DeepString - Syntax Deep Explorer

Integrating multi-corpora support into a corpus analysis tool

João Pedro dos Santos Trindade

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Professor Doutor Nuno João Neves Mamede
Professor Jorge Manuel Evangelista Baptista

Examination Committee

Chairperson: Professor Francisco António Chaves Saraiva de Melo
Supervisor: Professor Nuno João Neves Mamede
Member of the Committee: Professora Maria Luísa Torres Ribeiro Marques da Silva Coheur

June 2020

Acknowledgements

I would like to start this dissertation project by thanking my family for their unconditional support throughout all of my life as a student. I want to thank my girlfriend Teresa for all the times she motivated me on this journey, for never giving up on me and for her help with everything related to my dissertation project. I also want thank my friends for providing me an outlet to relax and for the support that they gave me. Lastly I want to thank Prof. Nuno Mamede and Prof. Jorge Baptista for their guidance and availability to help in this dissertation project.
This project could not be accomplished without their help and support.

Lisbon, 10th of May, 2020
João Pedro dos Santos Trindade

Resumo

Com o avanço da tecnologia, os linguistas passaram a dispor de diversas ferramentas para os ajudar nos seus estudos, entre as quais, ferramentas de análise de *corpora*. Estas ferramentas ajudam a determinar melhor como são usadas certas palavras em contexto, calculando diversas medidas de associação entre palavras coocorrentes e apresentando esses resultados sob a forma de um perfil distribucional.

Uma destas ferramentas é o Syntax Deep Explorer, que recebe como input um corpus previamente processado pela STRING e permite realizar diversas pesquisas com a informação sintática com que o corpus foi anotado. A STRING é uma cadeia de Processamento de Linguagem Natural desenvolvida pelo Laboratório de Tecnologias da Língua Humana no INESC-ID que realiza todas as tarefas básicas de processamento de texto em língua natural, incluindo a análise sintática e a extração das relações de dependência sintática entre constituintes.

O Syntax Deep Explorer diferencia-se de outras ferramentas de análise de *corpora* atuais por permitir pesquisas com base nessas dependências sintáticas (sujeito, complemento, etc.) e por oferecer um leque mais diversificado de medidas de associação do que o de outras ferramentas atuais.

Este projeto engloba algumas melhorias e novas funcionalidades implementadas no Syntax Deep Explorer. As principais funcionalidades que foram desenvolvidas são: a comparação entre os perfis distribucionais de 2 palavras no mesmo corpus e a comparação entre perfis da mesma palavra em 2 corpora distintos; a apresentação de exemplos, com destaque das palavras-alvo, bem como a melhoria do formato de apresentação dos perfis lexicais; e o suporte *multicorpora*. Dois novos *corpora* foram constituídos para suportar estas novas funcionalidades: um *corpus* de textos jornalísticos desportivos (*Desportivo*) e outro com as atas de sessões da Assembleia da República (*Parlamento*).

Palavras-Chave

Processamento de Língua Natural
Coocorrência
Medidas de associação
STRING
Linguística de corpus

Abstract

With the evolution of technology, linguists now have numerous tools to aid them in their studies, including, several *corpora* analysis tools. These tools help in determining how words are used in context within a *corpus*. Besides concordances, some tools can also automatically calculate several association measures between co-occurrent words and display these results in the form of a distributional profile.

One such tool is the *Syntax Deep Explorer*. This tool receives as input a *corpus* that has been previously processed by STRING and allows the user to execute several searches based on the syntactic information annotated on the *corpus*. STRING is a Natural Language Processing Chain for the Portuguese language developed by the Human Languages Technologies Laboratory at INESC-ID Lisboa. It performs all the basic tasks in natural language processing, including, syntactic analysis and the extraction of syntactic dependencies between constituents (dependency parsing). Syntax Deep Explorer distinguishes itself from other *corpora* analysis tools by allowing searches based on these syntactic dependencies (subject, direct object, etc.) and by offering a more diversified array of association measures, when compared to other current tools.

This project covers some improvements and some new features implemented to Syntax Deep Explorer: (i) the comparison between the distributional profiles of 2 words within the same *corpus* and (ii) the comparison of the distributional profiles of the same word in 2 distinct *corpora*; (iii) the presentation of examples, with the highlighting of target words, as well as, the improvement of the format in which distributional profiles are presented; and *multi-corpora* support. Two new *corpora* were constituted to support these new functionalities: a *corpus* from sports newspapers texts (*Desportivo*) and another with the minutes from the Portuguese Parliament (*Parlamento*).

Keywords

Natural Language Processing
Co-occurrence
Association measures
STRING
Corpus linguistics

Table of Contents

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface.

CQL Contextual Query Language.

INESC-ID Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento.

LexMan Lexical Morfological Analizer.

MARv Morphosyntactic Ambiguity Resolver.

PMI Pointwise Mutual Information.

POS Part-of-Speech.

RuDriCo Rule-Driven Converter.

SQL Structured Query Language.

STRING Statistical and Rule-Based Natural Language Processing Chain.

XIP Xerox Incremental Parser.

XML Extensible Markup Language.

Chapter 1

Introduction

A *corpus* analysis tool is usually built upon examining and extracting information from words in a *corpus*, relate them in some relevant way and present the user with some insightful knowledge about their relation. Using *corpus* analysis tools allows one to produce more apt examples and empirically better motivated descriptions of language use and strutcture in the sense that they use actual information from *corpora* to establish how words are used and discover certain patterns in the use of a language, in a given *corpus*. The *Syntax Deep Explorer* (for brevity reasons, will be referred to as *Explorer*) is one such tool that was developed in order to better analyze *corpora* processed by the Statistical and Rule-Based Natural Language Processing Chain (STRING). The Explorer differs from other *corpus* analysis tools by having multiple association measures for the user to choose from, and applying them to syntactic dependency relations between words.
This introduction is divided into three sections:

- Section 1.1 describes the objectives of this dissertation;
- Section 1.2 gives a brief explanation of STRING;
- Section 1.3 describes the original version of the Explorer.

1.1 Goal

The main goal of this dissertation is the improvement of the Explorer system, providing the user more information about the lemmas being searched for, and increasing the number of available relevant features. To accomplish these goals this project focused on:

- Adding support for multiple *corpora*, simplifying the task of adding information to the Explorer database, and allowing the user to search for lemmas in different *corpora*;
- Adding the option to compare two different lemmas in the same *corpus*;
- Adding the option to compare the same lemma in two different *corpora*;
- Highlight target words in example sentences;

- Adding more information to each search;
- Improving the user experience by correcting visual inconsistencies and helping the user in finding relevant information.

1.2 STRING

STRING[1] is a Natural Language Processing Chain for Portuguese, developed at Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID) Lisboa [11]. It has a modular structure, comprised of four main components.

- Lexical Morfological Analizer (LexMan)
  LexMan [15] receives a plain text from the *corpus* and performs 3 tasks:
  1. **Tokenization.** LexMan divides the text into tokens. These are words, multiword units, numbers, punctuation, symbols and other textual units.
  2. **Token annotation.** Token annotation is the main function of LexMan. LexMan gives the token an annotation with 11 characters. Each character represents different information that is associated to the token. For example, the annotation for the token *professor* 'professor' is "Nc...smn.==" meaning that the token is a common noun (c), masculine (m), singular(s) and with no affixes (n).
  3. **Sentence splitting.** This module splits the text into sentences. This is done by correctly identifying the sentences' boundaries.

- Rule-Driven Converter (RuDriCo)
  RuDriCo [7] is a rule-based morphological disambiguator that can also change the segmentation of the input. For example, it may receive the contracted form *no* 'in_the' and separate it into its constitutive elements, the preposition *em* 'in' and the define article *o* 'the' . Besides, RuDriCo2 also uses rule-based knowledge to morphologically disambiguate certain tokens. For instance, contextual rules are used to disambiguate the verbal participle *partido* 'broken' and the noun *partido*'political party'.

- Morphosyntactic Ambiguity Resolver (MARv)
  MARv [14] is a statistical Part-of-Speech (POS) disambiguator based on Markovian models, which uses the Viterbi algorithm to choose the most likely POS tag for each word. The language model used is based in trigrams, which encode contextual information; and unigrams, which encode lexical information.

  The probability of each tag is calculated based on a previously tagged, training *corpus* with approximately 250,000 words. This module offers a precision of over 97% when disambiguating words [11].

[1] https://string.hlt.inesc-id.pt (last visited in 10 of April 2020)

- Xerox Incremental Parser (XIP)

  This module analyzes the morphologically tagged and disambiguated text as input and syntactically analyzes it using *Xerox Incremental Parser (XIP)* [1]. XIP parses the text dividing it into elementary phrases, known as *chunks* such as noun phrase (`NP`) or prepositional phrase (`PP`); and extracts syntactic relations (dependencies) between these chunks such as subject (`SUBJ`) or direct complement (`CDIR`). The full list of the dependencies used in the Explorer is available on Section 1.3, along with an example for each one. These dependencies are derived from a set of pre-programmed, manually crafted, syntactic rules, which constitute the Portuguese grammar of XIP. The rule-based grammar is divided into modules, which are ordered by their depth level. Rules with a lower depth level are applied first. With this method, it is possible to build highly detailed and rich lexical and dependency-based descriptions.

1.3 Syntax Deep Explorer

Syntax Deep Explorer[2] is a tool initially developed by José Pereira [6] at INESC-ID Lisboa in 2015. Its objective is to provide easy access for an analysis of co-occurrence patterns between words in order to understand how they are used in sentences taken from a given *corpus*.
As it was previously mentioned, the Explorer collects the output of STRING. This is done by saving the dependencies originated from the XIP module into a SQLite database and calculating several association measures for each word co-occurrence.

A previously developed XIP Application Programming Interface (API) [5] is used to access the output of XIP. This API transforms the Extensible Markup Language (XML) output of XIP into Java structures. The structures implemented in the XIP API are the following:

- **XIPNode** is the basic structure of the chunk tree. A XIPNode can be a leaf or a branch on the chunk tree. If it is a branch, it contains children node, also represented as XIPNodes. A XIPNode also can contain a group of Features and Dependencies;

- **Token** represents the leaf XIPNode of a chunk tree. It presents information regarding the token that has been analysed by STRING, such as the word and the lemma;

- **Feature** contains the properties of the XIPNodes or the properties of a Dependency;

- **Dependency** is a structure that contains information about the word co-occurrence produced in XIP;

- **XIPDocument** is the structural representation of a chunk tree, containing both XIPNodes and their respective Dependencies.

These dependencies form the core of this project: what is shown to the user are the most prominent word co-occurrences in certain dependencies according to the selected association measure. Each dependency

[2] https://string.l2f.inesc-id.pt/demo/deepExplorer (last accessed in 27 of April 2020).

relates two arguments, the *modified* or *governed* element, and the *modifying* or *governor* element. The XIP dependencies [2] used by the Explorer are:

- `SUBJ`: Associating a subject to a verb

  ***O cão corre.* ‘The dog runs.’**

  `SUBJ(corre,cão)`;

- `CDIR`: Associating a direct (accusative) complement to a verb;

  ***Eu como o almoço.* ‘I eat lunch.’**

  `CDIR(como,almoço)`;

- `CINDIR`: Associating an indirect (dative) complement to a verb;

  ***Eu respondi ao Ricardo.* ‘I responded to Ricardo.’**

  `CINDIR(respondi,Ricardo)`;

- `MOD`: Associating a word or expression to its modifier;

  ***O caderno azul.* ‘The blue notebook.’**

  `MOD(caderno,azul)`;

- `COMPL`: Associating a predicative element (e.g. a verb) to its essential complement;

  ***O Pedro foi convencido pelo João.* ‘Pedro was persuaded by João.’**

  `COMPL(convencido,João)`;

- `QUANTD`: Associating a noun to its quantifier;

  ***A Teresa fez 7 sobremesas.* ‘Teresa made 7 desserts.’**

  `QUANTD(sobremesas,7)`;

- `CLASSD`: Associating a noun to its nominal classifier.

  ***O Diogo gosta deste tipo de comida.* ‘Diogo likes this kind of food.’**

  `CLASSD(comida,tipo)`;

To determine in which order the elements appear in the *corpus*, a property may be added to the end of each `MOD` dependency. If that property is `PRE` then the second element of the dependency appears on the text *before* the first element. For example, `MOD_PRE(homem,grande)` indicates that the word *grande* ‘big’ appears on the *corpus* before the word *homem* ‘man’ , as in the sentence: *Ele é um grande homem* ‘He is a great man.’ , whose meaning would be different if the adjective appeared after *after* the noun *Ele é um homem grande.* ‘He is a big man.’ .

`POST` indicates that the second word of the dependency appears *after* the first one, as it would be the case in the last example.

In addition to the `PRE/POST` properties added by STRING, this system also adds the POS of both words in the dependency. This addition is helpful when querying the database for specific dependencies. For example, `MOD_PRE_ADJ_NOUN` represents a dependency-property pattern where an adjective modifies a noun and the adjective appears before the noun in the sentence. In the case where the order of the co-occurrent words is unknown, the system adds a property called `SEM_PROP`, for example `SUBJ_SEM_PROP`.

Figure 1.1 is the entity-relationship model of the database used by this tool. It consists of 8 main tables, namely:

- **Corpus**, where information regarding the available *corpora* is stored (in this initial version of the Explorer it only contains information regarding CETEMPúblico);
- **Co-occurrence**, where the co-occurrences are stored, along with the corresponding measures and the information about the *corpus* from where they were extracted;
- **Dependency**, which contains the available types of XIP dependencies;
- **Exemplifies**, which connects co-occurrences to the name of the file and *corpus* from where they where extracted, and the number of the exact sentence (used to retrieve the example sentences);
- **Sentence**, where the sentences are stored and associated with the name of the files whence they came and the number of the sentence in that file;
- **Word**, where the lemma id is matched to the actual lemma and its corresponding POS;
- **Belongs to**, which contains the frequency of co-occurrences of every lemma;
- **Property**, where the properties of each co-occurrence (`PRE/POST`, and the POS of the lemmas of the co-occurrence) are matched with the corresponding XIP dependencies.

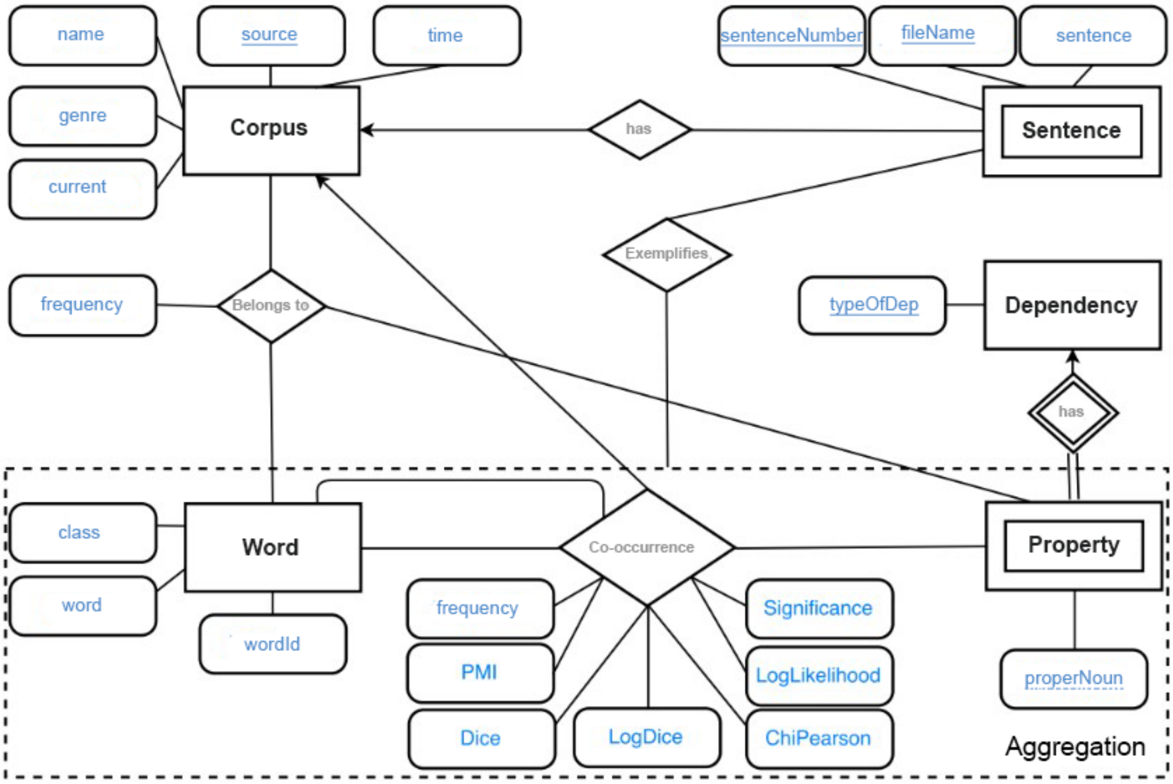
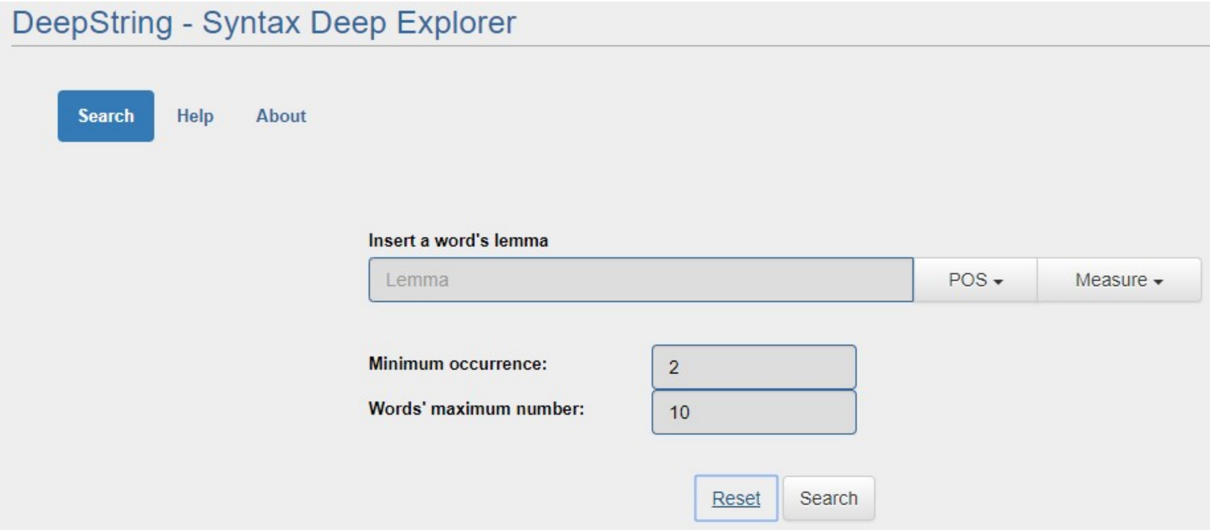


Figure 1.1: Syntax Deep Explorer's entity-relationship model [6].

The Explorer's main menu consists of a search prompt, which can be seen in Figure 1.2. The five distinct available options are:

- **The lemma field**, where the user types the lemma of the word to search;
- **The part-of-speech selector**, a dropdown menu, where the user can select the word class of the lemma. At the moment, the Explorer only supports the search for the following main POS:
  - Noun
  - Verb
  - Adjective
  - Adverb;
- **The association measure selector**, a dropdown menu, where the user can select which association measure is to be used to calculate the results of the query. The full description and rationale behind the choice of each measure is explained in [6], along with source references. The available association measures are:
  - Dice
  - LogDice
  - Pointwise Mutual Information (PMI)
  - ChiPearson
  - LogLikelihood
  - Significance
  - Frequency;
- **The minimum occurrence field**, where the user can select the minimum number of occurrences required for a co-occurrence to appear in the results. The default value is set to 2 occurrences;
- **The word's maximum number field**, which determines the maximum number of results that will appear on each section on the co-occurrence screen. The default value is set to 10.



DeepString - Syntax Deep Explorer

Insert a word's lemma

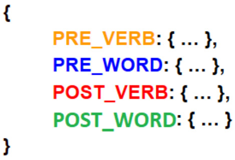
Minimum occurrence: 2

Words' maximum number: 10

Figure 1.2: Syntax Deep Explorer's main menu (from [6]).

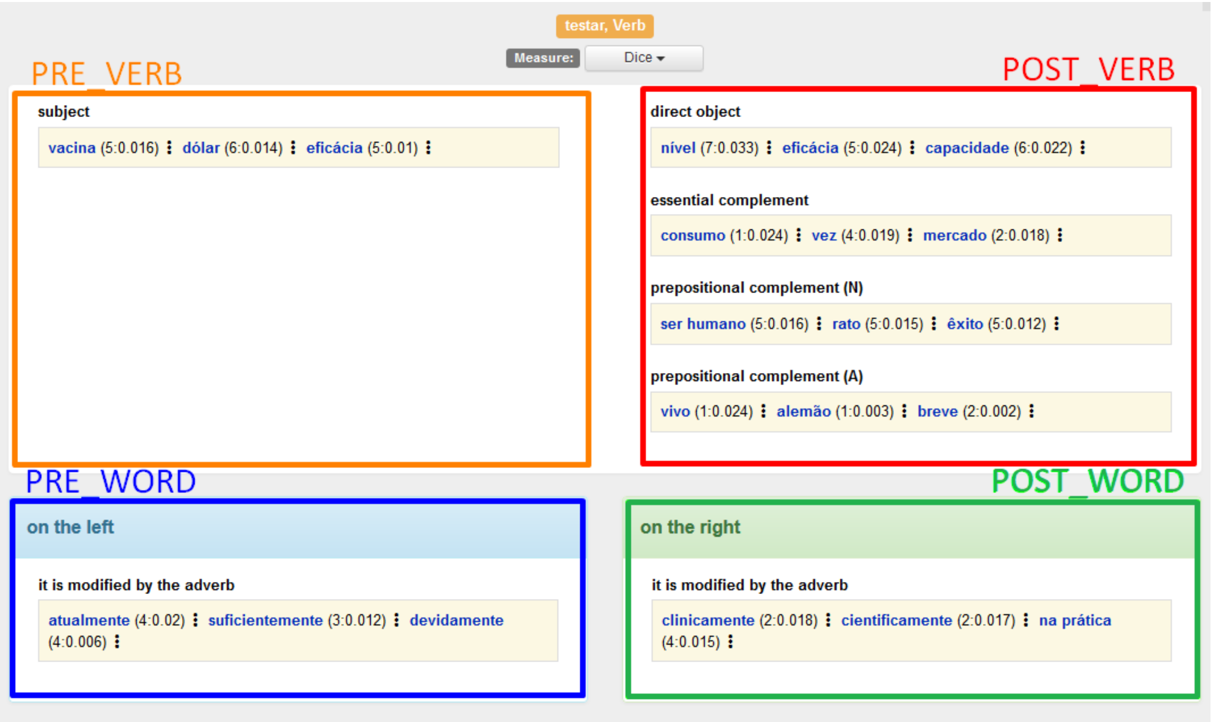
After choosing the desired options and clicking the **Search** button, the system displays the co-occurrences of the chosen lemma separated into two columns, “on the left” and “on the right”. Co-occurrence results are ordered by the desired metric. Co-occurrences with a higher score in the chosen metric will appear first than co-occurrences with a lower score.

