e aplicações linguísticas.

Uma das principais contribuições desta nova suite em código aberto é a criação de um ecossistema de ferramentas com diferentes níveis de complexidade. No primeiro nível, situam-se os módulos básicos de análise, que são utilizados para construir aqueles com uma complexidade maior, nomeadamente módulos de análise profunda e de extração. E estes, por sua vez, servem para desenvolver aplicações cada vez mais complexas, como a ferramenta de correção/avaliação linguística ou o anotador semântico.

O objetivo do presente artigo é descrever a arquitetura de LinguaKit, mencionando as metodologias utilizadas na implementação de cada módulo, e apresentar aquelas ferramentas que ainda não tinham sido tratadas em trabalhos precedentes.

Para além desta introdução, o artigo está organizado da seguinte maneira. Na secção 2 incluímos uma breve revisão do trabalho relacionado, e a secção 3 mostra a arquitetura do sistema. A seguir, apresentamos diferentes avaliações —já publicadas— dos diferentes módulos (secção 4), uma descrição pormenorizada dos extratores de termos (secção 5), e as conclusões do presente trabalho (secção 6).

2 Trabalho relacionado

Dado que existem numerosas ferramentas de PLN para diversas línguas e em várias linguagens de programação, nesta secção apresentamos sucintamente algumas das mais conhecidas e utilizadas *suites* de PLN em código aberto, tendo em conta também as línguas que cada uma delas suporta.

O *software* de PLN mais conhecido é provavelmente Stanford CoreNLP (Manning et al., 2014), que inclui módulos de análise tais como tokenizadores, etiquetadores morfossintáticos, reconhecedores de entidades, analisadores sintáticos, sistemas para a resolução da correferência, etc. Está escrito em Java e foi desenvolvido principalmente para o inglês, embora recentemente se tenham publicado modelos para diversas línguas como o chinês, o espanhol ou o árabe, entre outras.

FreeLing (Padró, 2011) é uma outra *suite* de PLN (escrita em C++) que inclui uma lista semelhante à de Stanford CoreNLP, mas dispõe de ferramentas para outras tarefas como a transcrição fonética ou a desambiguação semântica. A maior parte dos módulos analisa os textos em catalão, espanhol, português, galego, inglês, francês, e recentemente, alemão ou russo (entre outras línguas).

Um outro sistema de PLN escrito em Java é OpenNLP,[6] que realiza tarefas de análise similares aos que já foram referidos, mas que inclui, por exemplo, um módulo de categorização de documentos. Existem modelos disponíveis para várias línguas, nomeadamente inglês, espanhol e alemão.

Também programada em Java, IXA pipes (Agerri et al., 2014) é uma *suite* modular que realiza as tarefas mais habituais de processamento linguístico: tokenização, etiquetagem morfossintática, reconhecimento de entidades e análise sintática. Este sistema permite processar as seguintes línguas (com variações em função do módulo escolhido): espanhol, inglês, eusquera, italiano e galego.

Com a popularização da iniciativa *Universal Dependencies*,[7] que promove a unificação das diretrizes de anotação em diversas línguas, têm vindo a ser desenvolvidas algumas ferramentas compatíveis, como UDPipe (Straka et al., 2016). UDPipe inclui módulos de aprendizagem automática para tokenização, etiquetagem morfossintática, lematização e análise sintática.

Como foi referido, existem mais sistemas que realizam tarefas de PLN —alguns com objetivos ligeiramente diferentes, ou escritos noutras linguagens de programação—, tais como NLTK: *Natural Language Toolkit* (Bird et al., 2009), amplamente utilizado no ensino de PLN, ou spaCy[8] (mais focado em uso industrial), ambos escritos em *python*.

Para além dos diferentes *softwares* apresentados, cabe mencionar também CitiusTools (Garcia & Gamallo, 2015), *suite* de PLN a partir da qual foram desenvolvidos alguns dos módulos de LinguaKit. À diferença dos sistemas mencionados, que oferecem fundamentalmente módulos de análise, LinguaKit possui também um amplo le-

desenvolvido principalmente para a análise do galego—, e o conjugador verbal — que não funciona para o inglês.

[3] https://www.linguakit.com

[4] https://market.mashape.com/linguakit/linguakit-natural-language-processing-in-the-cloud

[5] https://github.com/citiususc/Linguakit

[6] http://opennlp.apache.org/

[7] http://universaldependencies.org/

[8] https://spacy.io/

tipo de módulo	módulos
análise básica	conjugador verbal
	segmentador de orações
	tokenizador e *splitter*
análise profunda	lematizador
	PoS-tagger
	identificador de entidades (NER)
	classificador de entidades (NEC)
	identificador de correferência
	analisador sintático em dependências
extração	palavras chave
	expressões multipalavra
	análise de sentimento/opinião
	relações semânticas (open IE)
aplicações	sumarização
	anotação semântica (com EL)
	concordâncias (palavras chave em contexto)
	identificação de línguas
	correção/avaliação linguística (léxica e gramatical)

Tabela 1: Módulos de LinguaKit organizados em quatro categorias.

que de ferramentas de extração, bem como de aplicações mais complexas baseadas nesses sistemas de extração.