Figure 1.3 represents the structure of the object that resulted from an Explorer search. This object can have up to four attributes. These attributes are named `PRE_VERB`, `PRE_WORD`, `POST_VERB` and `POST_WORD`, and they divide the extracted information, determining where it will be placed on the screen. Figure 1.4 shows how this information is displayed. Information contained on the `PRE_VERB` attribute will appear on the left part of the verb section, information contained on the `POST_VERB` attribute will appear on the right part of the verb section, information contained on the `PRE_WORD` attribute will appear on the left panel and finally information contained on the `POST_WORD` attribute will appear on the right panel.



```
{
    PRE_VERB: { ... },
    PRE_WORD: { ... },
    POST_VERB: { ... },
    POST_WORD: { ... }
}
```

Figure 1.3: Data structure retrieved from the database after a search.



subject

vacina (5:0.016) ⋮ dólar (6:0.014) ⋮ eficácia (5:0.01) ⋮

direct object

nível (7:0.033) ⋮ eficácia (5:0.024) ⋮ capacidade (6:0.022) ⋮

essential complement

consumo (1:0.024) ⋮ vez (4:0.019) ⋮ mercado (2:0.018) ⋮

prepositional complement (N)

ser humano (5:0.016) ⋮ rato (5:0.015) ⋮ êxito (5:0.012) ⋮

prepositional complement (A)

vivo (1:0.024) ⋮ alemão (1:0.003) ⋮ breve (2:0.002) ⋮

on the left

it is modified by the adverb

atualmente (4:0.02) ⋮ suficientemente (3:0.012) ⋮ devidamente (4:0.006) ⋮

on the right

it is modified by the adverb

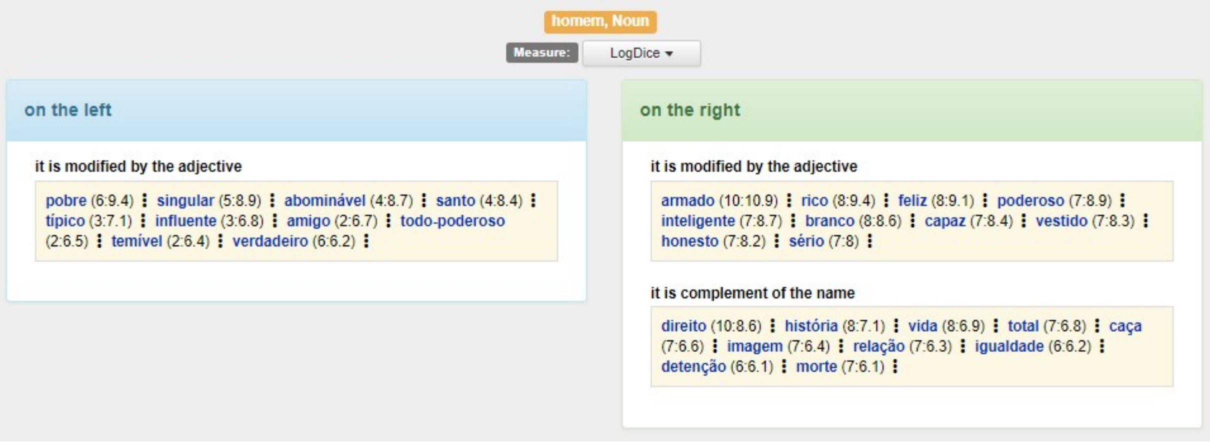
clinicamente (2:0.018) ⋮ cientificamente (2:0.017) ⋮ na prática (4:0.015) ⋮

Figure 1.4: Data structure retrieved from the database after a search for the verb *testar* ‘to test’ .

Next to each co-occurrence result, two values are provided, separated by a colon. The first value represents the binary logarithm of the frequency of the co-occurrence, while the second value is the score of the co-occurrence selected association measure.

The displayed information differs according to the chosen word class. For example, in Figure 1.5, it is possible to analyze the adjectives that modify the lemma *homem* 'man' and the words of which it depends as a complement. On the other hand, if the chosen lemma was an adjective, the Explorer would have displayed the adverbs that modify the chosen lemma and which nouns that adjective modifies.

Some of the dependency descriptions used in the following examples contain terms that are incorrect (name instead of *noun*) and some terminology that is not the most accurate in the scope of this project (object instead of *complement*). These terms were corrected in this dissertation and the new ones are displayed in Subsection 3.2.2. However, in the following, these terms, though inadequate, were kept, for consistency.



on the left

it is modified by the adjective

pobre (6:9.4) ⋮ singular (5:8.9) ⋮ abominável (4:8.7) ⋮ santo (4:8.4) ⋮ típico (3:7.1) ⋮ influente (3:6.8) ⋮ amigo (2:6.7) ⋮ todo-poderoso (2:6.5) ⋮ temível (2:6.4) ⋮ verdadeiro (6:6.2) ⋮

on the right

it is modified by the adjective

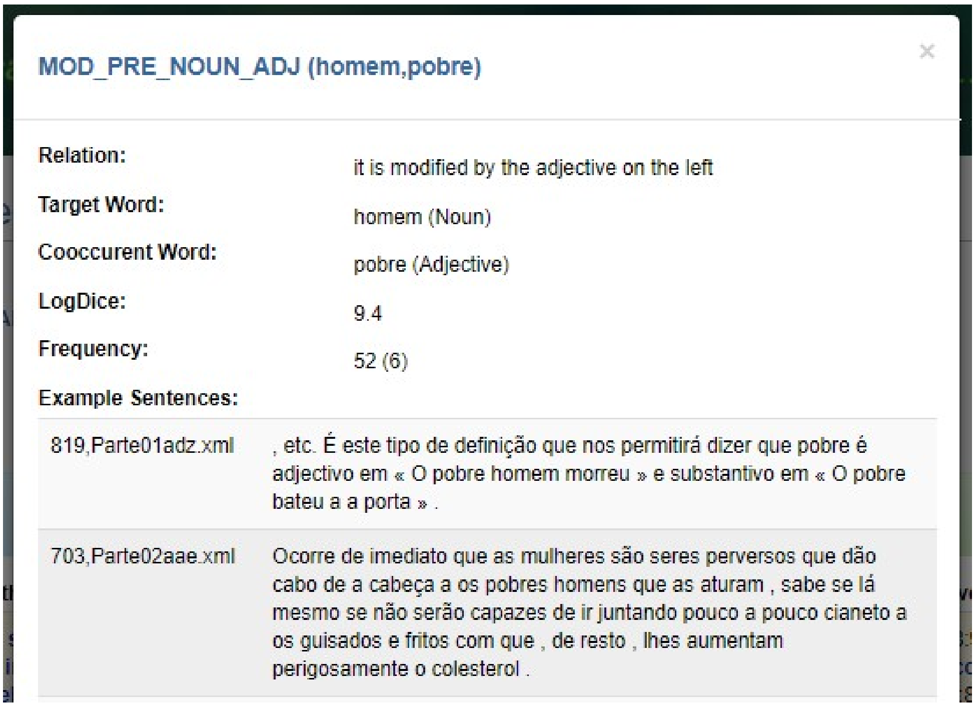
armado (10:10.9) ⋮ rico (8:9.4) ⋮ feliz (8:9.1) ⋮ poderoso (7:8.9) ⋮ inteligente (7:8.7) ⋮ branco (8:8.6) ⋮ capaz (7:8.4) ⋮ vestido (7:8.3) ⋮ honesto (7:8.2) ⋮ sério (7:8) ⋮

it is complement of the name

direito (10:8.6) ⋮ história (8:7.1) ⋮ vida (8:6.9) ⋮ total (7:6.8) ⋮ caça (7:6.6) ⋮ imagem (7:6.4) ⋮ relação (7:6.3) ⋮ igualdade (6:6.2) ⋮ detenção (6:6.1) ⋮ morte (7:6.1) ⋮

Figure 1.5: Co-occurrence profile (lexgram) of the noun *homem* 'man' using LogDice, with default values (from [6]).

After reaching the screen shown on Figure 1.5, the user can then click, for example on *pobre* 'poor' , situated on the left column, to get an in-depth look at the specific relation between these two lemmas. Clicking on that lemma will reveal a screen similar to the one on Figure 1.6, which shows some snippets of text from which the system extracted this specific relation.



MOD_PRE_NOUN_ADJ (homem,pobre)

Relation:	it is modified by the adjective on the left
Target Word:	homem (Noun)
Cooccurent Word:	pobre (Adjective)
LogDice:	9.4
Frequency:	52 (6)

Example Sentences:

819,Parte01adz.xml	, etc. É este tipo de definição que nos permitirá dizer que pobre é adjectivo em « O pobre homem morreu » e substantivo em « O pobre bateu a a porta » .
703,Parte02aae.xml	Ocorre de imediato que as mulheres são seres perversos que dão cabo de a cabeça a os pobres homens que as aturam , sabe se lá mesmo se não serão capazes de ir juntando pouco a pouco cianeto a os guisados e fritos com que , de resto , lhes aumentam perigosamente o colesterol .

Figure 1.6: In-depth view of the co-occurrence profile of *homem* 'man' , when modified on the left by the adjective *pobre* 'poor' (from [6]).

The example shown in Figure 1.5 did not contain all the dependencies that can be associated to a noun. The full list of dependencies will now be displayed and exemplified. For clarity, this information will be presented in the following manner:

- PRE_WORD (A keyword that indicates where a dependency would be shown on screen.)
  - is modified by the adjective (A sentence like this one indicates the dependency that is being shown.)

    ***Ele tem um bom carro.* 'He has a good car.'** (For this presentation, an example sentence is pprovided with its translation on the right.)

    MOD_PRE_NOUN_ADJ(carro,bom) (This indicates the dependency that has been searched in the Explorer database)

All the following examples were obtained using the LogDice metric with a minimum frequency of 2 and a maximum number of co-occurrences to display set to 3, in order to save some space.

Noun

Figure 1.5, displayed above, already represents an example of an Explorer search for the noun *homem* 'man' . However, not all the relevant dependencies have been shown. The full list of dependencies that can be shown for the noun result screen is the following:

- PRE_WORD

- is modified by the adjective

  ***Ele tem um bom carro.* 'He has a good car.'**

  `MOD_PRE_NOUN_ADJ(carro,bom)``;`
- is modified by the quantifier [*sic*: determined]

  ***Ela tem dois cães.* 'She has two dogs.'**

  `QUANTD_NOUN_NOUN(cão,dois)``;`
- is modified by the nominal classifier [*sic*: determined]

  ***Eu gosto deste tipo de fruta.* 'I like these kinds of fruit.'**

  `CLASSD_NOUN_NOUN(fruta,tipo)``;`

- `POST_WORD`

  - is modified by the adjective

    ***Ela faz pratos chineses.* 'She cooks Chinese dishes.'**

    `MOD_POST_NOUN_ADJ(pratos,chineses)``;`
  - it is complement of the name [*sic*: noun]

    ***Ela é uma advogada de defesa.* 'She is a defense lawyer.'**

    `MOD_POST_NOUN_NOUN(advogada,defesa)``;`

Verb

Figure 1.7 represents an example of a search for the verb *dizer* 'to say' .

In this Figure, there are two entries for prepositional complement. Prepositional complement (N) represents the dependency of the target verb with a noun (N), while prepositional complement (A) is when an adjective (A) is treated as a noun in the context of the sentence and can appear as a complement of a verb. This example contains all of the verb dependencies that can possibly appear when a verb is queried. The list below displays these dependencies in text form:

- `PRE_VERB`

  - subject

    ***Ele chora.* 'He cries.'**

    `SUBJ_SEM_PROP(chorar,Ele)``;`

- `POST_VERB`

  - direct complement

    ***O Adolfo escala a parede.* 'Adolfo climbs the wall.'**

    `CDIR_SEM_PROP(escalar,parede)``;`
  - indirect complement

    ***Ele conta um segredo ao Tiago.* 'He tells Tiago a secret.'**

    `CINDIR_SEM_PROP(contar,Tiago)``;`

  - essential complement

    Está a ser estudado por cientistas. **'[It] is being studied by scientists.'**

    `COMPL_SEM_PROP(estudar,cientistas);`

  - prepositional complement (N)

    O Pedro gosta de chocolates. **'Pedro likes chocolates.'**

    `MOD_VERB_NOUN(gostar,chocolates);`

  - prepositional complement (A)

    É um dos melhores do mundo **'It's one of the best in the world'**

    `MOD_VERB_ADJ(ser,melhor);`

- PRE_WORD

  - it is modified by the adverb

    Ela nunca pediu nada. **'She never asked for anything.'**

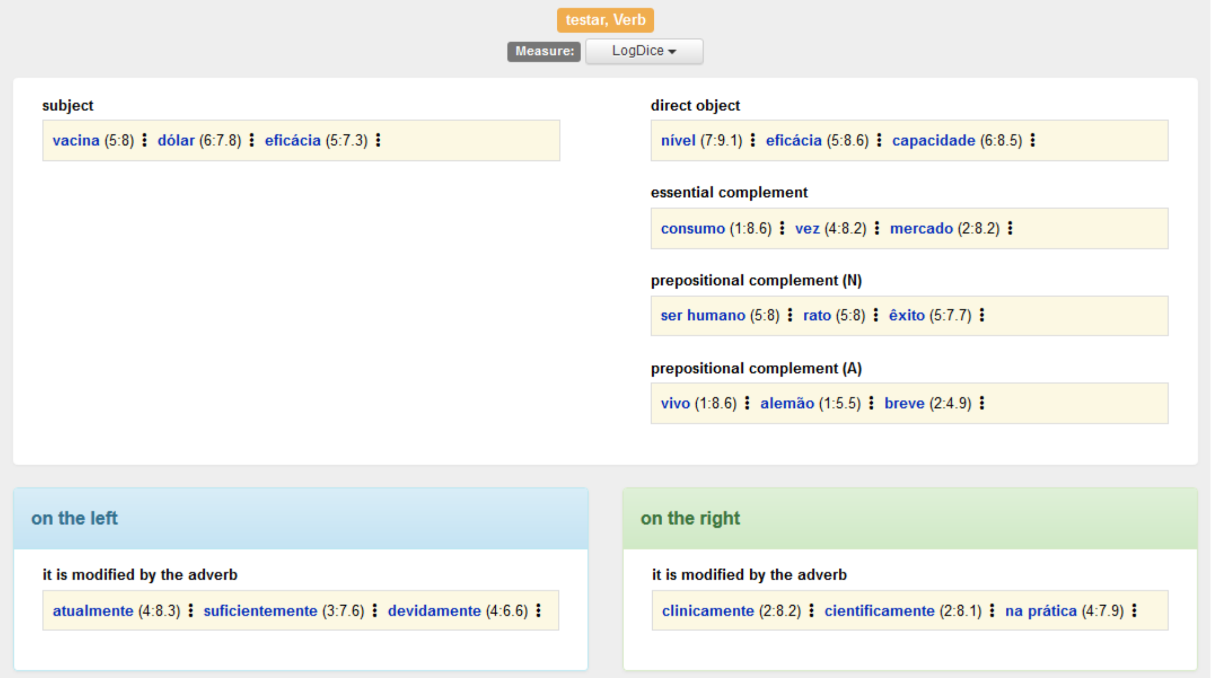
    MOD_ `PRE_VERB_ADV(pedir,nunca);`

- POST_WORD

  - it is modified by the adverb

    Ele come calmamente. **'He eats calmly.'**

    MOD_ `POST_VERB_ADV(comer,calmamente);`



testar, Verb

Measure: LogDice

subject

vacina (5:8) ⋮ dólar (6:7.8) ⋮ eficácia (5:7.3) ⋮

direct object

nível (7:9.1) ⋮ eficácia (5:8.6) ⋮ capacidade (6:8.5) ⋮

essential complement

consumo (1:8.6) ⋮ vez (4:8.2) ⋮ mercado (2:8.2) ⋮

prepositional complement (N)

ser humano (5:8) ⋮ rato (5:8) ⋮ êxito (5:7.7) ⋮

prepositional complement (A)

vivo (1:8.6) ⋮ alemão (1:5.5) ⋮ breve (2:4.9) ⋮

on the left

it is modified by the adverb

atualmente (4:8.3) ⋮ suficientemente (3:7.6) ⋮ devidamente (4:6.6) ⋮

on the right

it is modified by the adverb

clinicamente (2:8.2) ⋮ cientificamente (2:8.1) ⋮ na prática (4:7.9) ⋮

Figure 1.7: Co-occurrence profile of the verb *testar* 'to test' using LogDice, with a maximum of three results (from [6]).

Adjective

Figure 1.8 serves as an example of a search for the adjective *grande* 'big' . The list of possible dependencies on the adjective result screen is:

- PRE_WORD
  - is modified by the adverb

    O Pedro gosta mesmo muito de chocolates. **'Pedro really likes chocolates a lot.'**

    `MOD_PRE_ADJ_ADV(muito,mesmo)``;`
  - modifies the name [*sic*: noun]

    Tenho boas notícias. **'I have good news.'**

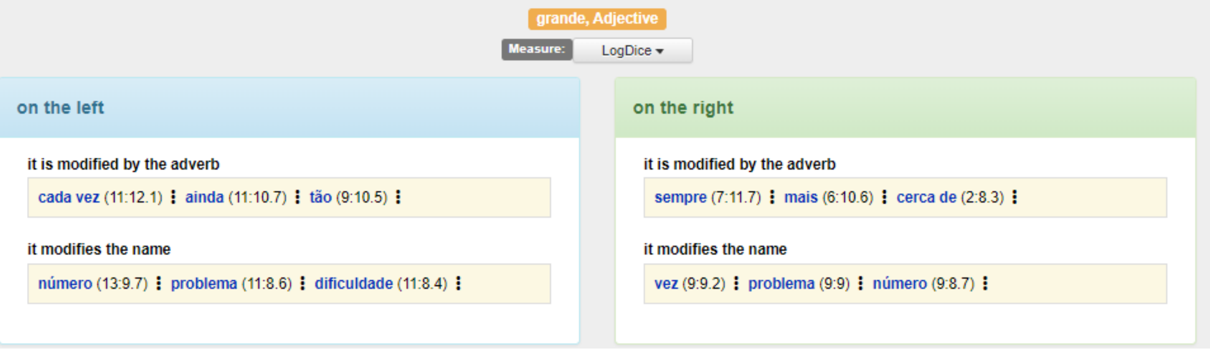
    `MOD_PRE_NOUN_ADJ(bom,notícia)``;`
- POST_WORD
  - is modified by the adverb

    Ele é forte de mais. **'He is too strong.'**

    `MOD_POST_ADJ_ADV(forte,mais)``;`
  - modifies the name [*sic*: noun]

    Ele é um rapaz inteligente. **'He is a smart boy.'**

    `MOD_POST_NOUN_ADV(rapaz,inteligente)``;`



grande, Adjective

Measure: LogDice ▾

on the left

it is modified by the adverb

cada vez (11:12.1) ⋮ ainda (11:10.7) ⋮ tão (9:10.5) ⋮

it modifies the name

número (13:9.7) ⋮ problema (11:8.6) ⋮ dificuldade (11:8.4) ⋮

on the right

it is modified by the adverb

sempre (7:11.7) ⋮ mais (6:10.6) ⋮ cerca de (2:8.3) ⋮

it modifies the name

vez (9:9.2) ⋮ problema (9:9) ⋮ número (9:8.7) ⋮

Figure 1.8: Co-occurrence of the adjective *grande* 'big' using LogDice, with a maximum of three results (from [6]).

Adverb

At last, the remaining POS supported by the Explorer is the adverb. Since adverbs can also modify whole sentences, the adverb result screen has an additional field in the bottom, where the number of modified sentences is shown. Figure 1.9 represents an example of a search of the lemma *sempre* 'always' and the list below depicts the complete list of possible dependencies that can appear on this screen:

- PRE_WORD
  - is modified by the adverb

    Ele lê muito bem. **'He reads very well.'**

    `MOD_PRE_ADV_ADV(bem,muito)``;`

  - it modifies the adjective

    Estou sempre pronto. **'I'm always ready.'**

    `MOD_PRE_ADJ_ADV(pronto,sempre)``;`

  - it modifies the name [*sic*: noun]

    A verdade é que isso é muito bom. **'The truth is that this is really good.'**

    `MOD_PRE_FOCUS_NOUN_ADV(verdade,é que)`[3]`;`

  - it modifies the verb

    Ele gosta imenso de cães. **'He likes dogs very much.'**

    `MOD_PRE_VERB_ADV(gostar,imenso)``;`

- `POST_WORD`

  - is modified by the adverb

    Ela trabalha muito rapidamente. **'She works really fast.'**

    `MOD_PRE_ADV_ADV(rapidamente,muito)``;`

  - it modifies the adjective

    Ela é a melhor de sempre. **'She is the greatest of all time.'**

    `MOD_POST_ADJ_ADV(bom,sempre)``;`

  - it modifies the name

    Como é que ela aguenta? **'How can she stand that?'**

    `MOD_POST_FOCUS_NOUN_ADV(ele,é que)``;`

  - it modifies the verb

    Eu sempre disse isso. **'I always said that.'**

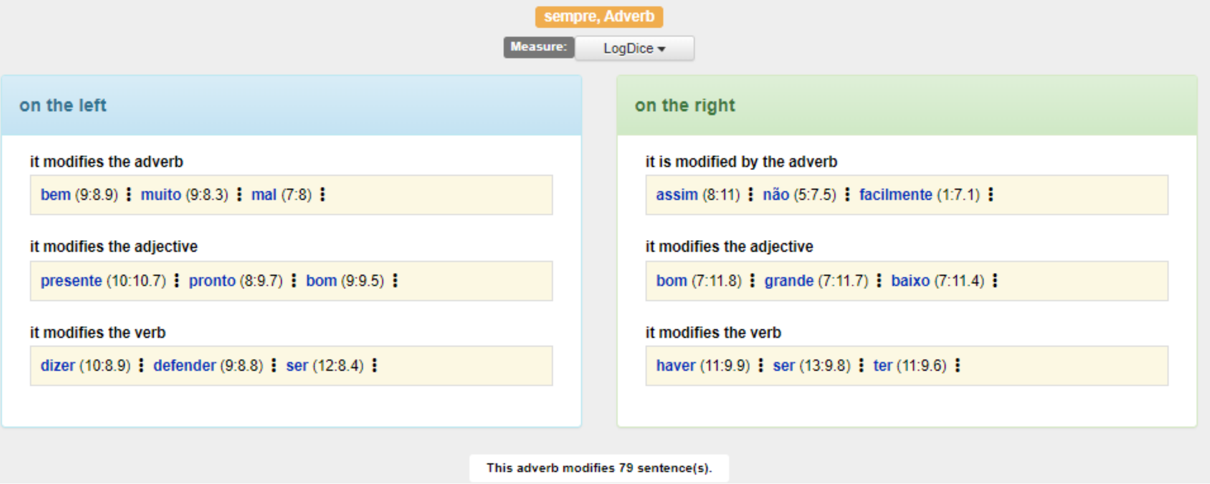
    `MOD_POST_VERB_ADV(dizer,sempre)``;`

- modifies a sentence

  Felizmente, isso pode ficar para outro dia. **'Luckily that can be done in another day.'**

  `MOD_TOP_ADV(Felizmente);`

[3] In STRING, *é que* is often analysed as a *focus* adverb, and in this example with its focusing scope on the noun *verdade*.



sempre, Adverb

Measure: LogDice

on the left

it modifies the adverb

bem (9:8.9) ⋮ muito (9:8.3) ⋮ mal (7:8) ⋮

it modifies the adjective

presente (10:10.7) ⋮ pronto (8:9.7) ⋮ bom (9:9.5) ⋮

it modifies the verb

dizer (10:8.9) ⋮ defender (9:8.8) ⋮ ser (12:8.4) ⋮

on the right

it is modified by the adverb

assim (8:11) ⋮ não (5:7.5) ⋮ facilmente (1:7.1) ⋮

it modifies the adjective

bom (7:11.8) ⋮ grande (7:11.7) ⋮ baixo (7:11.4) ⋮

it modifies the verb

haver (11:9.9) ⋮ ser (13:9.8) ⋮ ter (11:9.6) ⋮

This adverb modifies 79 sentence(s).