3 Arquitetura

A figura 1 mostra as dependências entre os diferentes módulos apresentados na tabela 1, sendo esta arquitetura comum às quatro línguas processadas pelo sistema.

A análise básica consiste na segmentação de um texto em orações, que são a entrada do processo de tokenização. Por sua vez, o texto tokenizado é melhorado com regras básicas de *splitting*, que separam os elementos que compõem contrações (e.g., "*do* → *de o*", em português e galego) ou sequências de verbo e pronome clítico (e.g., "*comelo* → *comer o*", em galego). Este último módulo é dependente da língua, enquanto os processos anteriores são realizados com uma ferramenta única (utilizando listas de abreviaturas também dependentes de cada variedade linguística).

O conjugador verbal é um módulo isolado que toma como entrada um verbo em infinitivo tanto em espanhol como em galego e português. Neste último caso, o sistema pode realizar até quatro modelos de conjugação verbal, em função quer da variedade (português de Portugal ou do Brasil), quer do sistema ortográfico utilizado (antes ou depois do Acordo Ortográfico de 1990).[9]

[9] https://pt.wikipedia.org/wiki/Acordo_Ortografico_de_1990

Com base nos módulos de análise básica, foram implementadas duas aplicações diferentes: um identificador de língua e um gerador de concordâncias (palavras chave em contexto). O identificador de língua é também utilizado internamente pelo sistema para fazer a escolha automática dos módulos de uma ou outra língua, permitindo que o utilizador possa analisar um texto sem ter de selecionar a língua desejada.

Os módulos de análise profunda tomam como entrada a saída da análise básica. O primeiro processo é a lematização, que atribui todos os lemas e todas as etiquetas possíveis a cada forma (já tokenizada) do texto de entrada. O lematizador baseia-se num léxico computacional disponível para cada língua. Antes do processo de desambiguação realizado pelo etiquetador morfossintático (*PoS-tagger*, na tabela 1), é possível identificar as entidades mencionadas ou nomes próprios (NER). As entidades identificadas pelo NER serão classificadas após a etiquetagem morfossintática mediante um sistema de classificação semântica: o classificador de entidades mencionadas (NEC). O último módulo de análise é o *parsing* sintático em dependências, que toma como entrada o etiquetador morfossintático (com ou sem aplicação dos módulos de NER e NEC).

Várias ferramentas utilizam a saída dos módulos de análise profunda para extrair informação dos textos: extratores de opiniões (também conhecidos como analisadores de sentimento), de palavras chave, de expressões multipalavra, e de relações semânticas. Todos estes extratores tomam como entrada a saída do módulo