Figure 1.9: Co-occurrence of the adverb *sempre* 'always' using LogDice, with a maximum of three results (from [6]).

Concluding this introduction, this dissertation project aims to improve the Explorer system by providing additional information, in the form of more dependencies, and new features; and to better the user experience by reformulating certain ambiguous dependency titles.

1.4 Document Structure

Following the Introduction, this dissertation is further divided into four chapters:

- Chapter 2 describes three other *corpora* analysis tools, as well as the previous version of Syntax Deep Explorer, ending with a table comparing the three systems;
- Chapter 3 depicts the solution that was developed, emphasizing the new features, as well as the problems faced and the reasoning behind their solution;
- Chapter 4 details the methods used to evaluate the developed solution;
- Chapter 5 concludes the dissertation and provides insights into future work.

Chapter 2

Related work

2.1 Related Work

This section illustrates three different *corpus* analysis tools. Each one has its own way of displaying relevant data to the user and consequently, each may have a different utility. While the focus of most of the available systems is on providing examples of the use of a word in a given *corpus*, this project is built on a higher degree of abstraction. The analyzed systems are *DeepDict* [4], *CQPWeb* [9] and *Sketch Engine* [10].

2.1.1 Sketch Engine

Sketch Engine[1] is a commercially developed *corpus* analysis tool, created in 2003 by Lexical Computing Ltd. This service provides most of the functionalities that have been also implemented in the Explorer, and overall is a more supported and more developed tool. This is why the main inspiration for the development of the Explorer has been drawn from Sketch Engine and its functionalities. The most relevant features that this service brings to the table are:

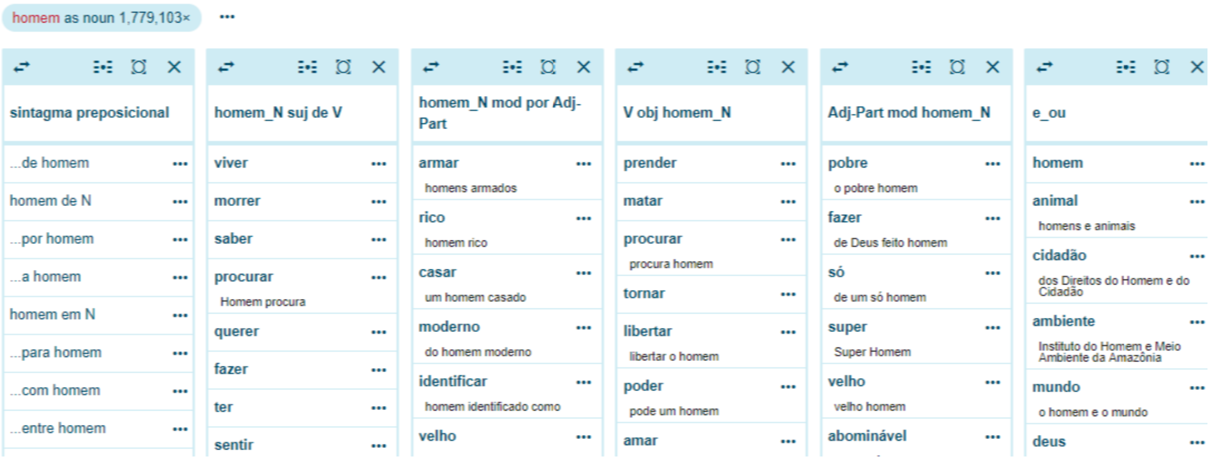
- access to various *corpora* and user inputted *corpora*;
- display of the lexical profiles (or *word sketches*) of a selected lemma. A lexical profile is a one-page, automatic, *corpus-derived* summary of a word's grammatical and collocational behaviour [10].
- comparison between the lexical profiles of two different words;
- Contextual Query Language (CQL), wildcard and regular expression search in concordance;
- displays the prepositional phrases often associated with the desired lemma;
- shows the verbs to which the lemma is a direct or indirect object.

In the following examples, retrieved from the Sketch Engine, the Portuguese Web 2011 was chosen, as it was the largest *corpus* already available on the application and because it featured texts from both Brazil and Portugal.

[1] https://www.sketchengine.eu/ (last visited in 27 of December 2018).

Sketch Engine's main feature is the *word sketch*, a comprehensive and detailed examination of how a chosen lemma is most commonly used in a certain language by showing the most frequently co-occurring words related to that lemma and how they fit together in a sentence.

In Figure 2.1, it is possible to see various columns corresponding to different roles of *homem* 'man' in the *corpus*. For example, the second column (*homem_N suj de V*) displays the verbs to which *homem* 'man' acts as a subject. These results are ordered based on a LogDice score.



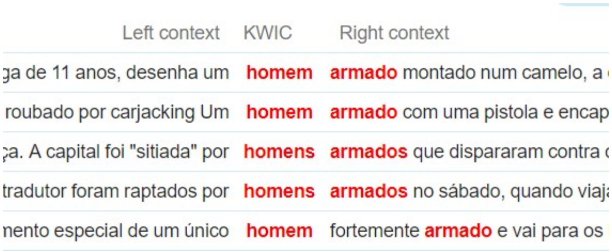
homem as noun 1,779,103×

sintagma preposicional	homem_N suj de V	homem_N mod por Adj-Part	V obj homem_N	Adj-Part mod homem_N	e_ou
...de homem	viver	armar homens armados	prender	pobre o pobre homem	homem
homem de N	morrer	rico homem rico	matar	fazer de Deus feito homem	animal homens e animais
...por homem	saber	casar um homem casado	procurar procura homem	só de um só homem	cidadão dos Direitos do Homem e do Cidadão
...a homem	procurar Homem procura	moderno do homem moderno	tornar	super Super Homem	ambiente Instituto do Homem e Meio Ambiente da Amazônia
homem em N	querer	identificar homem identificado como	libertar libertar o homem	velho velho homem	mundo o homem e o mundo
...para homem	fazer	velho	poder pode um homem	abominável	deus
...com homem	ter		amar		
...entre homem	sentir				

Figure 2.1: Part of the *Word Sketch* of the lemma *homem* 'man' .

Supporting the *word sketch*, the system provides the *concordance* for every use of the lemma. The *concordance* displays the context in which the lemma is used, with examples extracted directly from the *corpus*. Figure 2.2 shows how *homem* 'man' is modified by the adjective *armado* 'armed' using examples from the selected *corpus*. These results are aligned in KWIC format (keyword in context), the target lemma is placed at the center and the co-occurrent word appears on the left or on the right context. Both these words are highlighted.

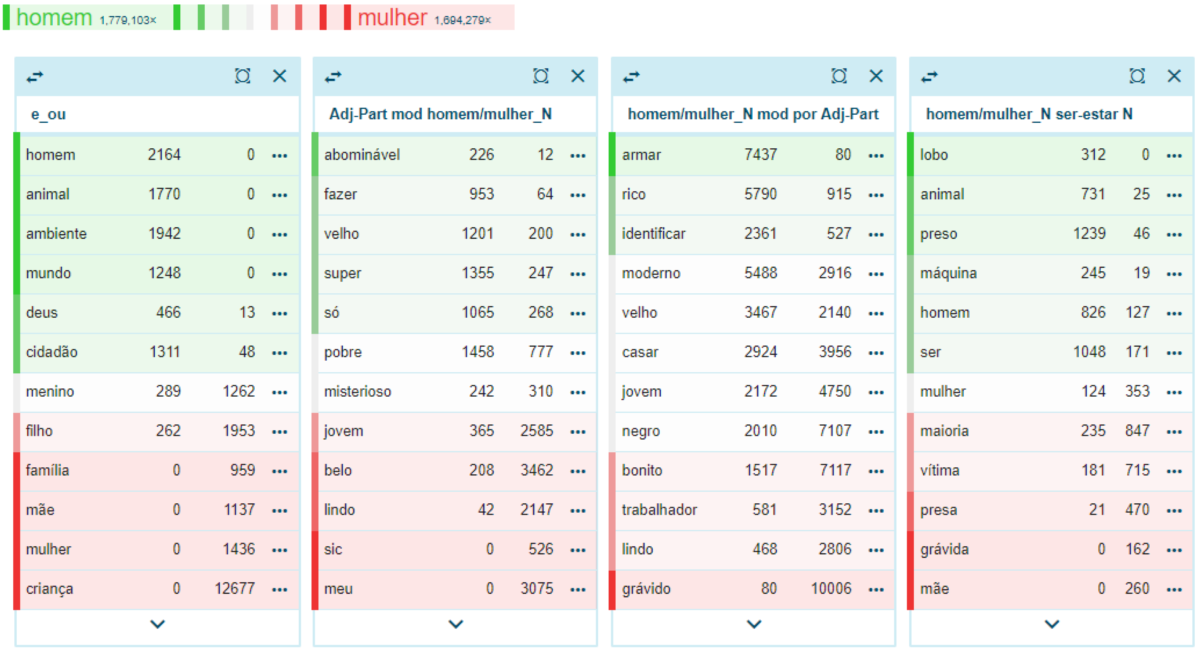
If the user is using the *Concordance* feature, it is also possible to search by wildcards, by regular expressions and by CQL. Wildcards act as placeholders for characters in the lemma facilitating the search for multiple lemmas in the database. This functionality is only used in simple concordance searches. On lemma, word, phrase or character concordance searches, the user can also search by Regex. Regex or Regular Expressions are a sequence of characters that define a search pattern and are mostly used to find words or phrases that share similar traits, like being on the beginning of a sentence or starting with the letter "b". Lastly, the user can also execute a CQL concordance search. CQL is a modified version of Corpus Query Processor, especially made for Sketch Engine. It allows searches with more complex grammatical patterns, for instance, searching for a verb that starts with the letter "m".



Left context	KWIC	Right context
ga de 11 anos, desenha um	homem	armado montado num camelo, a
roubado por carjacking Um	homem	armado com uma pistola e encap
ça. A capital foi "sitiada" por	homens	armados que dispararam contra
tradutor foram raptados por	homens	armados no sábado, quando viaj
nento especial de um único	homem	fortemente armado e vai para os

Figure 2.2: Part of the *Concordance* between the noun *homem* 'man' and the adjective *armado* 'armed' . Extracted from the table *homem_N mod por Adj-Part*.

The *word sketch difference* displays a hybrid *word sketch* between two different lemmas. The *word sketch* on Figure 2.3 shows the difference between *homem* 'man' and *mulher* 'woman' through a color-coded gradient, where darker green lemmas are more prominent when associated with the lemma *homem* 'man' and darker red lemmas are more often related to the lemma *mulher* 'woman' . According to the results on Figure 2.3 *homem* 'man' appears linked with words such as *moderno* 'modern' and *rico* 'rich' as opposed to *mulher* 'woman' , which is linked with such words as *lindo* 'beautiful' and *jovem* 'young' . The *word sketch difference* feature computes these results using only the frequency of the co-occurrences.



homem 1,779,103× mulher 1,694,279×

e_ou		
homem	2164	0
animal	1770	0
ambiente	1942	0
mundo	1248	0
deus	466	13
cidadão	1311	48
menino	289	1262
filho	262	1953
família	0	959
mãe	0	1137
mulher	0	1436
criança	0	12677

Adj-Part mod homem/mulher_N		
abominável	226	12
fazer	953	64
velho	1201	200
super	1355	247
só	1065	268
pobre	1458	777
misterioso	242	310
jovem	365	2585
belo	208	3462
lindo	42	2147
sic	0	526
meu	0	3075

homem/mulher_N mod por Adj-Part		
armar	7437	80
rico	5790	915
identificar	2361	527
moderno	5488	2916
velho	3467	2140
casar	2924	3956
jovem	2172	4750
negro	2010	7107
bonito	1517	7117
trabalhador	581	3152
lindo	468	2806
grávido	80	10006

homem/mulher_N ser-estar N		
lobo	312	0
animal	731	25
preso	1239	46
máquina	245	19
homem	826	127
ser	1048	171
mulher	124	353
maioria	235	847
vítima	181	715
presa	21	470
grávida	0	162
mãe	0	260

Figure 2.3: Part of the *Word Sketch Difference* between the lemma *homem* 'man' and the lemma *mulher* 'woman' .

The *thesaurus* functionality is another interesting feature in Sketch Engine's portfolio. It creates a list of words based on common collocations [10], that is, cases where both words are used in the same context, they would appear in each other's thesaurus. For instance, the word *livro* 'book' would appear in *jornal* 'newspaper' thesaurus and vice versa because they are both usually direct objects of the verb *ler* 'read' .

Despite offering other functionalities, such as *wordlists*, *n-grams count* and *keywords*, the last feature this project covers is the option for the users to input their own *corpus*. When adding *corpora* to Sketch Engine, the users have two choices; uploading the *corpus*, or finding texts on the web.

Uploading files

Sketch Engine allows input in various common formats, both uncompressed formats like *doc*, *html*, *txt* or *pdf*, and compressed formats like *zip* or *gz*. If the *corpus* is annotated it must be provided in Sketch Engine's "vertical" format [10], a format where each word is on a separate line, followed by the specific annotation.

Finding texts on the web

This option is further divided into three more: **URL**: the application downloads a specific page or group of pages. Links on these pages will not be followed; **Website**: the application downloads the content of an entire website. This feature has the limit of 2000 pages per website; **Web search**: using WebBootCat [3], the application creates a *corpus* by taking user-submitted seed words, grouping them randomly in pairs of 3 then inputting the resulting tuple in a Google search and then downloading the first results. Sketch Engine then removes unnecessary content like advertising from the results and processes the remaining content adding it into the *corpus*.

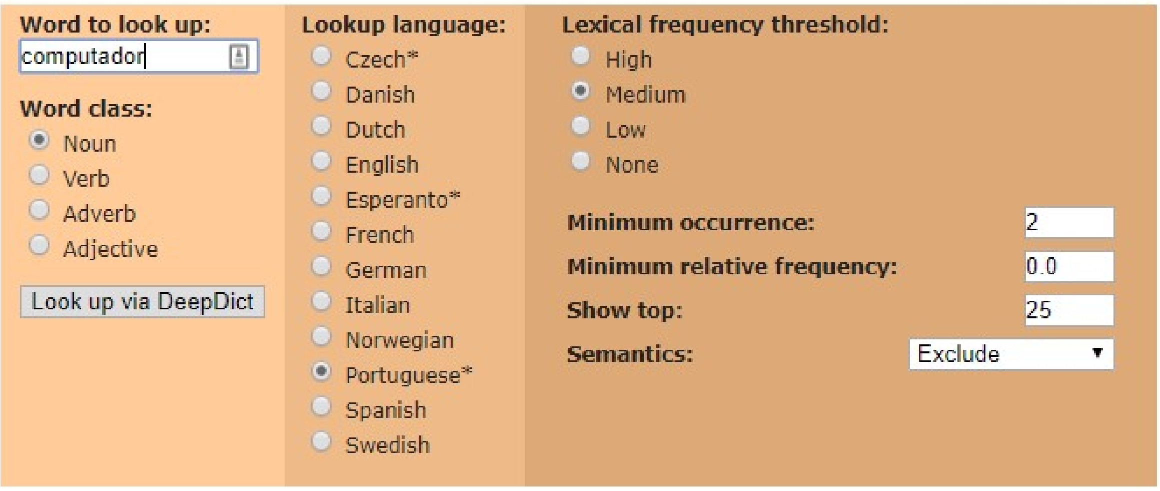
2.1.2 DeepDict

DeepDict[2] is a graphical *corpus-based* dictionary of word relations developed at GrammarSoft and commercially released in September of 2007 [4]. This dictionary tool is studied in detail below on the grounds that it was the main inspiration for the creation of Syntax Deep Explorer and it could offer additional insight on how to improve it.

Syntax Deep Explorer's main menu resembles the one found on DeepDict, the main glaring difference being the option to look up words from *corpora* from 12 different languages. It is possible to see the inspiration that Syntax Deep Explorer took from DeepDict's main menu which also allows the user to change the minimum occurrence of the word to look up as well as how many results to display.

Some other options include the lexical frequency threshold that, when used alongside the minimum occurrence can help eliminating rare words in the results; and the Semantics option which adds semantic tags to the results. This last option, at this time, is only available in the Norwegian *corpus* [4].

[2] https://gramtrans.com/deepdict/ (last visited on April 21, 2020).



Word to look up: computador

Word class:
● Noun
○ Verb
○ Adverb
○ Adjective

Look up via DeepDict

Lookup language:
○ Czech*
○ Danish
○ Dutch
○ English
○ Esperanto*
○ French
○ German
○ Italian
○ Norwegian
● Portuguese*
○ Spanish
○ Swedish

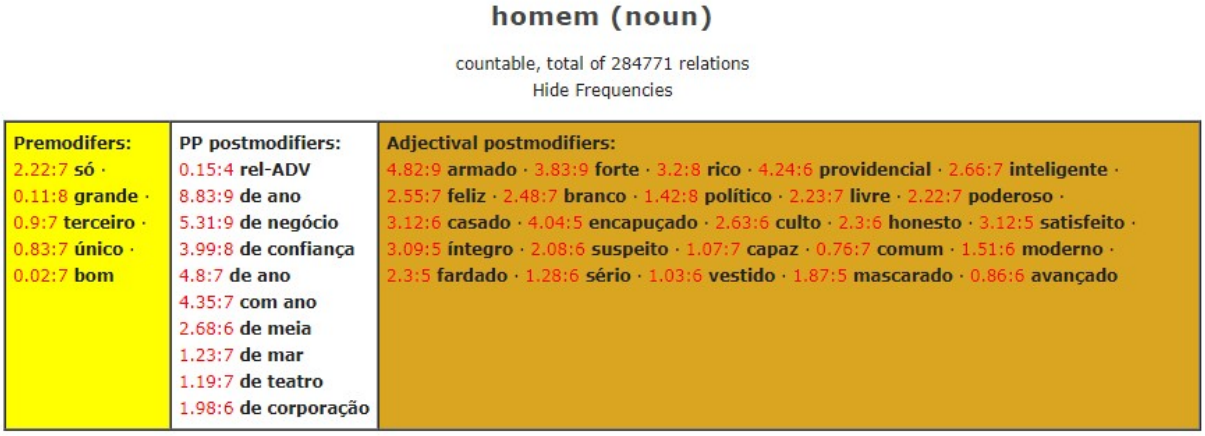
Lexical frequency threshold:
○ High
● Medium
○ Low
○ None

Minimum occurrence: 2
Minimum relative frequency: 0.0
Show top: 25
Semantics: Exclude ▼

Figure 2.4: DeepDict's main menu.

A search made on this system results in a semi-graphical representation of a dictionary entry called a DeepDict *lexicogram* [4]. DeepDict also associates different templates to each word class, for grammatical reasons and to avoid ambiguities. Syntax Deep Explorer took inspiration from the layout of DeepDict's *lexicogram*: The relations of the target word with words that come before the chosen lemma will appear on the left of the screen and the relations that come after will appear on the right. In DeepDict, the order of the results is determined by a single association measure, a modified Pointwise Mutual Information (PMI) score. Words with a higher PMI score appear first than words with a lower score. The Explorer also displays its results ranked by the selected association measure score.

When comparing the results given by the two systems it is possible to see some room for improvement in both of them. For example, Syntax Deep Explorer's noun template only shows to which words (in bold, below) the target lemma acts as a noun complement (e.g *a **vida** de um homem* 'the life of a man'); while DeepDict's noun template only shows the words (in bold, in the example) that act as complement of the target lemma (e.g *homem do **ano*** 'man of the year'). This difference is exemplified on Figure 1.5 and again on Figure 2.5.



homem (noun)

countable, total of 284771 relations
Hide Frequencies

Premodifers:	**PP postmodifiers:**	**Adjectival postmodifiers:**
2.22:7 **só** · 0.11:8 **grande** · 0.9:7 **terceiro** · 0.83:7 **único** · 0.02:7 **bom**	0.15:4 **rel-ADV** 8.83:9 **de ano** 5.31:9 **de negócio** 3.99:8 **de confiança** 4.8:7 **de ano** 4.35:7 **com ano** 2.68:6 **de meia** 1.23:7 **de mar** 1.19:7 **de teatro** 1.98:6 **de corporação**	4.82:9 **armado** · 3.83:9 **forte** · 3.2:8 **rico** · 4.24:6 **providencial** · 2.66:7 **inteligente** · 2.55:7 **feliz** · 2.48:7 **branco** · 1.42:8 **político** · 2.23:7 **livre** · 2.22:7 **poderoso** · 3.12:6 **casado** · 4.04:5 **encapuçado** · 2.63:6 **culto** · 2.3:6 **honesto** · 3.12:5 **satisfeito** · 3.09:5 **íntegro** · 2.08:6 **suspeito** · 1.07:7 **capaz** · 0.76:7 **comum** · 1.51:6 **moderno** · 2.3:5 **fardado** · 1.28:6 **sério** · 1.03:6 **vestido** · 1.87:5 **mascarado** · 0.86:6 **avançado**

Figure 2.5: DeepDict *lexicogram* of the word *homem* 'man' .

2.1.3 CQPweb

CQPweb[3] is a freeware *corpus* analysis tool, developed at Lancaster University, and it is made available via Lancaster University's CQPweb server. This tool resembles most of the tools available on-line and is one of the most popular and complete choices. Unfortunately, this system does not offer a Portuguese tagged *corpus* to explore. However, Centro de Linguística da Universidade de Lisboa (CLUL) developed an online platform[4] using the CQPweb's interface tool [12]. This online platform as well as the original tool supports multiple *corpora*, annotated or not. The *corpora* available on CLUL's platform encompasses some African varieties of Portuguese as well as the Reference Corpus of Contemporary Portuguese (CRPC)[5] .