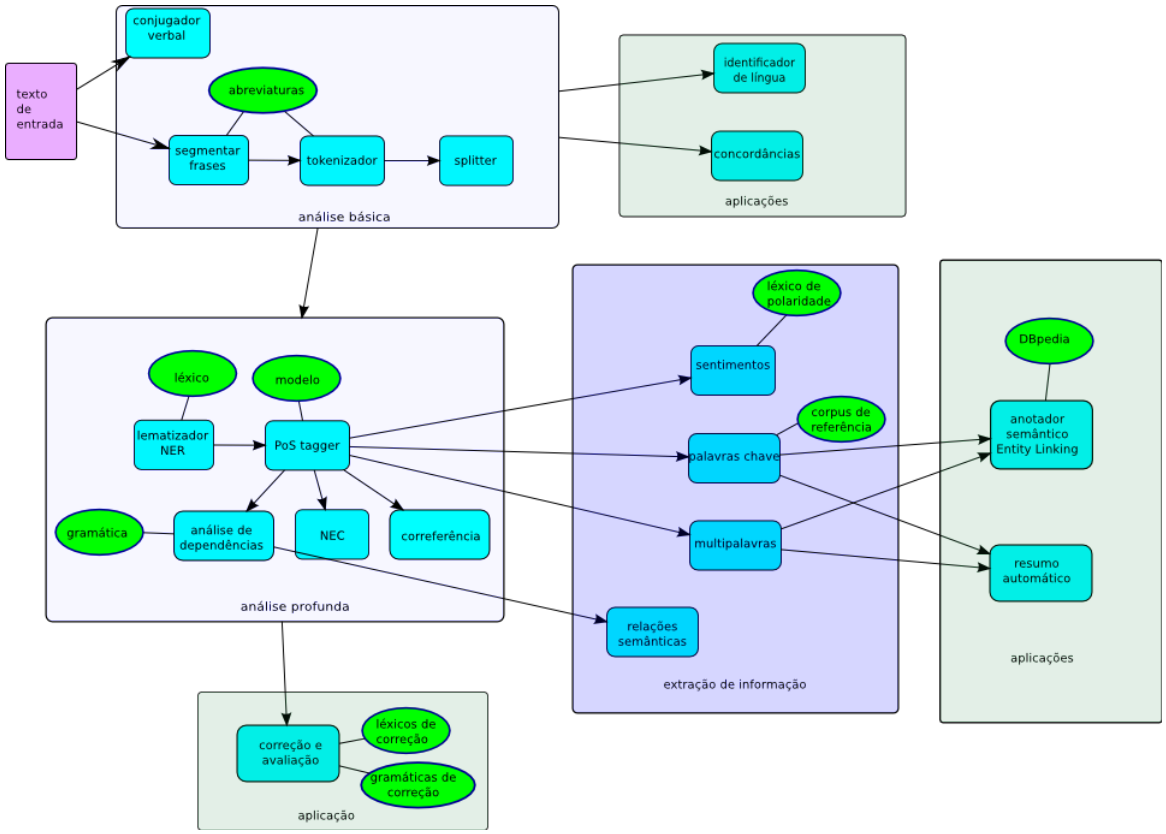

Figura 1: Arquitetura de LinguaKit.

de etiquetagem morfossintática. Para além disso, foi desenvolvida uma aplicação de correção lexical e gramatical que utiliza a saída do analisador sintático.

Finalmente, duas aplicações foram criadas a partir dos extratores de termos relevantes (isto é, palavras chave e expressões multipalavra): um gerador automático de resumos e um anotador semântico, que liga os termos extraídos a conceitos enciclopédicos armazenados em bases de conhecimento externas (por exemplo, a DBpedia).[10]

4 Módulos

Os principais módulos de LinguaKit foram desenhados e implementados nos últimos cinco anos, sendo a maior parte deles descritos em diferentes publicações. Assim, esta secção tem como objetivo pôr em conjunto as técnicas e metodologias empregadas em cada um dos principais módulos, bem como um breve resumo das avaliações realizadas.

[10] http://wiki.dbpedia.org/

Pré-processamento

Como foi referido, os primeiros módulos realizam um pré-processamento do texto que permite aplicar com maior precisão as ferramentas subsequentes: estes módulos realizam identificação de fronteiras de oração (com base em máquinas de estados finitas e em listas de abreviaturas que terminam com pontuação), de tokenização e *splitting* (processos pelos quais são separados os diferentes tokens de cada oração), e de lematização (que atribui um —ou mais— lemas possíveis a cada um dos tokens). Descrições mais pormenorizadas destes módulos podem encontrar-se em (Garcia & Gamallo, 2010) ou em (Garcia & Gamallo, 2015).

Etiquetagem morfossintática

Este módulo desambigua as etiquetas morfossintáticas[11] previamente atribuídos a cada token mediante um classificador *bayesiano* baseado em bigramas de tokens. Foi avaliado para três

[11] E também alguns lemas cuja atribuição varia em função da categoria morfossintática à que pertença o token. Por exemplo, as formas galegas/portuguesas *cala* ou *calas* podem ter como lema *calar* —se forem verbos—, ou *cala* —se forem nomes.

línguas: inglês, português e espanhol, com resultados próximos ao estado da arte: ≈ 96 para português e espanhol, e ligeiramente mais baixos (≈ 94%) para inglês (Gamallo et al., 2015b; Garcia & Gamallo, 2015).

Identificação e classificação de entidades mencionadas

O primeiro destes módulos identifica expressões *numex* (de base numérica) e *enamex* (nomes próprios) mediante máquinas de estados finitas, que têm em conta tanto as formas ortográficas (uso de maiúsculas) como palavras funcionais que possam conter (***U**niversidade **de** **S**antiago **de** **C**ompostela*). Uma vez identificadas as entidades, o módulo de classificação aplica um método de supervisão distante que lhe permite classificar as entidades em quatro classes: *pessoa*, *organização*, *local* ou *miscelânea*. O sistema emprega listas de entidades já conhecidas (*gazetteers*) e um conjunto de regras que permitem desambiguar as entidades que aparecem em mais de uma lista (que podem ser, por exemplo, *pessoa* ou *local*). Os *gazetteers* foram extraídos automaticamente de fontes externas com conhecimento enciclopédico.