The main appeal of CQPweb is its flexibility and its support for POS tags [9]. There are two main query methods present in CQPweb-based systems, "Simple query" and "CQP syntax". CQP syntax, uses the Corpus Query Processor [8], a specialized search engine for linguistic research. CQP is a powerful and complex language but it can be daunting for the end user. The Common Elementary Query Language (CEQL) [13] was developed as a simpler alternative to CQP. It gives access to CQP's most used features but in a more user-friendly and accessible way. CEQL is used in CQPweb's Simple query. For contrast here is a search of the common noun (CN) *amo* 'master' in CQP and in CEQL using the tags found in CRPC:

- **CQP:** `[(word = "amo") & (pos = "CN.*")];`
- **CEQL:** `amo_CN*`

The specialized corpus query languages and their flexibility allow for many sophisticated queries, beyond the scope of the other systems here mentioned. However, the time required, as well as the necessary knowledge and mastery of a relatively obscure query language to do so, makes this a less practical option. Having said that, the utility and versatility that CQP brings to the table is certainly useful when addressing this project's objectives.

2.1.4 Conclusions

There are very few available *corpus* analysis tools that aim to do what the Explorer does. This is one of the reasons why the tools analyzed in this section are all very different from each other.

Overall, Sketch Engine is the best performing system, since it does everything that the other systems do in a simpler or in a more complete way. Despite that, every system has something to offer to this project:

- Sketch Engine is a good demonstration of how much information can be extracted from a *corpus* and how to display it without overwhelming the user;
- DeepDict manages to display a lot of relevant information into a compact format;

[3] https://cqpweb.lancs.ac.uk/ (last visited in 21 of April 2020).
[4] http://alfclul.clul.ul.pt/CQPweb/ (last visited in 6 of April 2020).
[5] https://clul.ulisboa.pt/en/projeto/crpc-reference-corpus-contemporary-portuguese (last visited in 21 of April 2020).

- CQPweb offers this project a valuable tool as for its query language, CQP, which is specialized in querying large-sized *corpora*.

Table 2.1 compares the different systems for the aspects that are relevant to this project, namely, if they meet the previously defined objectives. This table offers some insight as to how to proceed in this project, namely:

- No system offered the option to compare the same lemma in different *corpora*. This could mean that this comparison is very difficult to do or that the results may not be sufficiently interesting;
- Sketch Engine accomplishes almost everything that this project is aiming to achieve. This leads to to a more detailed analysis not only of the way this system implemented these functionalities, but also of the manner chosen to present them;
- CQP or the simpler version, CEQL, will likely help with the development of some of the proposed objectives.

Features	**Sketch Engine**	**DeepDict**	**CQPweb**
User submitted *corpora*	Yes	No	No
Show to which verbs the lemma acts as a direct object	Yes	Only available in English	Only with CQP
Comparision between two lemmas within the same *corpus*	Yes	No	No
Comparision of the same lemma in different *corpora*	No	No	No
Prepositional phrases associated with the lemma	Yes	No	Only with CQP
Information regarding prefixes and suffixes	Only with CQL	No	Only with CQP
Association measure	LogDice	Modified PMI	Basic count

Table 2.1: Overview of the available functionalities in each system.

Chapter 3

Solution

This chapter aims to describe the architecture of the implemented solution as well as the problems faced and the reasoning behind the chosen solution. The main goal for the improvement of the Explorer tool was to allow the user to compare the distribution profiles of two different words or of the same word in two different *corpora*, and to display the results in a way that is both legible and insightful.

3.1 Used *Corpora*

Selecting which *corpora* to use is important because the application ultimately relies on *corpora* to provide interesting and relevant results to the end user. For this project, the selected *corpora* were:

- **CETEMPúblico**[1], a *corpus* composed by extracts of texts from the Portuguese daily national newspaper *Público* from 1991 to 1998. This *corpus* is available to the public and contains 175,350,145 words;

- **Parlamento**, a *corpus* composed by minutes from sessions of the Portuguese Parliament from 1976 to 2018. This *corpus* contains 123,633,859 words;

- **Desportivo**, a *corpus* specifically created for this dissertation. It is composed by texts from the Portuguese sports newspapers *O Jogo*, from 1999 to 2005; and *A Bola*, from 2000 to 2006. This *corpus* contains 100,161,374 words.

These *corpora* were selected mainly due to their size and the fact that their source texts were already available at INESC-ID. These *corpora* having different domains was also a benefit since really different writing styles are expected from these domains and, hopefully, it will provide in more diverse results in *corpora* comparisons. The Parlamento *corpus* had been previously processed by STRING, so it was ready to be used for this project. CETEMPúblico also had been previously processed by STRING. However, an updated version of STRING processing chain and of its Portuguese grammar (in XIP) were now available, and since the Explorer aims to be up-to-date with STRING developments, this *corpus* was reprocessed. The Desportivo *corpus* was created from texts from two different sports newspapers, *O*

[1] https://www.linguateca.pt/cetempublico/whatisCETEMP.html

Jogo and *A Bola*. To prepare these sports newspapers texts for a STRING analysis, some processing was done, namely:

- **Converting both sets of texts to UTF-8 encoding.** The original files for both newspapers were in ISO-8859-1 (Latin 1) encoding and had to be converted to UTF-8 in order to be processed by STRING. This conversion was not entirely smooth, since some characters could not be converted and automatically remained in their ISO-8859-1 hexadecimal code. These were later replaced case by case. Some of the most common unconverted character include 0x96, which has replaced by an hyphen, and 0x85, which represents an ellipsis and was replaced by three dots;

- **Removing irrelevant information and advertisements.** These texts were taken from the respective newspapers' websites, and, as such, they were littered with useless information that had to be removed. Some examples include website headers and breadcrumbs (devices to navigate a web page), as well as scheduling information. This type of information was deemed useless for this dissertation, since it does do not form actual sentences and just serve as guidelines to the reader. Since the information contained in these *corpora* span for 6 years, the format in which this information appeared differed wildly. Later years of the *O Jogo* texts also featured large amounts of advertisements. In some cases, up to 25% of the content of a file consisted of advertisements.

  Erasing only irrelevant information is unrealistic within the scope of this project. In this step, the goal was to maximize the amount of irrelevant information extracted while minimizing the amount of relevant information that was lost. To achieve this, several regular expressions were created to try to mimic the traits of irrelevant information and extract it from the text files. For example, lines that had no lowercase letters were deleted from the *O Jogo* texts and lines that had fewer than 50 characters and less than 4 words were deleted from the *A Bola* texts.

  The remaining information after this pruning amounted to roughly 88% of the total texts size.

- **Appending unfinished, split sentence fragments back together.** This was the most challenging part of the *corpus* preparation. Sentences in the *O Jogo* texts spanned numerous lines, meaning that the end of a line was not necessarily the end of a sentence. The goal was to re-unite the split sentences and make sure that they end in a punctuation mark (<.>, <!>, or <?>).

  The most obvious solution is to replace every new line character for a space character and then append a new line character after every full stop. However, two different problems arise when using this solution. First, not every full stop means that the sentence is over. For example, using this method *F.C. Porto* would be divided into three different sentences. An argument can be made to manually add exceptions like these, but these texts use abbreviations very liberally and managing every exception was not feasible. The second problem is the fact that not every sentence in the texts ends with a punctuation mark (<.>, <!>, or <?>), the most obvious example being the titles of articles.

  The implemented algorithm is a bit more complex, as it uses the last word of the previous line and the first word of the next line to predict if they are in the same sentence. If the last character of a line is a number, letter, bracket, parenthesis, hyphen, comma or colon and the first character of the next line is a lowercase letter, a number, a bracket, a parenthesis or a hyphen, then we are dealing with one sentence split across two lines.

This algorithm however, is not flawless. The main problem with it is that it fails to join a sentence if the first letter of the second line is capitalized. This happens frequently, where the second line begins with names of people and organizations. To mitigate the amount of times that this error happens, a second check is made to the last word of the previous line to see if it matches one of many keywords that indicate that the sentence is not finished. These keywords include words like *e* 'and' , *ao* 'to' and *de* 'of' . This check solves some errors but not all. For example, if a sentence mentions a person, that person's first name might be the last word of the previous line and their last name might be the first word of the next line. This type of error was corrected manually, when found.

- **Inserting common unknown words into STRING's dictionaries.** When processing a new *corpus* some words will certainly be missing from STRING's dictionaries. This occurs mostly with foreign proper nouns and technical jargon (in the case of this *corpus*, sports jargon). When STRING encounters an unfamiliar word, it guesses what that word might be and adds the feature "GUESS". A Python script was developed that counted the amount of times each word with the feature "GUESS" appeared in the processed sports texts. The 4,063 unknown words with a frequency above 150 where classified as:

  - 2,867 Proper nouns (Mantorras, Paulinho, Peseiro, Maniche, ...);
  - 520 Sports clubs (Bayern Munich, Lázio, Manchester United, ...);
  - 290 Places (Camp Nou, Sérvia-Montenegro, Old Trafford, ...);
  - 289 Abbreviations (Ita, USA, VIP, Ing, ...);
  - 55 Competitions (Masters Tournament, Champions League, Girabola ...);
  - 28 Common nouns (árbitros-assistentes, ponta-direita, primeira-mão, duplo-amarelo, ...);
  - 14 Sports (gira-Vólei, óquei, tennis, surfing, ...).

  The main challenges with this task were the ambiguity of some tokens and the fact that these tokens may be split by spaces. For example, the football team *Manchester United* appeared as the two unknown words *Manchester* and *United*. This example depicts why this task could not be automated with a high degree of accuracy because a certain degree of sports knowledge was needed. *Manchester* besides being a sports club is also a city, a surname and many other different things. Finally, *Manchester* ended up being classified as a sports club because the probability of it appearing as a sports club in this *corpus* was higher than of it appearing as a city or a surname.

  Football matches constituted another problem with the tokenization. For example, *F. C. Porto-S. L. Benfica* was not processed as a single token so the token *Porto-S.* was on the list of unknown words;

- **Correcting some typos on the *corpus*.** Finally, 250 typos were manually corrected in the *corpus*. These typos usually had a relatively small frequency. However, every improvement results in changes in the association measures of the correctly spelled words. One example of typos that were corrected is the various incorrect spellings of the Portuguese football team *Belenenses*. *Beleneneses, Belenenes, Belenenenses, Blenenses* were the different incorrect ways of spelling *Belenenses* that were found on the *corpus*. Together, the frequency of these misspelled words amounted to 192 occurrences.

These improvements were essential to this dissertation since each change can result in a massive difference in the calculations of the metrics and as such, alter the results displayed by the Explorer.

Table 3.1 is a comparison between the different *corpora* available on the Explorer. While the sizes of Desportivo and Parlamento are very similar, the CETEMPúblico *corpus* is almost twice as large as these two, in terms of size on the database and total number of different co-occurrences and words. In practice, this will result in slower searches on CETEMPúblico, when compared to the other two *corpora*.

Corpus	**CETEMPúblico**	**Parlamento**	**Desportivo**
Size on the database	6.28 Gb	3.47 Gb	3.64Gb
Time to load on the database	4 days, 21 hours and 21 minutes	2 days, 22 hours and 6 minutes	2 days, 23 hours and 59 minutes
Files	4,042	5,175	3,312
Different co-occurrences	8,108,800	4,281,650	4,028,456
Different words	249,100	112,387	129,439
Different nouns	180,674	70,036	92,928
Different verbs	16,303	11,083	10,840
Different adjectives	46,837	26,671	22,207
Different adverbs	5,286	4,597	3,464

Table 3.1: Constitution of the chosen *corpora*.

To support the addition of the new *corpora*, the database underwent some changes. Despite the original database having the *Corpus* table and the *nomeCorpus* (*Corpus* name) column in the appropriate tables, as it can be seen in Figure 1.1, the (initially) single *corpus* database was divided into multiple databases, each database belonging to a *corpus*. In this way, the addition of new *corpora* will not compromise the search speed of existing processed *corpora* and the addition and removal of *corpora* from the database will be simpler and faster, making this project more scalable. To add a processed *corpus* to the Explorer, one now just has to add a folder with the desired *corpus* name on the root of the database storage location and inside that folder add the database with the processed *corpus*, giving the name 'db_deep.db'.

3.2 Implementation

This section covers the process behind the three main challenges of this project. Each one of these challenges will be presented in a different subsection. Subsection 3.2.1 explains the approach taken to compare two different Explorer searches, be it different words on the same *corpus* or the same word in different *corpora*. Subsection 3.2.2 gives a detailed explanation of the web interface, making the link between what the application does and what the user sees. Finally, subsection 3.2.3 covers how are the target lemma and co-occurrent lemma are highlighted in the example sentences.

3.2.1 Comparison

The comparison of distributional profiles is the main contribution of this dissertation. This subsection aims to describe in-depth how this comparison is made, and the criteria to make a result appear on the user's screen. The response time of the comparison of distributional profiles is evaluated later on Subsection 4.2.2.

Two types of comparisons can be executed in this solution:

- A comparison where the user selects two different lemmas and the Explorer compares them in the context of a single *corpus* (*Word Comparison*),
- A comparison where the user selects just one lemma to be compared across two different *corpora* (*Corpora Comparison*).

The logistics behind these two comparisons is very similar. A comparison consists of two different Explorer searches, the only difference being the limit of the Structured Query Language (SQL) query. While a normal Explorer search has a limit set by the user selected via the *Maximum number of co-occurrences to display* option in the main screen, a comparison search does not utilize this value directly. Similarly, the *Minimum Frequency* established by the user in the main screen is not used directly in the query. The purpose of this change is for the search to return every result in order to achieve a more accurate comparison.

Both searches are returned in full to the Javascript part of the program were they are pruned to display only the relevant information to the user. The program iterates over the positions of the *first* search (`PRE_WORD, POST_WORD, PRE_VERB, POST_VERB`) and the type of dependency (`SUBJ_SEM_PROP`, etc.) and in every iteration, it executes the following changes:

- **Select the lemmas to appear in the result screen.** An array is created with the co-occurrent lemmas with the highest association measure scores from both search results. Since the search results are sorted by the selected measure, extracting the co-occurrent lemmas with the highest score is done by retrieving the first *X* positions of each result object with *X* being the value the user selected in the *Maximum number of co-occurrences to display* option on the main menu;
- **Calculate the metrics for each word that will appear on the result screen.** The array with the highest scored lemmas is iterated over, and every lemma is searched in the original result objects. Since using a key to return a value from a Javascript object has a temporal complexity of `O(1)`, this type of processing can be done in a reasonable execution time. In this step, a significance measure is also calculated that will determine the percentages shown on the result screen. This measure is explained in further detail on subsection 3.2.2;

- **Eliminate irrelevant results.** Finally, in this step the results are filtered using the user-defined option *Minimum Frequency* in order to eliminate rarer co-occurrences.

A comparison distributional profile has two inconsistencies with the user-defined settings. The first inconsistency is the fact that there can appear more results in each dependency than what the user requested.

The comparison result screen attempts to merge the distributional profiles of two individual search results, for example, if a user compares the word *comida* 'food' in the *corpus* Parlamento with the word *comida* on the *corpus* CETEMPúblico, the Explorer will merge the distributional profile of *comida* in Parlamento (*profile A*) with the one resulting from the search for *comida* on CETEMPúblico (*profile B*). So, for example, if the user selected the *Maximum number of co-occurrences to display* to 10 and *profile A* shows the 10 co-occurrent words with the highest score in the selected association measure, in a certain dependency-property pattern and *profile B* also shows 10 different co-occurrent words in the same dependency-property pattern. Thus, the comparison result screen shows all 20 distinct co-occurrent words on that dependency-property pattern. This was done in order to mimic the simple search, allowing every result that would appear on the simple search of one lemma to appear in a comparison where that lemma is present.

The second inconsistency consists in displaying results with frequency below the selected *Minimum Frequency*. As mentioned previously, the *Minimum Frequency* is still used to filter undesired results. Assuming, for a given co-occurrence result in a comparison, *frequency1* is the frequency of the co-occurrence of the first target lemma with the co-occurrent word and *frequency2* is the frequency of the co-occurrence of the second target lemma with the same co-occurrent word. Only one of these values needs to be greater than or equal to the *Minimum Frequency*. This eliminates results where *frequency1* and *frequency2* are both less than *Minimum Frequency*. The results fall under three categories:

- **Both *frequency1* and *frequency2* are greater than *Minimum Frequency***, the most obvious result to include since it matches the user defined criteria;

- **Either *frequency1* or *frequency2* is greater than *Minimum Frequency* while the other value is 0**, this is another easy conclusion to make, some lemmas have other co-occurrences that others do not have, so there is no reason for them to be cut off. On the interface the percentages for these values are 100% for the lemma with higher frequency of co-occurrences and 0% for the lemma with the no co-occurrences;

- **Either *frequency1* or *frequency2* is greater than the *Minimum Frequency* and the other value is greater than 0 but less than *Minimum Frequency***, when this happens two choices are possible, assuming *frequency1* is the value that is higher than *Minimum Frequency*, the system can display *frequency1* as 100% and *frequency2* as 0%, which could be true since this co-occurrence occurs less times than the desired *Minimum Frequency*, however, this solution displays the actual percentage of the co-occurrence frequency. The *Minimum Frequency* setting acts as a way to eliminate rare and unusual co-occurrences from the results in order to favor more consistent ones, however, since one of the searches returned a frequency above that of the defined *Minimum Frequency* there is no need to hide the less frequent results.

Both these inconsistencies are explained to the user in the help section of the Explorer.

As a result of the methodology used to execute the comparison, the symmetry of the comparison can be assured, meaning that changing the order of the lemmas only results in aesthetic differences, namely the color associated with each lemma; and, since the results are ordered by their normalized frequency, the order in which they appear also changes.

3.2.2 Web Interface

This subsection emphasizes the differences between the Explorer's old version and the new one. The web interface can be divided into several parts, namely:

- The main screen where the user selects which type of search to execute and fills in the parameters;
- An Explorer search result screen;
- The newly added Explorer comparison result screen.

The main screen was changed in order to allow the user to select a *corpus*, in which to search for the desired lemma and to allow the user to compare two lemmas in a *corpus* or to compare the same lemma in two different *corpora*.

Figure 3.1 shows the new main screen. In the first row the most notable addition was the addition of a field with a dropdown box where the user can select the desired *corpus*.

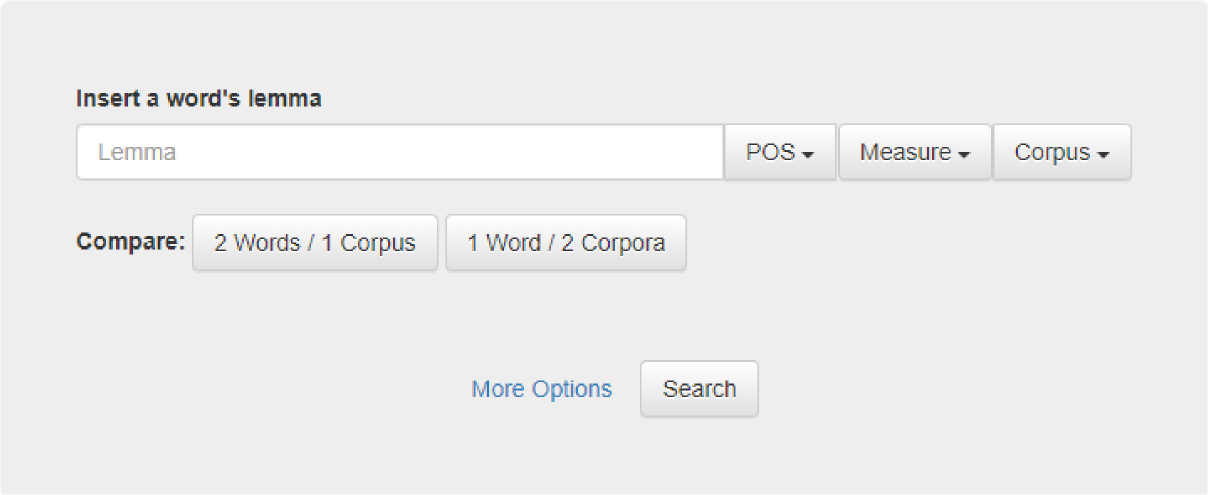
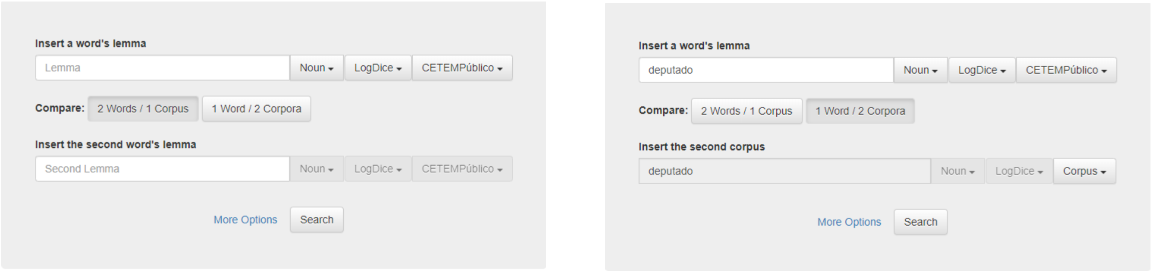


Figure 3.1: New main screen.

Two buttons were added to allow the user to compare different words and different *corpora*. Only one of these buttons can be active at a time, activating one of them while the other is activated will deactivate the other button and activate the desired one. Activating a comparison box displays an additional row to the user. This row contains only one additional field to be filled by the user while all the other redundant fields are greyed out since they must be the same as the ones on the first row, they only act as a reminder to the user. The field that is not greyed out depends on the desired comparison. If the user requests a *Word Comparison* by clicking on the button named *2 Words / 1 Corpus*, he or she has to provide the additional lemma to compare, as shown on Figure 3.2a. If the user requests a *Corpora Comparison* by clicking on the button named *1 Word / 2 Corpora*, he or she has to provide the additional *corpus* to compare as shown on Figure 3.2b.

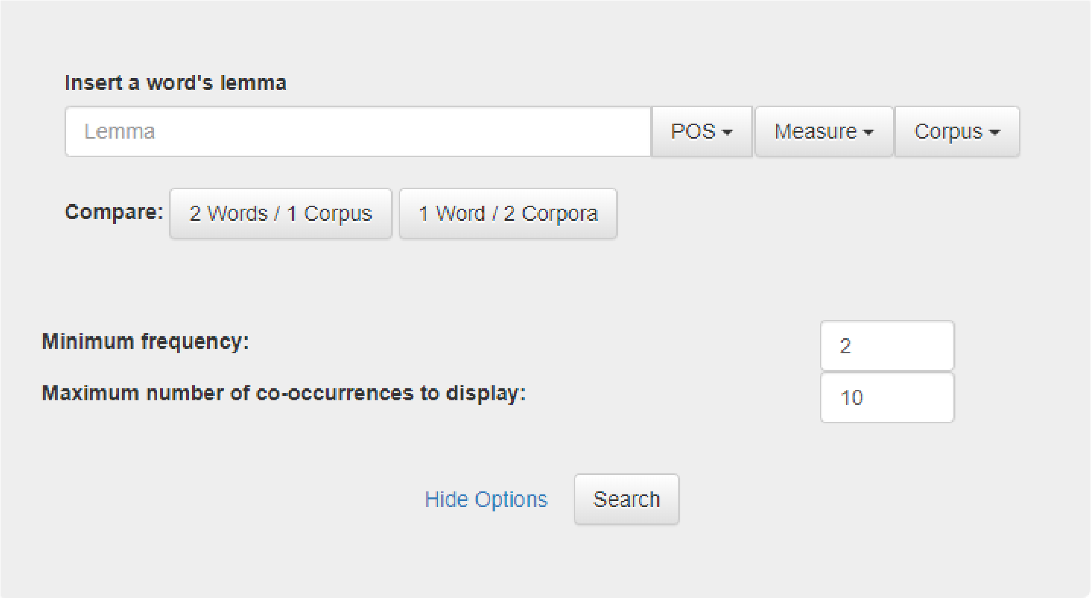


(a) Word comparison main screen.

(b) Corpora comparison main screen.

Figure 3.2: Comparison main screen.

Clicking on *More Options* reveals the two options that were present in the previous version of the Explorer, *Minimum frequency* changed from *Minimum occurrences* and *Maximum number of co-occurrences to display* changed from *Word's maximum number*. Figure 3.3 represents the main screen when *More Options* is clicked, showing the hidden options and their default values.



Minimum frequency: 2

Maximum number of co-occurrences to display: 10

Figure 3.3: Main screen with more options.

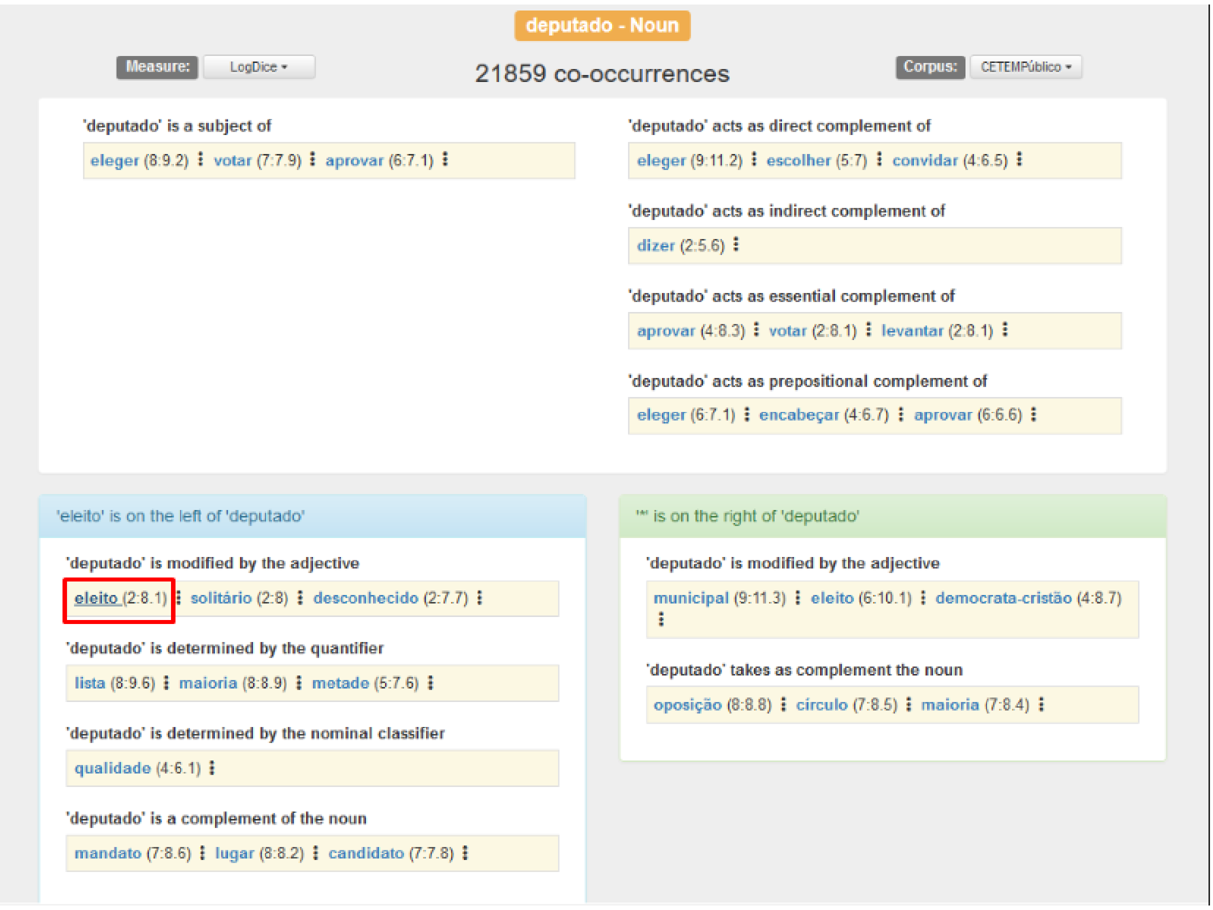
The layout of the result screen also saw some changes. In the previous version of the Explorer the results were deciphered by adding the panel title ("on the left" / "on the right") to the dependency title. For example, if an adjective has a dependency named **it modifies the noun** and it is "on the right" table we would know that the adjective modifies the noun on its right. This layout originated two problems:

- **“it” is an ambiguous term.** It may represent either the target word or the co-occurrent word leading to some confusion (especially if the target word and the co-occurrent word share the same POS);

- **The title of the position table alone does not clarify the relative position of the words.** One example of this issue is present on the noun result screen. **it is complement of the noun** is placed “on the right” panel. Using the logic described above, the user derives the sentence **it is complement of the noun on the right**. In this case the target lemma is on the right of the co-occurrent lemma. However, when deriving the sentence **it is modified by the adjective on the right**, we conclude that the target lemma is on the left of the co-occurrent lemma. Assuming the user searched for the lemma *homem* ‘man’ , an example of the dependency **it is complement of the noun** "on the right" is *vida de homem* ‘life of a man’ while an example of the dependency **it is modified by the adjective** "on the right" could be *homem rico* ‘rich man’ .

Concluding, when examining an Explorer search, dependencies that appear on the “on the right” panel can have their target lemma either to the left or to the right of the co-occurrent lemma. The same is true for the “on the left” panel.

The layout of the new system organizes the information by position relative to the target lemma. Dependencies on the left side of the lemma appear on its left side and the same is true for the right side. The word “it” was replaced by the target lemma in order to better clarify the dependency title. The titles of the panels also changed, “on the left” was changed to “’*’ appears on the left of ’*target lemma*’ and “on the right” was changed to ’*’ appears on the right of ’*target lemma*’ where the *target lemma* is the lemma that has been searched for. Hovering over a result in any of these panels will alter the panel title to include the co-occurrent word, replacing the asterisk. In addition to this feature, the co-occurrence frequency of the target lemma, in other words, it is the number of times that this lemma appears in a co-occurrence with another lemma is now displayed on top of the result screen. Figure 3.4 depicts an example of the new Explorer result screen when *eleito* ‘elected’ is hovered on the left panel.



deputado - Noun

Measure: LogDice ▾

21859 co-occurrences

Corpus: CETEMPúblico ▾

'deputado' is a subject of

eleger (8:9.2) ⋮ votar (7:7.9) ⋮ aprovar (6:7.1) ⋮

'deputado' acts as direct complement of

eleger (9:11.2) ⋮ escolher (5:7) ⋮ convidar (4:6.5) ⋮

'deputado' acts as indirect complement of

dizer (2:5.6) ⋮

'deputado' acts as essential complement of

aprovar (4:8.3) ⋮ votar (2:8.1) ⋮ levantar (2:8.1) ⋮

'deputado' acts as prepositional complement of

eleger (6:7.1) ⋮ encabeçar (4:6.7) ⋮ aprovar (6:6.6) ⋮

'eleito' is on the left of 'deputado'

'deputado' is modified by the adjective

eleito (2:8.1) ⋮ solitário (2:8) ⋮ desconhecido (2:7.7) ⋮

'deputado' is determined by the quantifier

lista (8:9.6) ⋮ maioria (8:8.9) ⋮ metade (5:7.6) ⋮

'deputado' is determined by the nominal classifier

qualidade (4:6.1) ⋮

'deputado' is a complement of the noun

mandato (7:8.6) ⋮ lugar (8:8.2) ⋮ candidato (7:7.8) ⋮

'*' is on the right of 'deputado'

'deputado' is modified by the adjective

municipal (9:11.3) ⋮ eleito (6:10.1) ⋮ democrata-cristão (4:8.7) ⋮

'deputado' takes as complement the noun

oposição (8:8.8) ⋮ círculo (7:8.5) ⋮ maioria (7:8.4) ⋮

Figure 3.4: Example of a change to the title of the left panel when *eleito* 'elected' is hovered over.

The left and right panels aim to separate co-occurrent lemmas that appear on the left of the target lemma from those that appear to its right. Unlike the left and right panels, the verb section does not guarantee the position of the target lemma in relation to the co-occurrent lemmas. A co-occurrent lemma placed on `PRE_VERB` can appear either on the left or on the right of the target lemma in the example sentences. The distinction between `PRE_VERB` and `POST_VERB` only affects the placement on the result screen, and not the placement of the co-occurrent word relative to the target word on the example sentences. This distinction is only made to maintain the consistency across the result screens of different POS (the subject relation always appears on the left, and the remaining dependencies always appear to the right of the verb section of the result screen).

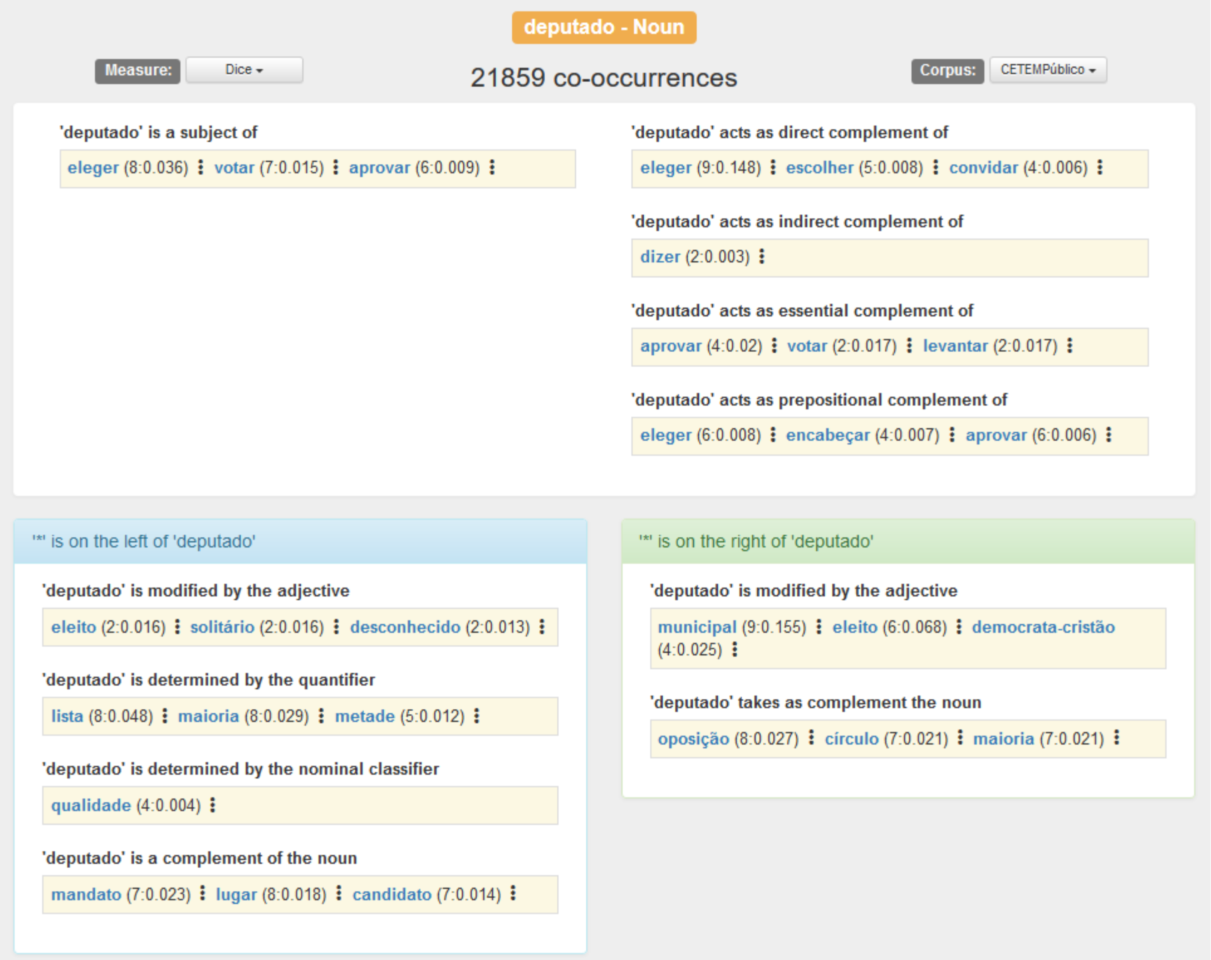
The full list of the dependencies available for the new version of the Explorer is depicted below. Each example was taken from the CETEMPúblico *corpus* with a minimum frequency of 2 and a maximum number of co-occurrences to display set to 3 in order to save space. The dependencies in bold are the new inclusions in this version. In addition to this, the title of each dependency underwent some changes, either adjusting the title to fit the target lemma in it (indicated by an asterisk), or by making the title more clear to the user or by correcting a few typos. Only new additions to the system will have examples:

Noun

Overall, the noun screen had the most changes with the addition of the `PRE_VERB` and POST_VERB. Since these dependencies were already being used in the Verb result screen, inserting them on the noun screen seemed logical.

Figure 3.5 is an example of a search for a noun. The list below encompasses all the information that is possible to appear on the noun result screen:

- `PRE_VERB`
  - *** is a subject of**
    ***Ele resistiu.* 'He resisted.'**
    `SUBJ_SEM_PROP(resistir,Ele);`
- `POST_VERB`
  - *** acts as direct complement of**
    ***Eu leio um livro.* 'I read a book.'**
    `CDIR_SEM_PROP(ler,livro);`
  - *** acts as indirect complement of**
    ***Ela contou ao jornalista.* 'She told the journalist.'**
    `CINDIR_SEM_PROP(contar,repórter);`
  - *** acts as essential complement of**
    ***Ele vai ser lembrado por outras pessoas* 'He will be remembered by other people.'**
    `COMPL_SEM_PROP(lembrar,pessoas);`
  - *** acts as prepositional complement of**
    ***Isto está na origem de problemas.* 'This is the source of problems.'**
    `MOD_VERB_NOUN(estar,origem);`
- `PRE_WORD`
  - * is modified by the adjective
  - * is determined by the quantifier
  - * is determined by the nominal classifier
  - *** is a complement of the noun**
    ***Eu como salada de fruta.* 'I eat fruit salad.'**
    `MOD_POST_NOUN_NOUN(salada,fruta);`
- `POST_WORD`
  - * is modified by the adjective
  - * takes as complement the noun


deputado - Noun

Measure: Dice ▾

21859 co-occurrences

Corpus: CETEMPúblico ▾

'deputado' is a subject of

eleger (8:0.036) ⋮ votar (7:0.015) ⋮ aprovar (6:0.009) ⋮

'deputado' acts as direct complement of

eleger (9:0.148) ⋮ escolher (5:0.008) ⋮ convidar (4:0.006) ⋮

'deputado' acts as indirect complement of

dizer (2:0.003) ⋮

'deputado' acts as essential complement of

aprovar (4:0.02) ⋮ votar (2:0.017) ⋮ levantar (2:0.017) ⋮

'deputado' acts as prepositional complement of

eleger (6:0.008) ⋮ encabeçar (4:0.007) ⋮ aprovar (6:0.006) ⋮

'*' is on the left of 'deputado'

'deputado' is modified by the adjective

eleito (2:0.016) ⋮ solitário (2:0.016) ⋮ desconhecido (2:0.013) ⋮

'deputado' is determined by the quantifier

lista (8:0.048) ⋮ maioria (8:0.029) ⋮ metade (5:0.012) ⋮

'deputado' is determined by the nominal classifier

qualidade (4:0.004) ⋮

'deputado' is a complement of the noun

mandato (7:0.023) ⋮ lugar (8:0.018) ⋮ candidato (7:0.014) ⋮

'*' is on the right of 'deputado'

'deputado' is modified by the adjective

municipal (9:0.155) ⋮ eleito (6:0.068) ⋮ democrata-cristão (4:0.025) ⋮

'deputado' takes as complement the noun

oposição (8:0.027) ⋮ círculo (7:0.021) ⋮ maioria (7:0.021) ⋮

Figure 3.5: Result of a co-occurrence search for the noun *deputado* 'congressman' in the new system.

Verb

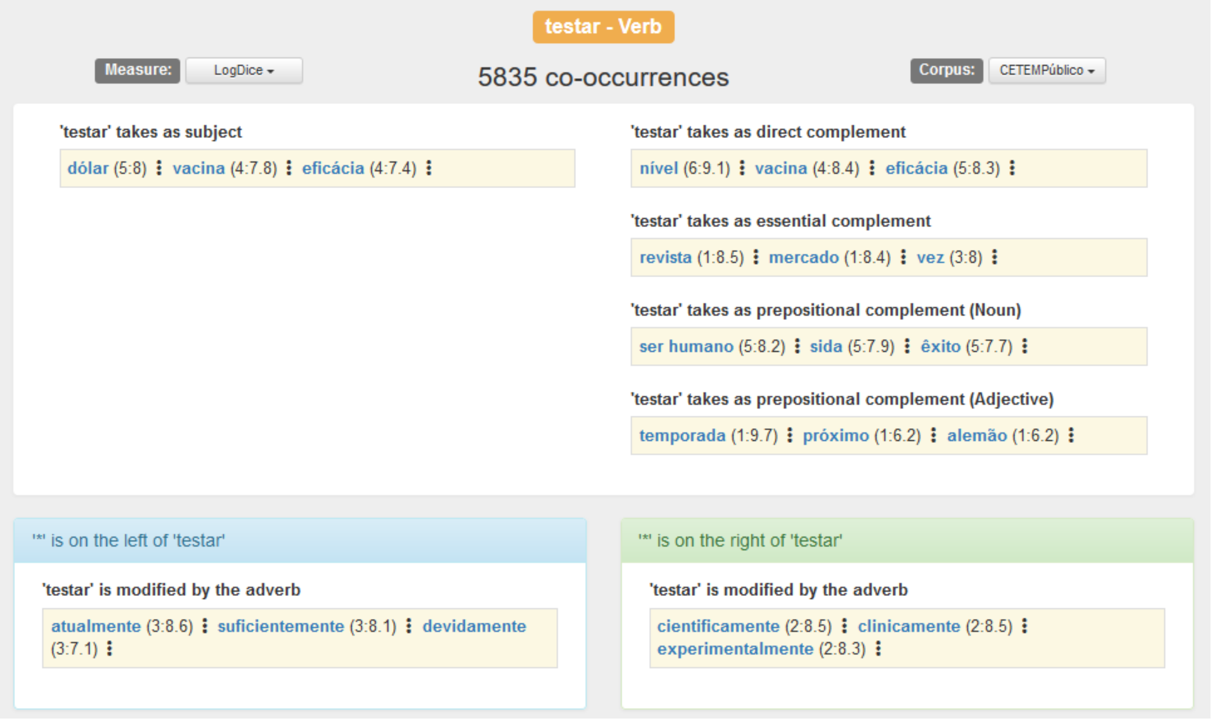
The only changes made to the verb screen was the modification of the dependency titles. However, these changes are worth noting, since they clear some ambiguities that users may have had with the previous version.

Figure 3.6 is an example of a search for a verb. The dependencies that may appear on this screen are:

- PRE_VERB
  - * takes as subject
- POST_VERB
  - * takes as direct complement
  - * takes as indirect complement
  - * takes as essential complement
  - * takes as prepositional complement (Noun)
  - * takes as prepositional complement (Adjective)
- PRE_WORD

  - * is modified by the adverb

- POST_WORD
  - * is modified by the adverb



testar - Verb

Measure: LogDice

5835 co-occurrences

Corpus: CETEMPúblico

'testar' takes as subject

dólar (5:8) ⋮ vacina (4:7.8) ⋮ eficácia (4:7.4) ⋮

'testar' takes as direct complement

nível (6:9.1) ⋮ vacina (4:8.4) ⋮ eficácia (5:8.3) ⋮

'testar' takes as essential complement

revista (1:8.5) ⋮ mercado (1:8.4) ⋮ vez (3:8) ⋮

'testar' takes as prepositional complement (Noun)

ser humano (5:8.2) ⋮ sida (5:7.9) ⋮ êxito (5:7.7) ⋮

'testar' takes as prepositional complement (Adjective)

temporada (1:9.7) ⋮ próximo (1:6.2) ⋮ alemão (1:6.2) ⋮

'*' is on the left of 'testar'

'testar' is modified by the adverb

atualmente (3:8.6) ⋮ suficientemente (3:8.1) ⋮ devidamente (3:7.1) ⋮

'*' is on the right of 'testar'

'testar' is modified by the adverb

cientificamente (2:8.5) ⋮ clinicamente (2:8.5) ⋮ experimentalmente (2:8.3) ⋮

Figure 3.6: Result of a co-occurrence search for the verb *test* 'to test' in the new system.