Este módulo foi avaliado para as quatro línguas analisadas (inglês, português, espanhol e galego), utilizando diversos corpora e sendo comparando com sistemas supervisionados (Gamallo & Garcia, 2011; Garcia et al., 2012; Garcia & Gamallo, 2015). Os resultados obtidos —apesar de que não são sempre diretamente comparáveis— foram próximos aos atingidos por FreeLing e Stanford CoreNLP, superando nitidamente os modelos disponibilizados para OpenNLP.

Resolução de correferência a nível de entidade

Um outro módulo de análise linguística incluído em LinguaKit é o de resolução de correferência a nível de entidade. Este módulo utiliza como entrada um texto com as entidades mencionadas classificadas semanticamente, e aplica uma estratégia determinística baseada em filtros mediante os quais atribui um identificador numérico a cada uma das ocorrências (menções) das entidades previamente analisadas. Idealmente, este identificador será igual para cada uma das menções que refiram a mesma entidade do discurso (e.g., "António_Variações$_{\text{PESSOA_1}}$", "John$_{\text{PESSOA_2}}$", "John_Lennon$_{\text{PESSOA_2}}$", "António$_{\text{PESSOA_1}}$", "Lennon$_{\text{PESSOA_2}}$", ...). Este módulo é uma versão simplificada do apresentado em (Garcia & Gamallo, 2014).

Para além disso, este sistema inclui uma saída alternativa que aproveita a resolução de correferência para tentar corrigir erros prévios da classificação semântica. Assim, se a citada forma "Lennon" tivesse sido anteriormente classificada como *local*, mas identificada como menção da mesma entidade que "John_Lennon", a etiqueta semântica da primeira seria corrigida para *pessoa* (Garcia, 2016).

Analisador em dependências

O módulo de análise sintática, chamado DepPattern, baseia-se em regras formais de dependências e num algoritmo de *parsing* com técnicas de estados finitos. Foi avaliado para português e espanhol e comparado com MaltParser (Nivre et al., 2007), um *parser* determinístico de transições baseado em aprendizagem supervisionada. Os resultados obtidos por DepPattern com corpora de teste construído a partir de textos de diferentes domínios foram semelhantes aos obtidos por MaltParser: ≈ 82% de F-score (Gamallo, 2015).

Em Gamallo & González (2011) descrevem-se as características principais da gramática formal na qual se baseia o conhecimento linguístico de DepPattern. Um compilador transforma as regras formais, escritas com os princípios da gramática de dependências, em *scripts* Perl que representam os *parsers* de estados finitos.

Análise de sentimentos

O sistema de análise de sentimentos (tarefa também conhecida como minaria de opiniões) classifica uma oração como tendo uma opinião positiva, negativa ou neutra. O núcleo deste módulo é um classificador *bayesiano* treinado com texto previamente anotado com as opiniões referidas, que também utiliza um léxico de polaridade e regras sintáticas para a identificação de marcadores linguísticos que intensificam ou mudam a polaridade das palavras. Foi avaliado para inglês e espanhol, e participou em duas competições focadas na análise de opiniões em redes sociais: TASS 2013 (Gamallo et al., 2013a) para espanhol, e SemEval-2014 (Gamallo & Garcia, 2014) para inglês, mostrando um desempenho competitivo em ambas as línguas.

Extrator de relações

Este módulo consiste num sistema de extração de informação não supervisionado cujo obje-

tivo é obter um conjunto aberto de relações entre dous objetos. As relações (ou tripletas: *obj1,relação,obj2*) selecionadas por um sistema de extração de informação aberta (*Open Information Extraction*, OIE) representam as proposições básicas do texto de entrada. O nosso sistema, argOE (Gamallo & Garcia, 2015), está baseado em regras e toma como entrada um texto analisado em dependências em formato CoNLL-X. Foi avaliado em inglês, português e espanhol, e comparado com sistemas de OIE focados na extração numa única língua. O módulo incluído em LinguaKit melhora os resultados de muitos dos sistemas com os quais foi comparado, como ReVerb (Etzioni et al., 2011), embora os resultados sejam mais baixos do que um outro sistema baseado em regras, ClausIE (Corro & Gemulla, 2013).

Anotação e ligação semântica

Este módulo identifica os termos relevantes do texto que podem ser ligados a conceitos presentes em bases de dados externas, tais como a DBpedia. Esta tarefa, que consiste em relacionar os termos mencionados no texto e os conceitos de uma base ontológica e enciclopédica, é normalmente conhecido como *ligação de entidades* (*entity linking*, EL). O nosso sistema utiliza como recursos externos algumas relações da DBpedia e uma nova base construída mediante similaridade distribucional a partir das entradas textuais da Wikipedia. Foram avaliadas as versões portuguesa e inglesa (Gamallo & Garcia, 2016), com resultados similares a outros sistemas EL de referência, como *DBpedia Spotlight* (Mendes et al., 2011).

Corretor linguístico

O sistema de correção linguística de LinguaKit está, por enquanto, só disponível como módulo experimental na versão web.[12]

Esta ferramenta foi desenvolvida principalmente para galego, variedade na qual foi avaliada e comparada com revisões manuais de textos por parte de docentes profissionais (Gamallo et al., 2015a). O sistema contém diversos módulos que identificam e classificam diferentes tipos de erros habituais em aprendizes de galego, tanto de tipo léxico (castelhanismos, hipercorreções, etc.), como gramatical (concordância de género e número, posição dos pronomes átonos, etc.).

Existem, contudo, versões básicas para português e espanhol, mas precisam de um maior desenvolvimento no que diz respeito a recursos linguísticos tais como listas de tipologias de erros, ou regras sintáticas para a identificação e classificação de erros.

[12] https://linguakit.com/es/supercorrector

Outras ferramentas

Para além das ferramentas referidas (e das aplicações de extração mostradas na secção 5), LinguaKit também inclui as seguintes aplicações: (i) um gerador automático de resumos (sumarizador), (ii) um visualizador de palavras chave em contexto (concordâncias), e (iii) conjugadores verbais automáticos.

O sumarizador extrai as frases ou orações mais relevantes do texto de entrada. Utiliza a segmentação de orações, a análise morfossintática, e os extratores de palavras e multipalavras para ponderar as orações em graus de relevância. A partir da lista ponderada de orações, o usuário escolhe a percentagem de texto que quer extrair para construir o resumo.

O visualizador de concordâncias, também conhecido como *key word in context*, é uma ferramenta útil para estudos em linguística de corpus que procura no texto selecionado a palavra escolhida pelo utilizador, obtendo o seu contexto anterior e posterior em cada uma das suas ocorrências.

O módulo de conjugação verbal permite obter de modo automático a conjugação completa de um verbo a partir da sua forma em infinitivo. O sistema contém as regras de conjugação verbal do espanhol peninsular, do galego e de quatro normas do português: duas variedades diatópicas: português europeu e brasileiro; e duas variantes ortográficas para cada uma das anteriores: antes e depois do Acordo Ortográfico de 1990. Uma vez que o conjugador funciona aplicando diferentes regras em função do paradigma verbal, este pode gerar as formas conjugadas de verbos desconhecidos, tais como neologismos. Para além disso, identifica se o verbo é conhecido, com base em listas de verbos obtidos de recursos académicos para cada uma das línguas (Gamallo et al., 2013b).

Usabilidade

Para executar qualquer módulo em linha de comandos, disponibilizamos de um *script*, *linguakit*, que requer três argumentos: língua, nome do módulo e ficheiro TXT a ser processado. Por exemplo, o comando que faz a chamada básica do módulo de etiquetagem morfossintática em português é o seguinte:

```
./linguakit pt tagger input.txt
```

Com este comando, o utilizador não precisa de conhecer quais os módulos que dependem da etiquetagem (segmentação, tokenização, etc). De facto, o código executado por *linguakit* é um *pipeline* de *scripts*, cada um deles representando um módulo da suite. No caso da etiquetagem morfossintática para um texto em português, o *pipeline* invocado é o seguinte:

```
cat input.txt
|./tagger/pt/sentences-pt_exe.perl
|./tagger/pt/tokens-pt_exe.perl
|./tagger/pt/splitter-pt_exe.perl
|./tagger/pt/lemmas-pt_exe.perl
|./tagger/pt/tagger-pt_exe.perl
```

Na próxima versão de LinguaKit, os módulos poderão ser invocados também mediante funções Perl.

5 Extratores de termos

Uma vez apresentados os módulos e aplicações que já tinham sido avaliadas em diferentes publicações, nesta secção mostramos duas ferramentas de extração, que têm como objetivo identificar e selecionar os termos chave e relevantes de um texto. Consideram-se termos relevantes aquelas expressões mais importantes de um texto que são utilizadas como índices para —entre outras aplicações— a deteção imediata do tema ou tópico, para o etiquetado textual automático, ou bem para a classificação de documentos. Estes dous módulos de extração diferenciam-se no tipo de termos relevantes que extraem: (i) unidades monolexicais e nomes próprios (*termos básicos*), e (ii) unidades plurilexicais (*termos multipalavra*).

Termos básicos

Chamamos termos básicos àquelas unidades lexicais relevantes para um texto que se codificam como nomes comuns, nomes próprios (simples ou compostos), adjetivos e verbos. Exceto os nomes próprios, que podem ser expressões compostas por várias palavras (por exemplo, "Nova Iorque", "Universidade Nova de Lisboa", etc), os termos básicos são palavras simples monolexicais. O método de extração leva-se a cabo em duas fases: seleção de candidatos e ordenação por relevância.