Adjective

Figure 3.7 illustrates the new result screen for the search of an adjective. A new field displays the verbs to which the adjective acts as a prepositional complement (when parsed as the head of that constituent). The complete list of possible dependencies that can occur in this result screen is:

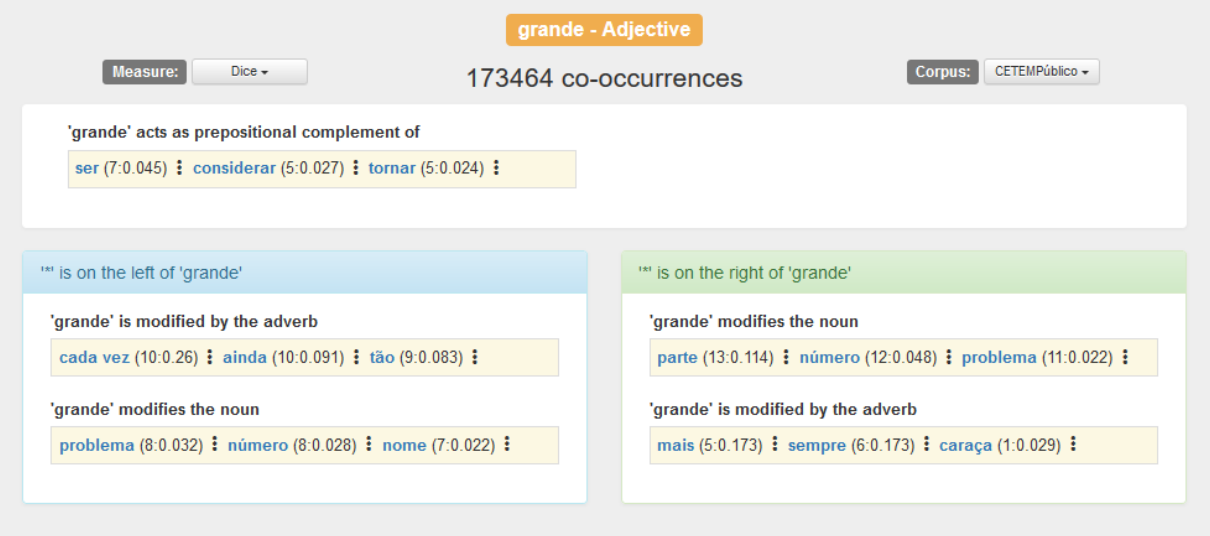
- PRE_VERB
  - *** acts as prepositional complement of**

    As coisas estão a mudar para melhor. **'Things are changing for the better.'**

    `MOD_VERB_ADJ(mudar,melhor);`

- PRE_WORD
  - * is modified by the adverb
  - * modifies the noun

- POST_WORD
  - * is modified by the adverb
  - * modifies the noun



173464 co-occurrences

'grande' acts as prepositional complement of

ser (7:0.045) ⋮ considerar (5:0.027) ⋮ tornar (5:0.024) ⋮

'*' is on the left of 'grande'

'grande' is modified by the adverb

cada vez (10:0.26) ⋮ ainda (10:0.091) ⋮ tão (9:0.083) ⋮

'grande' modifies the noun

problema (8:0.032) ⋮ número (8:0.028) ⋮ nome (7:0.022) ⋮

'*' is on the right of 'grande'

'grande' modifies the noun

parte (13:0.114) ⋮ número (12:0.048) ⋮ problema (11:0.022) ⋮

'grande' is modified by the adverb

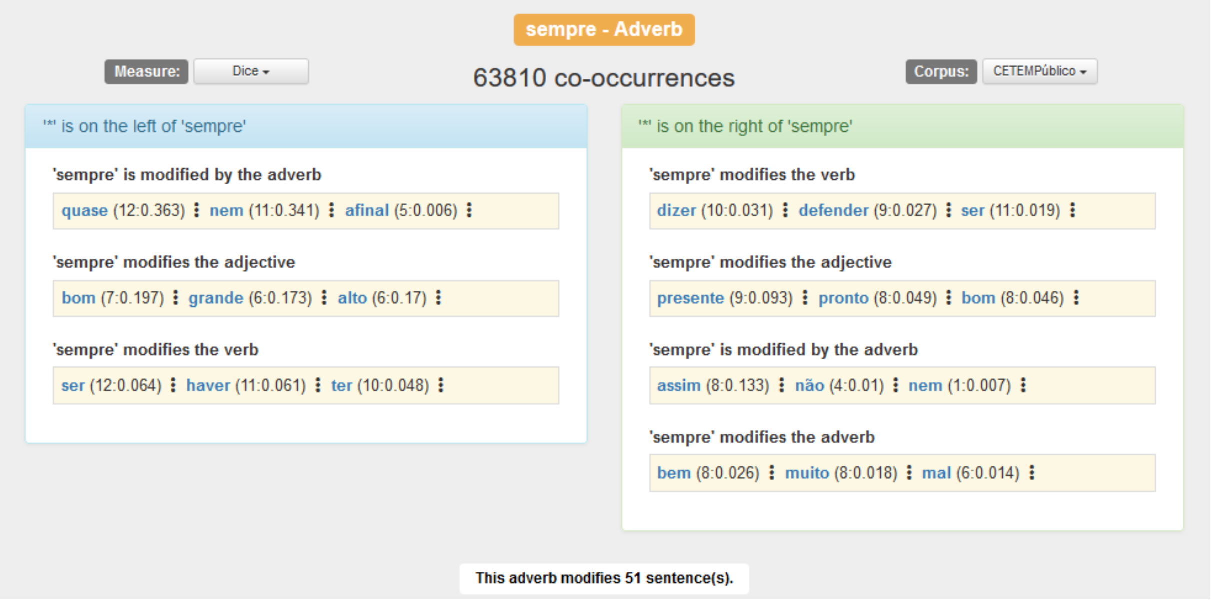
mais (5:0.173) ⋮ sempre (6:0.173) ⋮ caraça (1:0.029) ⋮

Figure 3.7: Result of a co-occurrence search for the adjective *grande* 'big' in the new system.

Adverb

The adverb screen only saw changes to the dependency titles as well. Figure 3.8 represents an example of an adverb result screen and the list below shows all the possible dependencies that may appear:

- PRE_WORD
    - * is modified by the adverb
    - * modifies the adjective
    - * modifies (focus) the noun
    - * modifies the verb
- POST_WORD
    - * modifies the adverb
    - * modifies the adjective
    - * modifies (focus) the noun
    - * modifies the verb
- * modifies the sentence



sempre - Adverb

Measure: Dice

63810 co-occurrences

Corpus: CETEMPúblico

'*' is on the left of 'sempre'

'sempre' is modified by the adverb

quase (12:0.363) ⋮ nem (11:0.341) ⋮ afinal (5:0.006) ⋮

'sempre' modifies the adjective

bom (7:0.197) ⋮ grande (6:0.173) ⋮ alto (6:0.17) ⋮

'sempre' modifies the verb

ser (12:0.064) ⋮ haver (11:0.061) ⋮ ter (10:0.048) ⋮

'*' is on the right of 'sempre'

'sempre' modifies the verb

dizer (10:0.031) ⋮ defender (9:0.027) ⋮ ser (11:0.019) ⋮

'sempre' modifies the adjective

presente (9:0.093) ⋮ pronto (8:0.049) ⋮ bom (8:0.046) ⋮

'sempre' is modified by the adverb

assim (8:0.133) ⋮ não (4:0.01) ⋮ nem (1:0.007) ⋮

'sempre' modifies the adverb

bem (8:0.026) ⋮ muito (8:0.018) ⋮ mal (6:0.014) ⋮

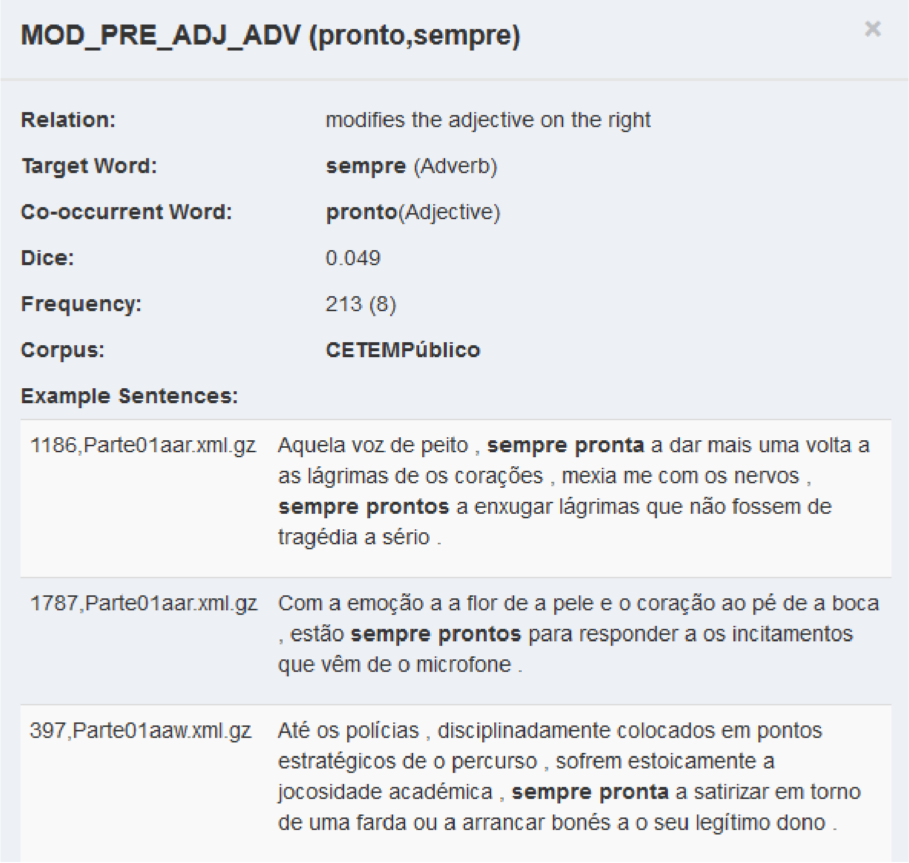
This adverb modifies 51 sentence(s).

Figure 3.8: Result of a co-occurrence search for the adverb *sempre* 'always' in the new system.

These additions were made to better convey a sense of symmetry and cohesion between result screens. For example, since the verb result screen shows which nouns act as direct complement to that verb, the noun result screen should also show to which verbs that noun acts as direct complement. The impact that these additions have on the response time of the Explorer is evaluated on Subsection 4.2.1.

To the result screen was also added the number of co-occurrences of the target word. The number of co-occurrences may differ from the number of times a word appears on the *corpus*. Another addition to the noun result screen was the `PRE_VERB` and `POST_VERB` co-relations. The verb co-relations were also added to the noun and adjective result screens as this information was already present in the database but not visible in the interface. Still no changes were made on the result screen to the actual representation of the co-occurrences, each co-occurrence is followed by the logarithm of the frequency a colon and the corresponding selected measure, like in the previous version of the tool.

Clicking on a co-occurrence result will generate a pop-up window with a detailed view of that co-occurrence, additional information, and example sentences from where that co-occurrence was extracted. Figure 3.9 represents this detailed view. The most prominent feature added in this screen was the highlighting of the target lemma and the co-occurrent lemma, making it easier for the user to find from where exactly was this co-occurrence within the example sentences. This feature is further described and evaluated in the sections below. In addition to this feature, the *corpus* row was added in order to better contextualize the results.



MOD_PRE_ADJ_ADV (pronto,sempre)

Relation:	modifies the adjective on the right
Target Word:	**sempre** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.049
Frequency:	213 (8)
Corpus:	**CETEMPúblico**

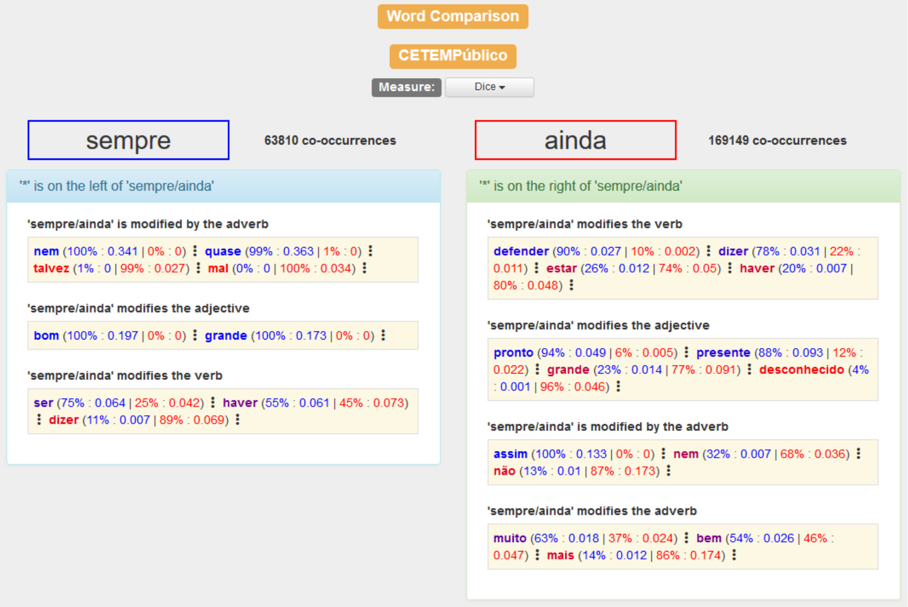
Example Sentences:

1186,Parte01aar.xml.gz	Aquela voz de peito , **sempre pronta** a dar mais uma volta a as lágrimas de os corações , mexia me com os nervos , **sempre prontos** a enxugar lágrimas que não fossem de tragédia a sério .
1787,Parte01aar.xml.gz	Com a emoção a a flor de a pele e o coração ao pé de a boca , estão **sempre prontos** para responder a os incitamentos que vêm de o microfone .
397,Parte01aaw.xml.gz	Até os polícias , disciplinadamente colocados em pontos estratégicos de o percurso , sofrem estoicamente a jocosidade académica , **sempre pronta** a satirizar em torno de uma farda ou a arrancar bonés a o seu legítimo dono .

Figure 3.9: Detailed view of a co-occurrence result.

Figure 3.10 depicts the result screen of a *Word Comparison*, i.e. a comparison between two different lemmas in the same *corpus*. The main differences between this screen and the simple search screen are the header and the values next to each co-occurrence result. When comparing two different lemmas in a given *corpus*, the header of the result screen has five different components:

- The type of comparison, in this case being a *Word Comparison*;
- The name of the *corpus* where the comparison is taking place;
- A drop-down menu indicating the measure being used and allowing the user to change it;
- On the left side, enveloped by a blue rectangle is the first lemma the user inserted and, at its' right, the number of co-occurrences of this lemma in the chosen *corpus*, present in the database;
- On the right side, enveloped by a red rectangle is the second lemma the user inserted and, at its' right, the number of co-occurrences of this lemma in the chosen *corpus*, present in the database;



Word Comparison

CETEMPúblico

Measure: Dice

sempre

63810 co-occurrences

ainda

169149 co-occurrences

'*' is on the left of 'sempre/ainda'

'sempre/ainda' is modified by the adverb

nem (100% : 0.341 | 0% : 0) ⋮ quase (99% : 0.363 | 1% : 0) ⋮ talvez (1% : 0 | 99% : 0.027) ⋮ mal (0% : 0 | 100% : 0.034) ⋮

'sempre/ainda' modifies the adjective

bom (100% : 0.197 | 0% : 0) ⋮ grande (100% : 0.173 | 0% : 0) ⋮

'sempre/ainda' modifies the verb

ser (75% : 0.064 | 25% : 0.042) ⋮ haver (55% : 0.061 | 45% : 0.073) ⋮ dizer (11% : 0.007 | 89% : 0.069) ⋮

'*' is on the right of 'sempre/ainda'

'sempre/ainda' modifies the verb

defender (90% : 0.027 | 10% : 0.002) ⋮ dizer (78% : 0.031 | 22% : 0.011) ⋮ estar (26% : 0.012 | 74% : 0.05) ⋮ haver (20% : 0.007 | 80% : 0.048) ⋮

'sempre/ainda' modifies the adjective

pronto (94% : 0.049 | 6% : 0.005) ⋮ presente (88% : 0.093 | 12% : 0.022) ⋮ grande (23% : 0.014 | 77% : 0.091) ⋮ desconhecido (4% : 0.001 | 96% : 0.046) ⋮

'sempre/ainda' is modified by the adverb

assim (100% : 0.133 | 0% : 0) ⋮ nem (32% : 0.007 | 68% : 0.036) ⋮ não (13% : 0.01 | 87% : 0.173) ⋮

'sempre/ainda' modifies the adverb

muito (63% : 0.018 | 37% : 0.024) ⋮ bem (54% : 0.026 | 46% : 0.047) ⋮ mais (14% : 0.012 | 86% : 0.174) ⋮

Figure 3.10: Word comparison between adverbs *sempre* 'always' and *ainda* 'yet' in the CETEMPúblico *corpus*.

Figure 3.11 represents the second type of comparison implemented in this project, the *Corpora Comparison* where a single lemma is compared in two different *corpora*. What differentiates a *Word Comparison* from a *Corpora Comparison*, besides the results given, is only the header. A comparison of a lemma in two different *corpora* will have the following components on its header:

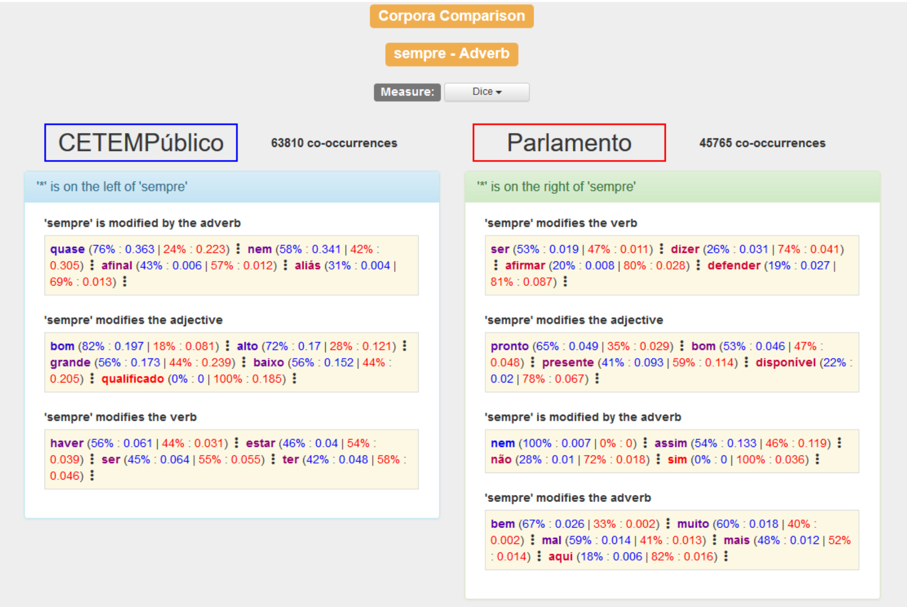
- The type of comparison, in this case being a *Corpora Comparsion*;

- The lemma that is being compared in the different *corpora*;

- A drop-down menu indicating the measure being used and allowing the user to change it;

- On the left side, enveloped by a blue rectangle is the first *corpus* selected by the user and, at its' right, the number of co-occurrences of the chosen lemma in this *corpus*, present in the database;

- On the right side, enveloped by a red rectangle is the second *corpus* selected by the user and, at its' right, the amount of co-occurrences of the chosen lemma in this *corpus*, present in the database;



Corpora Comparison

sempre - Adverb

Measure: Dice ▾

CETEMPúblico 63810 co-occurrences

'*' is on the left of 'sempre'

'sempre' is modified by the adverb

quase (76% : 0.363 | 24% : 0.223) ⋮ nem (58% : 0.341 | 42% : 0.305) ⋮ afinal (43% : 0.006 | 57% : 0.012) ⋮ aliás (31% : 0.004 | 69% : 0.013) ⋮

'sempre' modifies the adjective

bom (82% : 0.197 | 18% : 0.081) ⋮ alto (72% : 0.17 | 28% : 0.121) ⋮ grande (56% : 0.173 | 44% : 0.239) ⋮ baixo (56% : 0.152 | 44% : 0.205) ⋮ qualificado (0% : 0 | 100% : 0.185) ⋮

'sempre' modifies the verb

haver (56% : 0.061 | 44% : 0.031) ⋮ estar (46% : 0.04 | 54% : 0.039) ⋮ ser (45% : 0.064 | 55% : 0.055) ⋮ ter (42% : 0.048 | 58% : 0.046) ⋮

Parlamento 45765 co-occurrences

'*' is on the right of 'sempre'

'sempre' modifies the verb

ser (53% : 0.019 | 47% : 0.011) ⋮ dizer (26% : 0.031 | 74% : 0.041) ⋮ afirmar (20% : 0.008 | 80% : 0.028) ⋮ defender (19% : 0.027 | 81% : 0.087) ⋮

'sempre' modifies the adjective

pronto (65% : 0.049 | 35% : 0.029) ⋮ bom (53% : 0.046 | 47% : 0.048) ⋮ presente (41% : 0.093 | 59% : 0.114) ⋮ disponível (22% : 0.02 | 78% : 0.067) ⋮

'sempre' is modified by the adverb

nem (100% : 0.007 | 0% : 0) ⋮ assim (54% : 0.133 | 46% : 0.119) ⋮ não (28% : 0.01 | 72% : 0.018) ⋮ sim (0% : 0 | 100% : 0.036) ⋮

'sempre' modifies the adverb

bem (67% : 0.026 | 33% : 0.002) ⋮ muito (60% : 0.018 | 40% : 0.002) ⋮ mal (59% : 0.014 | 41% : 0.013) ⋮ mais (48% : 0.012 | 52% : 0.014) ⋮ aqui (18% : 0.006 | 82% : 0.016) ⋮

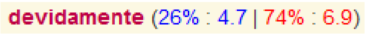
Figure 3.11: Comparison of the word *sempre* 'always' in the CETEMPúblico and Parlamento *corpora*.

Figure 3.12 is a close-up of a co-occurrence result. On the left is the co-occurrent word, and on the right, in parenthesis, are four values. The two values on the left and colored in blue concern the lemma or *corpus* in the blue rectangle, and the two values on the right and colored in red concern the lemma or *corpus* in the red rectangle. The first value is a percentage score that is calculated using the expression below:

$$\frac{\frac{fb_{cooc}}{fb_{totalcooc}}}{\frac{fb_{cooc}}{fb_{totalcooc}} + \frac{fr_{cooc}}{fr_{totalcooc}}} \times 100 \tag{3.1}$$

where $\frac{fb_{cooc}}{fb_{totalcooc}}$ is the relative frequency of the co-occurrence between the lemma in the blue rectangle and the co-occurrent lemma and $\frac{fb_{cooc}}{fb_{totalcooc}}$ is the relative frequency of the co-occurrence between the lemma in the red rectangle and the co-occurrent lemma. This expression aims to form a ratio between these two relative frequencies, in order to show the user which co-relation is quantitatively more important, while also taking into account the differences in the total number of occurrences of the two different lemmas, or the same lemma in two different *corpora*.

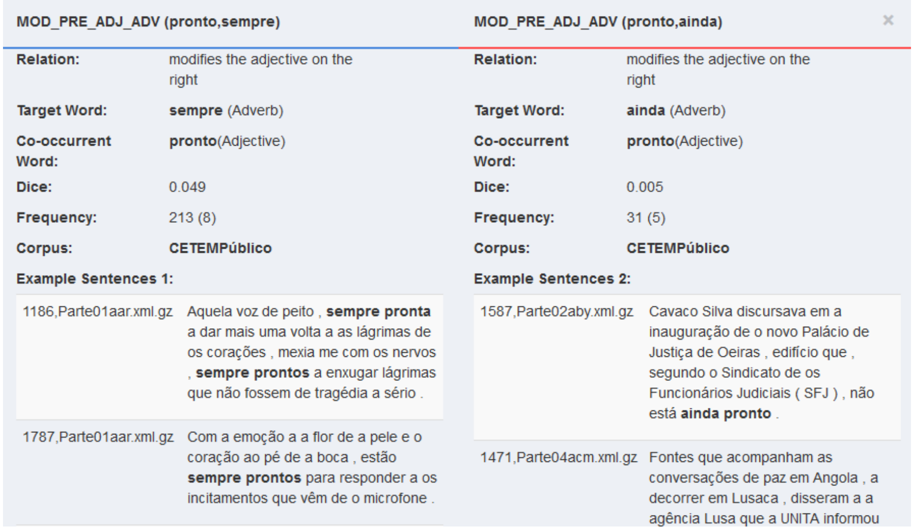
The second value is the selected measure for the co-occurrence, similarly to the search result screen.



devidamente (26% : 4.7 | 74% : 6.9)

Figure 3.12: Co-occurrence result.

Clicking on one of the co-occurrences on the comparison result screen generates a popup with a detailed view of that co-occurrence with additional information and example sentences like in the search result screen. Figure 3.13 illustrates an example of a detailed view of a co-occurrence result in a comparison between two words (adjective *pronto* 'ready' and adverbs *sempre* 'always' and *ainda* 'yet'). This window is divided into two parts. The left part corresponds to the first element of the comparison, present in the blue rectangle on Figure 3.2. The right part corresponds to the second element of the comparison, present in the red rectangle. The choice of colors is used once again on the lines beneath the title to indicate which part of the screen belongs to each element of the comparison.



MOD_PRE_ADJ_ADV (pronto,sempre)

Relation:	modifies the adjective on the right
Target Word:	**sempre** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.049
Frequency:	213 (8)
Corpus:	**CETEMPúblico**

Example Sentences 1:

1186,Parte01aar.xml.gz	Aquela voz de peito , **sempre pronta** a dar mais uma volta a as lágrimas de os corações , mexia me com os nervos , **sempre prontos** a enxugar lágrimas que não fossem de tragédia a sério .
1787,Parte01aar.xml.gz	Com a emoção a a flor de a pele e o coração ao pé de a boca , estão **sempre prontos** para responder a os incitamentos que vêm de o microfone .

MOD_PRE_ADJ_ADV (pronto,ainda)

Relation:	modifies the adjective on the right
Target Word:	**ainda** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.005
Frequency:	31 (5)
Corpus:	**CETEMPúblico**

Example Sentences 2:

1587,Parte02aby.xml.gz	Cavaco Silva discursava em a inauguração de o novo Palácio de Justiça de Oeiras , edifício que , segundo o Sindicato de os Funcionários Judiciais (SFJ) , não está **ainda pronto** .
1471,Parte04acm.xml.gz	Fontes que acompanham as conversações de paz em Angola , a decorrer em Lusaca , disseram a a agência Lusa que a UNITA informou

Figure 3.13: Detailed view of a co-occurrence comparison.

Finally, the Help[2] and About[3] pages of the Explorer were updated to reflect the changes made on the system. To the Help page information was added about the comparison feature, the word highlighting algorithm and the changes to the main screen interface. On the About page some examples were added to the XIP syntactic dependencies, as well as a brief description of the available *corpora*. These pages have also been added as appendices at the end of this document.

3.2.3 Word highlighting in example sentences

Another feature that deserves attention is the word highlighting in example sentences (Figure 3.13). Highlighting the target and co-occurrent words allows for a faster inspection of the example sentences. The original version of the Explorer did not display this feature. Highlighting a word on the original version of the Explorer is not a trivial task since there were various setbacks and several different ways to develop this feature. This subsection describes the approach taken to highlight the searched word and the co-occurrent word in the example sentences.

The biggest hurdle when highlighting these words is the fact that the system registers only the lemmas present in the co-occurrence while the example sentences may contain an inflected form of those lemmas. So, the challenge consisted in selecting the inflected form of a lemma from a string while only having the lemma to work with. The developed solution tries to match the inflected form to its lemma by using a regular expression. This regular expression removes the last letter from the lemma and is used to test every word in the example sentence for a match, if a match occurs that word is highlighted. So if we have the lemma *funcionário* 'employee' the corresponding regular expression will be **funcionári*** and, for example, will highlight words such as *funcionário*, *funcionária*, *funcionários* and *funcionárias*.

[2] https://string.hlt.inesc-id.pt/demo/deepExplorer/#/help (last visited in 22 of April 2020).

[3] https://string.hlt.inesc-id.pt/demo/deepExplorer/#/about (last visited in 22 of April 2020).

This solution has two major flaws. It is possible to highlight words other than the target word and co-occurrent word, and it is also possible not to highlight the co-occurrent word or the target word (or both). For example, when searching for the lemma *ser* 'to be' , the corresponding regular expression will be **se*** which will match correctly to words like *ser*, *seja* or *será*. However it will miss words such as *foi*, *era* or *é* which are all conjugations of the verb *ser* and match with words such as *serra* 'saw' , *semente* 'seed' or *sempre* 'always' .

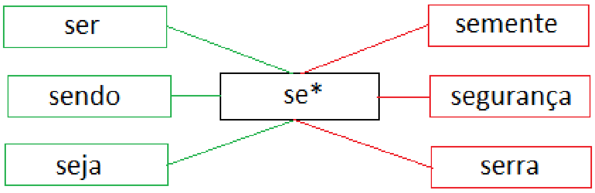


Figure 3.14: Example of the highlighting algorithm using the lemma *ser* 'to be' .

The accuracy of this algorithm and the impact it has on the usability of the system is evaluated in Section 4.1.

Despite its issues, this was the algorithm chosen in this project to highlight co-occurrent words in example sentences. It may also be possible to store the exact position of each word in the example sentences in order to always identify correctly the words to highlight. However, managing the changes done to the database and minimizing the increase in database size was not on the scope of this dissertation project and it would likely be complex enough to justify another dissertation.

Chapter 4

Evaluation

In this chapter, the evaluation method of the newly developed features is described as well as the obtained results. Section 4.1 evaluates the performance of the word highlighting algorithm presented in Subsection 3.2.3, and Section 4.2 evaluates the performance of the web application, described in Subsection 3.2.2.

4.1 Word highlighting in example sentences

This functionality was subjected to two different tests. The first test measured the accuracy of the word highlighting algorithm and the percentage of times a word was incorrectly highlighted in the test group. The second test was a user test with the goal of measuring the impact of this functionality on the user experience.

The Explorer allows the search for 4 different POS, and so the test sample consists of the **10 Nouns, Verbs, Adjectives and Adverbs with the largest number of co-occurrences (ignoring named entities).** Table 4.1 represents the chosen words and their co-occurrence frequency.

Nouns	Verbs	Adjectives	Adverbs
país (122,202)	*ser* (1,412,972)	*grande* (173,464)	*não* (836,293)
cento (121,185)	*ter* (749,747)	*novo* (141,588)	*já* (215,513)
empresa (107,251)	*fazer* (488,665)	*português* (86,778)	*mais* (211,902)
pessoa (100,393)	*haver* (382,119)	*bom* (83,305)	*ainda* (169,149)
problema (99,799)	*estar* (327,799)	*último* (52,344)	*também* (157,391)
situação (92,663)	*dar* (302,478)	*pequeno* (41,836)	*muito* (129,727)
projeto (91,589)	*dizer* (279,171)	*principal* (40,293)	*agora* (82,221)
parte (87,818)	*ir* (199,954)	*atual* (37,668)	*bem* (78,839)
caso (85,342)	*passar* (172,564)	*europeu* (34,164)	*assim* (72,384)
processo (84,593)	*ver* (163,623)	*antigo* (33,429)	*sempre* (63,810)

Table 4.1: The most frequently co-occurrent lemmas in the *corpus* CETEMPúblico (and their frequency).

An Explorer search was made for every lemma in the test sample with a *Maximum number of co-occurrences to display* set to 5 and selecting the CETEMPúblico *corpus* with the Dice metric. For every co-occurrence resulting from the search, the first example sentence was extracted. Since each POS may enter in different dependency relations, the resulting number of extracted sentences was different for each one. This testing sample consisted of:

- 476 sentences where the target word is a **Noun**
- 358 sentences where the target word is a **Verb**
- 220 sentences where the target word is an **Adjective**
- 310 sentences where the target word is an **Adverb**

The results were assessed as follows: each sentence gains one accuracy point when either the target word or the co-occurrent word is highlighted; and two accuracy points were given when both words are highlighted; conversly no accuracy points were given when the algorithm failed to highlight either word. The *Accuracy* row on Table 4.2 represents the amount of points each POS had divided by the maximum number of possible points (two times the number of sentences for each POS). The *Error rate* row represents the number of sentences where one or more words, other than the target word or the co-occurrent word, have been highlighted, divided by the total number of sentences for each POS.

	Nouns	**Verbs**	**Adjectives**	**Adverbs**
Accuracy	85%	48%	67%	87%
Error rate	11%	18%	11%	21%

Table 4.2: Evaluation of the highlighting algorithm.

When analysing the results, certain factors that lower the accuracy of the algorithm became evident, namely:

- **The 1990 Portuguese spelling reform**, for most of the corpora were written in a pre-reform spelling. In this new spelling, silent etymological consonants were removed from a few thousand frequent words, so that currently spelled lemmas in the lexical resources of the Explorer can no longer be matched to the inflected surface forms in the text. The algorithm fails sometimes because it tries to match a word with the modern spelling to one with the old spelling (*projeto/**projecto**, ação/**acção**, infetado/**infectado**, etc.*) ;
- **Named entities.** The algorithm fails to match a named entity (TDATA, PESSOA, LOC) to the word that appears on the actual sentence (*ontem*, *Ricardo*, *Lisboa*);
- **N-grams with size 2 or more.** The algorithm matches the regular expression pattern extracted from the target and co-occurrent lemmas to every word. Since the algorithm splits each sentence into individual words, this regular expression finds no matches for n-grams with n>1 (*no entanto*, *mais de*, *etc.*) ;

- **Verbs with irregular forms.** The conjugated form of these verbs usually differs a lot from its lemma and the algorithm is unable to make the connection (*ir/vai*, *ser/é*, etc.);

- **Adjectives with irregular forms.** Similarly to the previous factor, the algorithm cannot match the comparative form of a small set of adjectives to its lemma form (*grande/maior*, *bom/melhor*, *pequeno/menor*).

The incorrect highlighting occurred mainly with shorter lemmas (*ser*, *ter*, *ir*, *já*) since the regular expression was more forgiving, for example the lemma *ir* matched with every word starting with an 'i'.

Table 4.3 represents the average amount of time a user takes to find the target word and the co-occurrent word in example sentences. This test was done using the example sentences in the sample, both with the word highlighting algorithm and with no highlighting, (as a control). Since the word highlighting algorithm has a varying performance depending on the POS of the words, as it has been seen on the previous test, the words to test contain examples from the four different POS. In addition to this, the example sentences were divided into short sentences and long sentences, to determine how the length of the sentence affects the results. Short sentences have no more than 3 lines while long sentences have 4 or more lines. In the example sentences, one line has approximately 9 words.

In this test, each person would have to point to both the target and the co-occurrent word in 48 sentences divided into 2 sets of sentences named Group A and Group B. In Group A, none of the sentences were highlighted and in Group B all the sentences were highlighted following the word highlighting algorithm. Some sentences in Group B contained incorrect highlighting and missing highlighted words to better simulate the functioning of the highlighting algorithm. These example sentences were taken from the Explorer's interface.

From the 24 sentences that each group had, 12 were short sentences while the remaining 12 were long sentences. These sentences were divided equally by the POS available to the Explorer. Each POS appeared as a target word in 6 different sentences. The full list of sentences is available as an appendix at the end of this document.

This test was presented to 10 people: half of them first processed Group A, followed by Group B and the other half first processed Group B followed by Group A. Each user was given oral instructions and a first example to practice. Table 4.3 presents the results from the user tests. These people were selected from a group of my acquaintances who had no previous knowledge of the goal of this dissertation. This test only required the users to be able to read, the sample group was chosen by convenience only.

	Noun		**Verb**		**Adjective**		**Adverb**	
Sentence Length	short	long	short	long	short	long	short	long
Lookup Time of Group A (s)	4.24	6.13	3.45	7.04	3.56	5.32	4.2	9.37
Lookup Time of Group B (s)	2.31	2.67	2.3	2.62	2.58	2.88	3	2.76
Lookup time improvement (s)	1.93	3.46	1.15	4.42	0.99	2.44	1.2	6.61
Lookup time improvement (%)	45.5	56.4	33.3	62.8	27.8	45.9	28.6	70.5

Table 4.3: User test of the highlighting algorithm.

Despite the imperfect accuracy and the incorrect highlightings of the algorithm, the lookup time improved in every situation, on average, 46%. The most notable changes occur in longer sentences as this algorithm allows the user to skim through the text more easily while looking for the selected co-occurrence.

The users also commented that the algorithm made larger sentences less intimidating, even when multiple incorrect highlightings occurred.

4.2 Web application

This section will contrast the response time between the original Explorer with the one modified in this project as well as measure the time needed to execute a comparison. Each search was executed on a personal laptop, with an Intel Core i5-8265U CPU, 8 Gb of RAM and running an Apache server locally on a SSD storage device (this is a fairly standard configuration). These tests were performed locally in order to mitigate the impact of internet issues in the results.

4.2.1 Search performance

To evaluate the time needed to execute a search, two lemmas from each part of speech were selected. One of those lemmas was randomly selected from the 10 most frequently co-occurrent lemmas for that part of speech, and the other one was randomly selected from the 10 less frequently co-occurrent lemmas but, with more than 1000 co-occurrences for that part of speech. These searches were made using the default settings and using the same *corpus*, CETEMPúblico.

This method of analysis was inspired by the evaluation of the original version of the Explorer [6] and it was chosen because it takes into account the disparity in co-occurrence frequency (more frequent results will take longer to process) as well as disparity in information available for different parts of speech. Table 4.4 shows the results of the search evaluation. Each search was executed 3 times and the average time was inserted into this table. The response time was measured using the *Network Monitor* present in the *Mozilla Firefox* web browser version 75.0 (64-bit). The database used for both systems was the same and both systems were deployed locally.

	Noun		**Verb**		**Adjective**		**Adverb**	
Word	pessoa	astronauta	fazer	desferir	antigo	rígido	mais	seriamente
Explorer v.1 Time (ms)	95	11	463	13	65	10	122	15
Explorer v.2 Time (ms)	153	13	473	13	72	9	131	17
Response time difference (%)	61.1	18.2	2.2	0	10.8	-10	7.4	13.3

Table 4.4: Search performance evaluation.

As expected, the changes made to the system slightly increased the response time. In every POS there is now an extra SQL query for the total number of co-occurrences of the target lemma. In addition to this, the noun and verb result screens have additional dependency-property patterns, which results in more queries for those particular POS, thus increasing the response time even further.

4.2.2 Comparison performance evaluation

To execute a similar evaluation as the one described on subsection 4.2.1, four lemmas from each part of speech were needed instead, two were randomly selected from the 10 most frequent co-occurrent lemmas for that part of speech and the other two from the least frequent co-occurrent lemmas. Considering *most1* and *most2* as the two, randomly selected, high co-occurrent lemmas; and *least1* and *least2* as the two, randomly selected, low co-occurrent lemmas, the following comparisons are possible:

- Comparing a high co-occurrent lemma with another high co-occurrent lemma (comparing *most1* with *most2*); in this evaluation, this type of comparison will be named **High Frequency Comparison**;
- Comparing a high co-occurrent lemma with a low co-occurrent lemma (e.g comparing *most1* with *least1*), here named as **Mixed Frequency Comparison**;
- Comparing a low co-occurrent lemma with another low co-occurrent lemma (comparing *least1* with *least2*), here named as **Low Frequency Comparison**.

Again, the *corpus* used in these comparisons was CETEMPúblico. Each comparison was executed 3 times and the average time is presented in Table 4.5. The response time was measured in the same way as in the previous evaluation.

	Noun			**Verb**			**Adverb**			**Adjective**		
Comparison	High	Mixed	Low	High	Mixed	Low	High	Mixed	Low	High	Mixed	Low
Time (ms)	720	319	32	1850	870	36	576	384	35	312	271	29

Table 4.5: Word Comparison performance evaluation.

As it was expected, the response times increase drastically with the frequency of the lemmas. The verbs used in the High Frequency Comparison presented a considerable higher frequency than the other POS, which resulted in the impressive difference in response time. When comparing words with low frequencies the response time is similar in every POS.

Table 4.6 compares the difference in response time in a Corpora Comparison. Each pair of *corpora* is tested with the same words to better understand the difference in response time under the same conditions. The 8 lemmas used in this table represent an example of high co-occurrent and low co-occurrent lemmas for every POS.

Each comparison was executed 3 times and the average time is shown in Table 4.6. The response time was measured in the same way as in the previous evaluations.

	CETEMPúblico and Desportivo	**CETEMPúblico and Parlamento**	**Parlamento and Desportivo**
pessoa	994	1,120	714
astronauta	63	87	71
fazer	4,146	4,290	2,870
desferir	65	52	46
antigo	398	409	156
rígido	47	57	35
mais	764	826	675
seriamente	57	56	42

Table 4.6: Time (in ms) to execute a *corpora* comparison for different words.

The response time in the first two columns is very similar since the Desportivo and Parlamento *corpora* have similar sizes. The main difference comes between the first two columns and the third one. Since the CETEMPúblico *corpus* is twice as large as Desportivo and Parlamento, comparisons featuring CETEMPúblico are expected to take a longer time than comparisons that do not involve this corpus.

Chapter 5

Conclusions

This dissertation covered some improvements to the *corpora* analysis tool DeepString - Syntax Deep Explorer. The main added features were the support for multiple *corpora*; the comparison between words within the same *corpus*, and the comparison of the same word accross two different *corpora*; and the algorithm for highlighting target words in the example sentences.

To add support for multiple *corpora*, the database was changed in order to be more modular and facilitate the addition of new *corpora*, as well as changing or removing existing *corpora*. This also aims at improving the response time of the SQL queries, should the database grow to a much more considerable size.

Two new *corpora* (Desportivo and Parlamento) were added to the system, one of which had to be built almost from scratch before being processed by STRING. The CETEMPúblico corpus has also been processed again with a more recent version of the XIP Portuguese Grammar.

The comparison feature took inspiration from Sketch Engine's comparison, but expanded upon this concept by allowing different association measures to be compared, and by allowing the user to compare a lemma in two different *corpora*.

The word highlighting algorithm, although still rudimentary, proved to be a helpful feature, drastically reducing the time that a user needs to read the example sentences, even when not providing the most accurate results. The changes applied to the interface also improved the consistency of the placement of dependencies and reduced the ambiguity found in certain cases, all of which resulted in an overall improvement of the user experience.

5.1 Future work

This section aims to introduce a few ideas on how this system can be improved.

Someone looking to improve the Explorer can utilize cookies in order to maintain the user's previous settings when searching for another lemma. This would reduce some of the attrition when making consecutive searches, which is often the case. Due to time constraints this feature was not implemented.

The word highlighting algorithm could also be improved. This feature would probably require a change to the way the database is maintained in order to save which words serve as example in every example sentence and, therefore, was not implemented. Managing the size and structure of the database is not a trivial task and it could be an entirely different dissertation.

The comparison feature could also be expanded upon, by implementing several new types of comparison such as comparing different lemmas in different *corpora*, comparing more than two different lemmas or comparing the same lemma in more than two *corpora*. This would most likely require several changes to the interface and in the way a comparison is executed.

One problem that, to this date, has not been solved by STRING is the PP-attachment problem. This problem is a result of the structural ambiguity raised by prepositional phrases as to the exact word they are modifying. Currently, this problem is being treated by STRING with a simple heuristic that could be made more accurate. However, if that problem is eventually solved, displaying which prepositions introduce either the target lemma or co-occurrent lemmas (a 3 word co-occurrence) could be an interesting development of the Explorer, eventually improving the PP-attachment solutions.

There is also room for improvement on the selection of POS that the Explorer has to offer. The addition of prepositions (*contra* 'against' , *com* 'with' , etc.) and conjunctions (*porque* 'because' , *se* 'if' , etc.) could be implemented in the next iteration of the Explorer. Searching for prepositions, specifically, would be interesting if the previous feature (prepositions that introduce either the target lemma or co-occurrent lemmas) was implemented.

Another interesting feature would be the implementation of *thesauri* for a given lemma. With the data retrieved from the co-occurrences in a *corpus*, it would be possible to determine which lemmas are used in a similar way as the target lemma, on the selected *corpus*. For example, in the *corpus* Desportivo, the word *golo* 'goal' could have as synonyms the words *ponto* 'point' or *bola* 'ball' based on the way that these words are used in the context of this *corpus*.

Finally, a major and challenging feature would be allowing the users to submit their own *corpora* to be processed by STRING and later be analyzed by the Explorer. Taking inspiration from Sketch Engine, a user could upload his or her *corpora* in a raw format, let it be processed by STRING, and then use it with the Explorer; or, else, submit a *corpus* that had already been processed by STRING. Along with uploading his or her own files, a *corpora* could also be generated from websites or from Google search results, given a set of seed words. When developing this feature, special care must be given to eventual security and copyright issues, as well as managing disk space, if the *corpora* are to be stored in the server.

5.2 Contributions

This document concludes with a list of the main contributions made within the scope of this dissertation:

- Constitution of the *corpus* Desportivo;
- Changes in the database populating script, in order to support multiple *corpora*;
- Addition of two new *corpora* to the Syntax Deep Explorer system, the Desportivo and the Parlamento *corpora*;

- Implementation of two types of comparison: comparison of 2 words within the same *corpus*, and of the same word across 2 *corpora*;
- Implementation of an algorithm for word highlighting in example sentences, making it easier for the user to find the target and the co-occurrent words in every example sentence;
- Improving the user interface and user experience with the DeepString - Syntax Deep Explorer system.

The new system has been made available to the public through the INESC-ID website of STRING and it has been in operation since 29 of April 2020.

References

[1] AÏT-MOKHTAR, SALAH; CHANOD, J-P & ROUX, CLAUDE. 2002. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, **8**(2-3), 121–144.

[2] BAPTISTA, JORGE & MAMEDE, NUNO. 2016. *Nomenclature of chunks and dependencies in Portuguese XIP Grammar 4.6*. Technical Report. L2F-Spoken Language Laboratory, INESC-ID Lisboa, Lisboa.

[3] BARONI, MARCO; KILGARRIFF, ADAM; POMIKÁLEK, JAN & RYCHLỲ, PAVEL. 2006. WebBootCaT: a web tool for instant corpora. *Pages 123–132 of: Proceeding of the EuraLex Conference*.

[4] BICK, ECKHARD. 2009. DeepDict – A Graphical Corpus-based Dictionary of Word Relations. *Pages 268–271 of: Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*. Odense, Denmark: Northern European Association for Language Technology (NEALT).

[5] CARAPINHA, FILIPE. 2013. *Extração automática de conteúdos documentais*. Master thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

[6] CORREIA, JOSÉ; BAPTISTA, JORGE & MAMEDE, NUNO. 2016. Syntax Deep Explorer. *Pages 189–201 of: Computational Processing of the Portuguese Language: 12th International Conference, PROPOR 2016, Tomar, Portugal, July 13-15, 2016, Proceedings*. LNAI/LNCS, vol. 9727. Cham: Springer.

[7] DINIZ, CLÁUDIO; MAMEDE, NUNO & PEREIRA, JOÃO D. 2010. RuDriCo2 – A Faster Disambiguator and Segmentation Modifier. *II Simpósio de Informática (INForum), Universidade do Minho*, 573–584.