Na primeira fase, o sistema identifica todos os candidatos a serem termos básicos mediante o etiquetador morfossintático. Deste modo, selecionam-se como candidatos todas as unidades lexicais que foram etiquetadas como nomes (comuns e próprios), adjetivos e verbos.

Na segunda fase, os termos ordenam-se por relevância e escolhem-se os N primeiros, sendo N um valor numérico parametrizável. Para calcular a relevância dos termos básicos recorremos à noção de *termhood*, é dizer, ao grau com que a unidade linguística está relacionada com conceitos específicos do domínio do texto (Kageura & Umino, 1996). Esta noção de *termhood* pode ver-se também como a probabilidade de um termo formar parte do domínio. O *termhood* não é, portanto, uma medida discreta, mas contínua. Em consequência, medimos a relevância de um termo básico (*termhood*) mediante um peso estatístico que é calculado contrastando as frequências dos candidatos no texto de entrada (dados observados) com um corpus de referência (dados esperados). Mais precisamente, o peso de um termo é o valor qui-quadrado que mede a divergência entre os dados observados e os esperados. Estes últimos são os dados obtidos a partir de um corpus de referência com um tamanho médio de 100M de tokens por língua, compilado pelo grupo ProLNat@GE, e que é composto por textos de vários géneros e domínios: jornalístico, técnico, literário, de redes sociais, etc. Finalmente, os termos são organizados em função do seu peso, de maior a menor, e o usuário escolhe os N mais relevantes em função do tamanho do texto e das necessidades de análise.

Termos multipalavra

Os termos multipalavra são expressões relevantes codificadas como unidades plurilexicais que instanciam padrões específicos de etiquetas morfossintáticas. Por exemplo, *língua natural*, *processamento da língua*, *tecnologias da língua* ou *analisador sintático* podem ser unidades multipalavra relevantes dentro de um texto de domínio científico focado em questões de PLN. Como no caso dos termos básicos, o processo de extração de multipalavras divide-se em duas fases: seleção de candidatos e ordenação dos mesmos por relevância. Porém, tanto a seleção de candidatos como a ordenação realizam-se mediante estratégias diferentes às utilizadas para a extração dos termos básicos.

Para a primeira fase utilizamos um conjunto de padrões de etiquetas (tabela 2) para identificar todas aquelas expressões plurilexicais que os instanciam (os artigos e determinantes das expressões não se tomam em conta na instanciação). O conjunto foi desenhado para a identi-

$nome - adj$	$adj - nome$
$nome - nome$	$nome - prep - nome$
$nome - prep - adj - nome$	$nome - prep - nome - adj$
$adj - nome - prep - nome$	$nome - adj - prep - nome$
$adj - nome - prep - nome - adj$	$nome-adj-prep-nome-adj$
$adj - nome - prep - adj - nome$	$nome-adj-prep-adj-nome$

Tabela 2: Conjunto de padrões de etiquetas utilizado para a identificação de candidatos a termos multipalavra (*adj* é adjetivo e *prep* é preposição).

peso	**multipalavra**	**padrão de etiquetas**
9,95	dación en pago	nome-prep-nome
7,94	viviendas vacías	nome-adj
7,27	renta básica	nome-adj
5,24	iniciativas legislativas	nome-adj
2,99	reuniones de representantes	nome-prep-nome

Tabela 3: As cinco multipalavras mais relevantes (*unithood*) extraídas do programa eleitoral do partido político espanhol *Podemos* para as eleições do 20D/2015.

ficação de multipalavras nas quatro línguas tratadas. Este método é semelhante ao descrito noutros trabalhos sobre extração terminológica (Vivaldi & Rodríguez, 2001; Sánchez & Moreno, 2006). Os padrões foram selecionados a partir da revisão manual de uma lista de n-gramas de etiquetas ordenadas por frequência em corpora de diferentes línguas.

Na segunda fase, a ordenação por relevância, utilizamos uma estratégia diferente à empregada na ordenação por termos básicos. Enquanto estes se ordenam em função da noção de *termhood*, a relevância das expressões multipalavra define-se mediante o conceito de *unithood*. Esta noção faz referência à associação das sequências de palavras com unidades lexicais estáveis. Mais concretamente, *unithood* refere-se ao grau de força e coesão entre as unidades lexicais que constituem os sintagmas e colocações (Kageura & Umino, 1996). A *unithood* só se aplica, portanto, a unidades plurilexicais com alguma coesão interna e não a unidades monolexicais.

O grau de coesão, ou *unithood*, pode calcular-se com diferentes medidas de associação lexical. O módulo de LinguaKit permite escolher entre 5 medidas para ordenar os candidatos a multipalavra: (a) qui-quadrado, (b) função de verosimilhança (*loglikehood*), (c) informação mutua (*mi*), (d) probabilidade condicional simétrica (*scp*), e (e) simples co-ocorrência. As medidas de associação aplicam-se para verificar se os constituintes co-ocorrem num sintagma aleatoriamente ou por atração. Assim, os valores observados equivalem à frequência da expressão multipalavra no texto de entrada, e os valores esperados calculam-se a partir das frequências dos constituintes por separado.

É importante sublinhar que estas estratégias básicas de extração são de propósito geral pois não estão adaptadas a um domínio específico. São aplicáveis portanto a qualquer domínio. No entanto, para serem mais eficientes, precisavam de incluir novos sub-módulos que permitissem uma fácil adaptação a domínios de especialidade. Na atualidade, a extração só permite selecionar e identificar candidatos a termo em geral, e não unidades terminológicas de um domínio previamente identificado.

Como exemplo de utilização, as tabelas 3 e 4 mostram as expressões multipalavra mais relevantes (usando qui-quadrado como peso para a ordenação) extraídas de dous programas de partidos políticos, *Podemos* e o *Partido Popular*, para as eleições ao parlamento espanhol de 20 de dezembro de 2015. Assim, este exemplo mostra como o extrator permite identificar as prioridades programáticas dos partidos políticos com uma simples vista de olhos sobre os termos mais relevantes.

Mesmo se a eficiência da extração de termos não foi avaliada quantitativamente, podemos encontrar alguns elementos que demonstram a sua usabilidade desde um ponto de vista qualitativo. Por um lado, os dous extratores de termos (básicos e multipalavra) foram inseridos no módulo mais complexo de anotação e ligação semântica, o qual sim foi avaliado quantitativamente e comparado com outros sistemas de anotação. Por outro lado, estes módulos foram utilizados por utentes muito variados com dife-

peso	multipalavra	padrão de etiquetas
20,37	inversores extranjeros	nome-adj
11,44	creación de empleo	nome-prep-nome
9,75	competitividad de economía	nome-prep-nome
7,73	reducción de impuestos	nome-prep-nome
2,93	ciudadanos españoles	nome-adj

Tabela 4: As cinco multipalavras mais relevantes (*unithood*) extraídas do programa eleitoral do partido político espanhol *Partido Popular* para as eleições do 20D/2015.

rentes aplicações e objetivos, tais como análises dos programas de partidos políticos feitas por jornalistas.[13]

6 Conclusões e trabalho futuro

Este artigo apresentou LinguaKit, um pacote linguístico que permite os utilizadores ter um acesso fácil e unificado a módulos de análise linguística muito diversos.

O conjunto de ferramentas disponível, mesmo se amplo e variado, fica ainda longe de cobrir todos as necessidades dos profissionais e utilizadores da língua. A este respeito, como trabalho futuro pretendemos, por um lado, continuar a melhorar o desempenho de alguns dos módulos de análise, e por outro lado ampliar o número de módulos com sistemas de transcrição fonética e fonológica. Além disso, está prevista a adaptação dos módulos de análise morfossintática e sintática para a sua compatibilidade com as diretrizes de anotação das *dependências universais*.

Para além de novos módulos, o sistema pode enriquecer-se com funcionalidades simples mas úteis para linguistas e investigadores. Por exemplo, um buscador de contextos léxico-sintáticos que utilize o analisador sintático para permitir procurar que nomes funcionam como sujeitos de um verbo específico, adjetivos que modifiquem um dado nome, etc. Em relação às novas funcionalidades, será preciso identificar os principais objetivos dos utilizadores para tentar que o sistema cubra as suas necessidades.

Agradecimentos

Este trabalho foi realizado graças ao financiamento da *Ayuda da Fundación BBVA para Investigadores y Creadores Culturales*, do projeto TELEPARES (MINECO, ref:FFI2014-51978-C2-1-R), da *Consellería de Cultura, Educación e Ordenación Universitaria* (2016-2019, ED431G/08), do *European Regional Development Fund (ERDF)*, e de um contrato *Juan de la Cierva-formación*, com referência FJCI-2014-22853.

[13] http://www.galiciaconfidencial.com/noticia/27170-son-galiza-galicia-marea

Referências

Agerri, Rodrigo, Josu Bermudez & German Rigau. 2014. IXA pipeline: Efficient and ready to use multilingual NLP tools. Em *9th International Conference on Language Resources and Evaluation (LREC)*, 3823–3828.

Bird, Steven, Edward Loper & Ewan Klein. 2009. *Natural language processing with Python*. O'Reilly Media Inc.

Corro, Luciano Del & Rainer Gemulla. 2013. ClausIE: Clause-based open information extraction. Em *The World Wide Web Conference*, 355–366.

Etzioni, Oren, Anthony Fader, Janara Christensen, Stephen Soderland & Mausam. 2011. Open information extraction: the second generation. Em *International Joint Conference on Artificial Intelligence (IJCAI)*, 3–10.

Gamallo, Pablo. 2015. Dependency parsing with compression rules. Em *International Workshop on Parsing Technology (IWPT)*, 107–117.

Gamallo, Pablo & Marcos Garcia. 2011. A resource-based method for named entity extraction and classification. Em *Portuguese Conference on Artificial Intelligence (EPIA 2011)*, 610–623.

Gamallo, Pablo & Marcos Garcia. 2014. Citius: a naive-bayes strategy for sentiment analysis on English tweets. Em *8th International Workshop on Semantic Evaluation (SemEval)*, 171–175.

Gamallo, Pablo & Marcos Garcia. 2015. Multilingual open information extraction. Em *17th Portuguese Conference on Artificial Intelligence (EPIA)*, 711–722.

Gamallo, Pablo & Marcos Garcia. 2016. Entity linking with distributional semantics. Em *International Conference on the Computational Processing of the Portuguese Language (PROPOR)*, 177–188.

Gamallo, Pablo, Marcos Garcia & Santiago Fernández-Lanza. 2013a. TASS: a naive-bayes strategy for sentiment analysis on Spanish tweets. Em *Workshop on Sentiment Analysis (TASS@SEPLN)*, 126–132.

Gamallo, Pablo, Marcos Garcia, Isaac González, Marta Mu noz & Iria del Río. 2013b. Learning verb inflection using Cilenis conjugators. *The Eurocall Review* 21(1). 12–19.

Gamallo, Pablo, Marcos Garcia, Iria del Río & Isaac González López. 2015a. Avalingua: Natural language processing for automatic error detection. Em *Learner Corpora in Language Testing and Assessment*, vol. 70 Studies in Corpus Linguistics, 35–58. John Benjamins Publishing Company.

Gamallo, Pablo & Isaac González. 2011. A grammatical formalism based on patterns of part-of-speech tags. *International Journal of Corpus Linguistics* 16(1). 45–71.

Gamallo, Pablo, Juan Carlos Pichel, Marcos Garcia, José Manuel Abuín & Tomás Fernández-Pena. 2015b. Análisis morfosintáctico y clasificación de entidades nombradas en un entorno big data. *Procesamiento del Lenguaje Natural* 53. 17–24.

Garcia, Marcos. 2016. Incorporating lexico-semantic heuristics into coreference resolution sieves for named entity recognition at document-level. Em *10th edition of the Language Resources and Evaluation Conference (LREC)*, 3357–3361.

Garcia, Marcos & Pablo Gamallo. 2010. Análise morfossintáctica para português europeu e galego: Problemas, soluções e avaliação. *Linguamática* 2(2). 59–67.

Garcia, Marcos & Pablo Gamallo. 2014. An entity-centric coreference resolution system for person entities with rich linguistic information. Em *25th International Conference on Computational Linguistics: Technical Papers (COLING)*, 741–752.

Garcia, Marcos & Pablo Gamallo. 2015. Yet another suite of multilingual NLP tools. Em *Symposium on Languages, Applications and Technologies (SLATE)*, 65–75.

Garcia, Marcos, Isaac González & Iria del Río. 2012. Identificação e classificação de entidades mencionadas em Galego. *Estudos de Linguística Galega* 4. 13–25.

Kageura, Kyo & Bin Umino. 1996. Methods of automatic term recognition: A review. *Terminology* 3(1). 259–289.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard & David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. Em *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

Mendes, Pablo N., Max Jakob, Andrés García-Silva & Christian Bizer. 2011. DBpedia spotlight: Shedding light on the web of documents. Em *7th International Conference on Semantic Systems*, 1–8.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov & Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering* 13(2). 115–135.

Padró, Lluís. 2011. Analizadores multilingües en FreeLing. *Linguamática* 3(2). 13–20.

Real Academia Galega e Instituto da Lingua Galega. 2004. *Normas ortográficas e morfolóxicas do idioma galego*. Editorial Galaxia.

Sánchez, David & Antonio Moreno. 2006. A methodology for knowledge acquisition from the web. *Journal of Knowledge-Based and Intelligent Engineering Systems* 10(6). 453–475.

Straka, Milan, Jan Hajič & Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. Em *10th International Conference on Language Resources and Evaluation (LREC)*, 4290–4297.

Vivaldi, Jordi & Horacio Rodríguez. 2001. Improving term extraction by combining different techniques. *Terminology* 7(1). 31–47.