[8] EVERT, STEFAN & HARDIE, ANDREW. 2011. Twenty-first Century Corpus Workbench: Updating a Query Architecture for the New Millennium. *Presentation at Corpus Linguistics 2011, University of Birmingham*.

[9] HARDIE, ANDREW. 2012. CQPweb — Combining Power, Flexibility and Usability in a Corpus Analysis Tool. *International Journal of Corpus Linguistics*, **17**(3), 380–409.

[10] KILGARRIFF, ADAM; BAISA, VÍT; BUŠTA, JAN; JAKUBÍČEK, MILOŠ; KOVÁŘ, VOJTĚCH; MICHELFEIT, JAN; RYCHLỲ, PAVEL & SUCHOMEL, VÍT. 2014. The Sketch Engine: ten years on. *Lexicography*, **1**(1), 7–36.

[11] MAMEDE, NUNO; BAPTISTA, JORGE; DINIZ, CLÁUDIO & CABARRÃO, VERA. 2012. STRING - A Hybrid, Statistical and Rule-Based, Natural Language Processing Chain for Portuguese. *In: Computational Processing of the Portuguese Language - PROPOR 2012*. PROPOR.

[12] Mendes, Amália; Généreux, Michel; Hendrickx, Iris; Pereira, Luísa; do Nascimento, Maria Fernanda Bacelar & Antunes, Sandra. 2012. CQPWeb: Uma nova plataforma de pesquisa para o CRPC. *A. Costa, C. Flores, & N. Alexandre, (Eds.) Textos Selecionados do XXVII Encontro Nacional da Associação Portuguesa de Linguística*, 466–477.

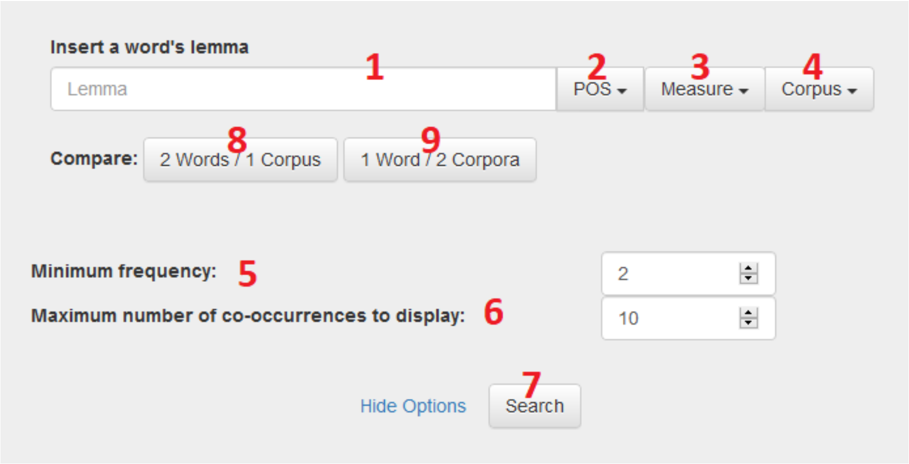
[13] O'Donnell, Matthew Brook; Hoffmann, Sebastian; Evert, Stefan; Smith, Nicholas; Lee, David & Berglund-Prytz, Ylva. 2008. *Corpus linguistics with BNCweb – A Practical Guide*. Vol. 6. Bern, Switzerland: Peter Lang.

[14] Ribeiro, Ricardo. 2003. *Anotação morfossintáctica desambiguada do português*. Master thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

[15] Vicente, Alexandre. 2013. *LexMan: um Segmentador e Analisador Morfológico com transdutores*. Master thesis, Instituto Superior Técnico-Universidade Técnica de Lisboa, Portugal.

Appendix A

Help page



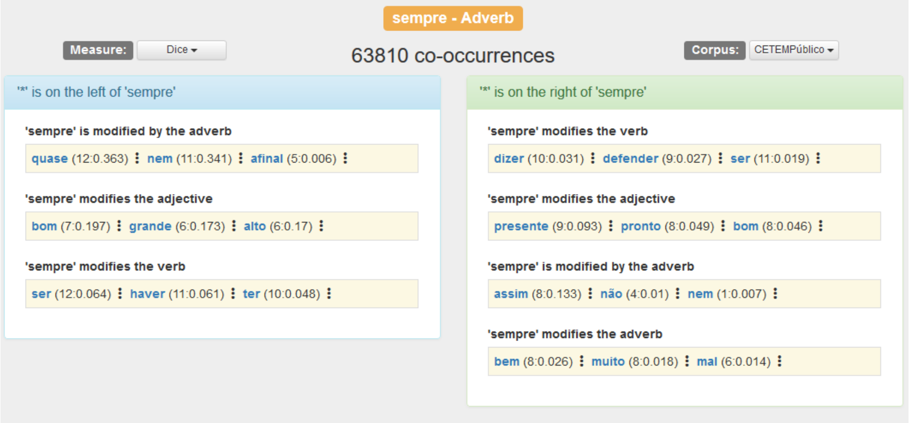
Insert a word's lemma

Minimum frequency: 2

Maximum number of co-occurrences to display: 10

In this form the user inserts the **word's lemma (1)**, and choses its **POS (2)**, the **association measure (3)** to be calculated and the **Corpus (4)** where the search will be performed. However, the user has additional options like the **minimum frequency (5)** of each co-occurrent element and the **maximum number of co-occurrences to display (6)** in each dependency co-occurrence pattern. By default, these are set to 2 and 10, respectively. Press the button **Search (7)** to show the results for the selected lemma and the chosen set of options (example below).

Buttons **8** and **9** enable the comparison feature (explained below).



sempre - Adverb

Measure: Dice

63810 co-occurrences

Corpus: CETEMPúblico

'*' is on the left of 'sempre'

'sempre' is modified by the adverb

quase (12:0.363) ⋮ nem (11:0.341) ⋮ afinal (5:0.006) ⋮

'sempre' modifies the adjective

bom (7:0.197) ⋮ grande (6:0.173) ⋮ alto (6:0.17) ⋮

'sempre' modifies the verb

ser (12:0.064) ⋮ haver (11:0.061) ⋮ ter (10:0.048) ⋮

'*' is on the right of 'sempre'

'sempre' modifies the verb

dizer (10:0.031) ⋮ defender (9:0.027) ⋮ ser (11:0.019) ⋮

'sempre' modifies the adjective

presente (9:0.093) ⋮ pronto (8:0.049) ⋮ bom (8:0.046) ⋮

'sempre' is modified by the adverb

assim (8:0.133) ⋮ não (4:0.01) ⋮ nem (1:0.007) ⋮

'sempre' modifies the adverb

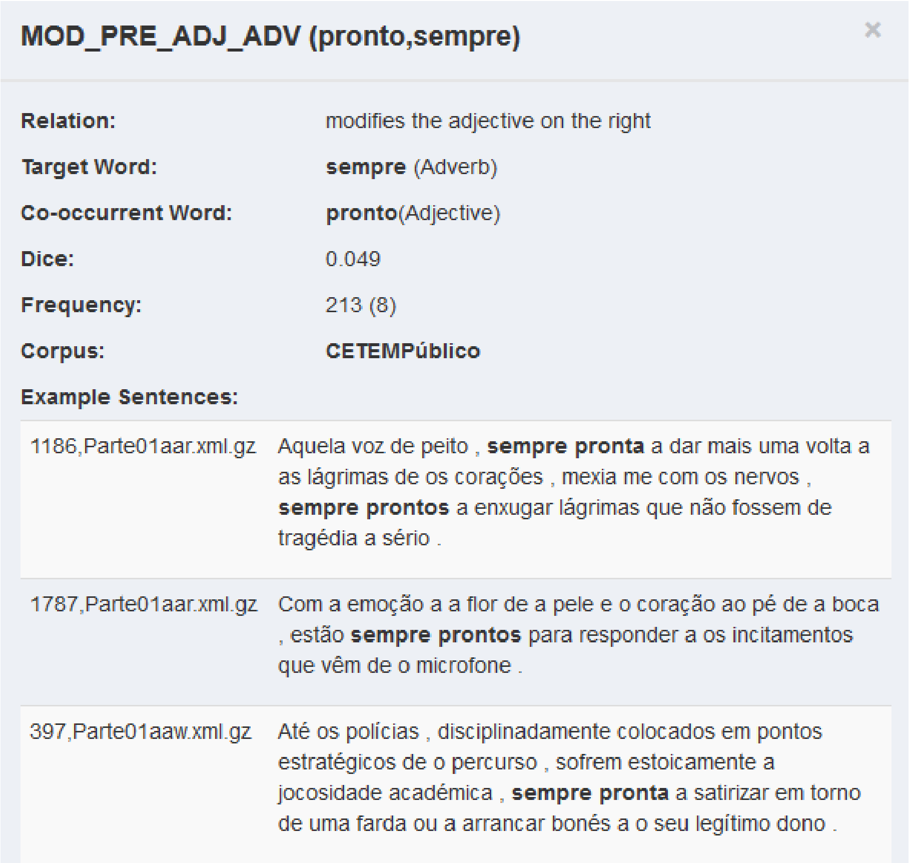
bem (8:0.026) ⋮ muito (8:0.018) ⋮ mal (6:0.014) ⋮

The image above shows an example of a lex-gram with a set of co-occurrences detected for the **adverb 'sempre'** in the **CETEMPúblico** corpus. The association measure selected for this result was **Dice**, the maximum number of co-occurrences to display was **3** and the minimum frequency was **2**.

On top-left of the page it is possible to change the association measure and on top-right the corpus of the search. Below the target lemma the number of co-occurrences of this lemma in the selected corpus is shown. For each dependency-property pattern in which the target word is involved, the words that co-occur with the target word are presented in descending order of the selected association measure score. Each word that co-occurs is displayed in the format **lemma (l:m)**, where **l** is the base 2 logarithm of the co-occurrence's frequency, and **m** the value of the selected association measure. The user may change the association measure without having to re-entering the information about the current word.

In the lex-gram, each dependency-property pattern is placed in relation to the target lemma. Patterns on the blue grid appear on the left of the target lemma and patterns on the green grid appear on the right of the target lemma. In order to avoid confusion, the title of each grid changes when a co-occurrent lemma is hovered. In this example, the co-occurrent lemma hovered is **'pronto'** so the title of the grid changed to **'pronto' is on the right of 'sempre'**.

From the lex-gram, it is possible to get more information about a specific co-occurrence (see below). By clicking on any word, the details of this co-occurrence are shown and a set of sentences that exemplifies that co-occurrence in the corpus is shown. In each example sentence, both the target word and the co-occurrent can be highlighted.



MOD_PRE_ADJ_ADV (pronto,sempre)

Relation:	modifies the adjective on the right
Target Word:	**sempre** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.049
Frequency:	213 (8)
Corpus:	**CETEMPúblico**

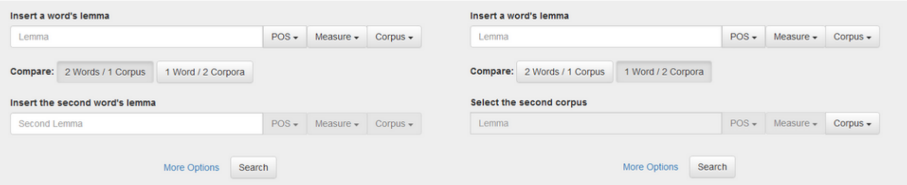
Example Sentences:

1186,Parte01aar.xml.gz	Aquela voz de peito , **sempre pronta** a dar mais uma volta a as lágrimas de os corações , mexia me com os nervos , **sempre prontos** a enxugar lágrimas que não fossem de tragédia a sério .
1787,Parte01aar.xml.gz	Com a emoção a a flor de a pele e o coração ao pé de a boca , estão **sempre prontos** para responder a os incitamentos que vêm de o microfone .
397,Parte01aaw.xml.gz	Até os polícias , disciplinadamente colocados em pontos estratégicos de o percurso , sofrem estoicamente a jocosidade académica , **sempre pronta** a satirizar em torno de uma farda ou a arrancar bonés a o seu legítimo dono .

In the main menu there is also the possibility to execute a comparison. Clicking on **2 Words / 1 Corpus (8)** will execute a word comparison, where 2 words are compared within the same corpus. Clicking on **1 Word / 2 Corpora (9)** will execute a corpora comparison, the same word in two different corpora. De-selecting any of these buttons will return the interface to its initial state.

When executing a word comparison, the user is required to insert the second word (the remaining options remain equal).

When executing a corpora comparison, the user is required to insert the second corpus (the remaining options remain equal).



Insert a word's lemma

Compare: 2 Words / 1 Corpus | 1 Word / 2 Corpora

Insert the second word's lemma

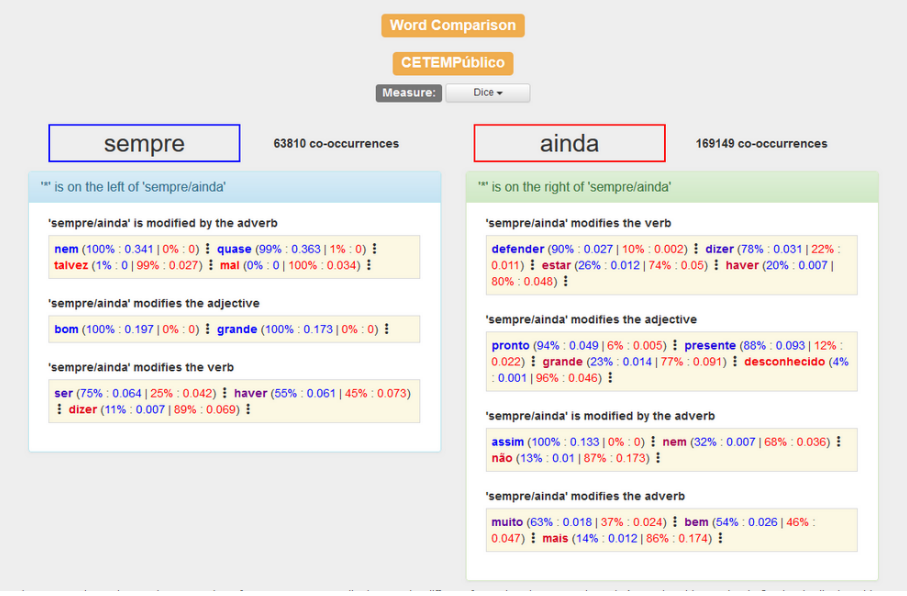
Insert a word's lemma

Compare: 2 Words / 1 Corpus | 1 Word / 2 Corpora

Select the second corpus

Both types of comparison behave similarly. The elements that are being compared appear in blue and red rectangles on top of the page. Each co-occurrence is followed by 4 values. The percentages represent a comparison between the co-occurrence relative frequency that are being compared (the color of the letters also changes based on this percentage) and the other values represent the selected association measure.

The image below depicts a word comparison example between the adverbs **'sempre'** and **'ainda'** in the **CETEMPúblico** corpus using **Dice** association measure with a minimum frequency of **2** and a maximum number of co-occurrences to display of **2**.



Word Comparison

CETEMPúblico

Measure: Dice

sempre — 63810 co-occurrences

ainda — 169149 co-occurrences

'*' is on the left of 'sempre/ainda'

'sempre/ainda' is modified by the adverb

nem (100% : 0.341 | 0% : 0) ⋮ quase (99% : 0.363 | 1% : 0) ⋮ talvez (1% : 0 | 99% : 0.027) ⋮ mal (0% : 0 | 100% : 0.034) ⋮

'sempre/ainda' modifies the adjective

bom (100% : 0.197 | 0% : 0) ⋮ grande (100% : 0.173 | 0% : 0) ⋮

'sempre/ainda' modifies the verb

ser (75% : 0.064 | 25% : 0.042) ⋮ haver (55% : 0.061 | 45% : 0.073) ⋮ dizer (11% : 0.007 | 89% : 0.069) ⋮

'*' is on the right of 'sempre/ainda'

'sempre/ainda' modifies the verb

defender (90% : 0.027 | 10% : 0.002) ⋮ dizer (78% : 0.031 | 22% : 0.011) ⋮ estar (26% : 0.012 | 74% : 0.05) ⋮ haver (20% : 0.007 | 80% : 0.048) ⋮

'sempre/ainda' modifies the adjective

pronto (94% : 0.049 | 6% : 0.005) ⋮ presente (88% : 0.093 | 12% : 0.022) ⋮ grande (23% : 0.014 | 77% : 0.091) ⋮ desconhecido (4% : 0.001 | 96% : 0.046) ⋮

'sempre/ainda' is modified by the adverb

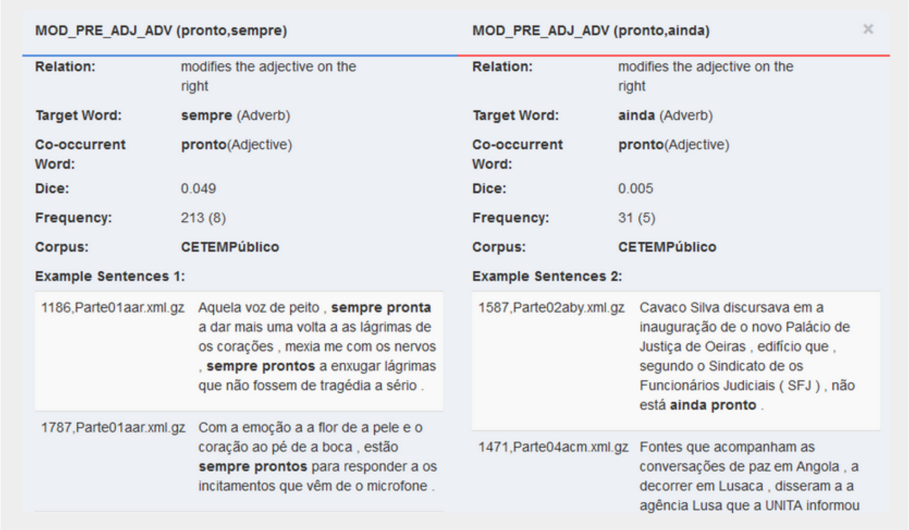
assim (100% : 0.133 | 0% : 0) ⋮ nem (32% : 0.007 | 68% : 0.036) ⋮ não (13% : 0.01 | 87% : 0.173) ⋮

'sempre/ainda' modifies the adverb

muito (63% : 0.018 | 37% : 0.024) ⋮ bem (54% : 0.026 | 46% : 0.047) ⋮ mais (14% : 0.012 | 86% : 0.174) ⋮

In a comparison, the maximum number of co-occurrences to display can be different from what the user selected. Assuming this number is 2, what is displayed is the top 2 co-occurrences with the highest measure for each lemma with duplicates being removed. So in this example, a dependency-property pattern can display up to 4 co-occurrences.

Similarly to the simple search, clicking on a co-occurrence opens a detailed view. However, in a comparison, this view is divided into two. On the left is the detailed view of the lemma (or corpus) in the blue rectangle and on the right is the detailed view of the lemma (or corpus) in red.



MOD_PRE_ADJ_ADV (pronto,sempre)

Relation:	modifies the adjective on the right
Target Word:	**sempre** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.049
Frequency:	213 (8)
Corpus:	**CETEMPúblico**

Example Sentences 1:

1186,Parte01aar.xml.gz	Aquela voz de peito , **sempre pronta** a dar mais uma volta a as lágrimas de os corações , mexia me com os nervos , **sempre prontos** a enxugar lágrimas que não fossem de tragédia a sério .
1787,Parte01aar.xml.gz	Com a emoção a a flor de a pele e o coração ao pé de a boca , estão **sempre prontos** para responder a os incitamentos que vêm de o microfone .

MOD_PRE_ADJ_ADV (pronto,ainda)

Relation:	modifies the adjective on the right
Target Word:	**ainda** (Adverb)
Co-occurrent Word:	**pronto**(Adjective)
Dice:	0.005
Frequency:	31 (5)
Corpus:	**CETEMPúblico**

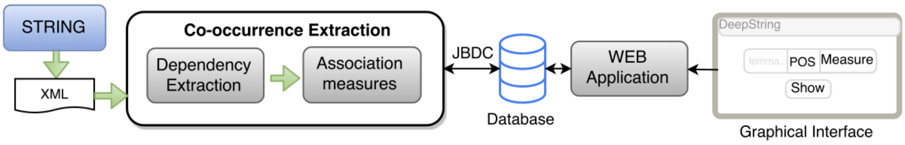
Example Sentences 2:

1587,Parte02aby.xml.gz	Cavaco Silva discursava em a inauguração de o novo Palácio de Justiça de Oeiras , edifício que , segundo o Sindicato de os Funcionários Judiciais (SFJ) , não está **ainda pronto** .
1471,Parte04acm.xml.gz	Fontes que acompanham as conversações de paz em Angola , a decorrer em Lusaca , disseram a a agência Lusa que a UNITA informou

Appendix B

About page

The analysis of the co-occurrence patterns between words allows for a better understanding of their use (and meaning) and its most straightforward applications are lexicography and linguist description in general. DeepString - Syntax Deep Explorer is a tool that allows to obtain an easy and efficient access to co-occurrence data obtained from Portuguese texts. The resulting co-occurrence statistics are represented in lex-grams, that is, a synopsis of the syntactically-based co-occurrence patterns of a word distribution within a given corpus. DeepString - Syntax Deep Explorer will allow the development of finer lexical resources and the improvement of NLP systems in general, as well as providing public access to co-occurrence information derived from parsed corpora.



DeepString - Syntax Deep Explorer builds the lex-grams from a corpus processed by STRING. Lex-grams are presented in a user-friendly way through a graphical interface. It uses several association measures to quantify several co-occurrence types, defined on the syntactic dependencies (e.g. subject, complement, modifier) between a target word lemma and its co-locates. The syntactic dependencies are extracted by XIP, the STRING's parser. Currently, and for this demo version, DeepString uses the output of 3 large-sized Portuguese journalistic text (**CETEMPublico**, **Parlamento** and **Desportivo**).

The XIP syntactic dependencies analysed in DeepString are:

- `SUBJ`: links a verb and its subject,

  O Pedro dorme. (Pedro sleeps.)

  `SUBJ(dorme, Pedro);`

- `CDIR`: links a verb and its direct complement,

  O Pedro leu o livro. (Pedro read the book.)

  `CDIR(leu, livro);`

- `CINDIR`: links the verb with a prepositional phrase,

  O Pedro deu um livro ao João. (Pedro gave a book to João.)

  `CINDIR(deu, João);`

- `MOD`: links a modifier with the modified element; it includes prepositional complements of that element,

  O Pedro leu um bom livro de poesia. (Pedro read a good poetry book.)

  `MOD(livro, bom) MOD(livro, poesia);`

- `COMPL`: links a predicate (verb, noun or adjective) to each of its essential complements,

  O livro já foi lido pelos estudantes. (The book has already been read by the students.)

  `COMPL(lido, estudantes);`

- `QUANTD`: links a nominal head and a quantifier,

  O Pedro comprou um pacote de açúcar e cinco maçãs. (Peter bought a pack of sugar and five apples.)

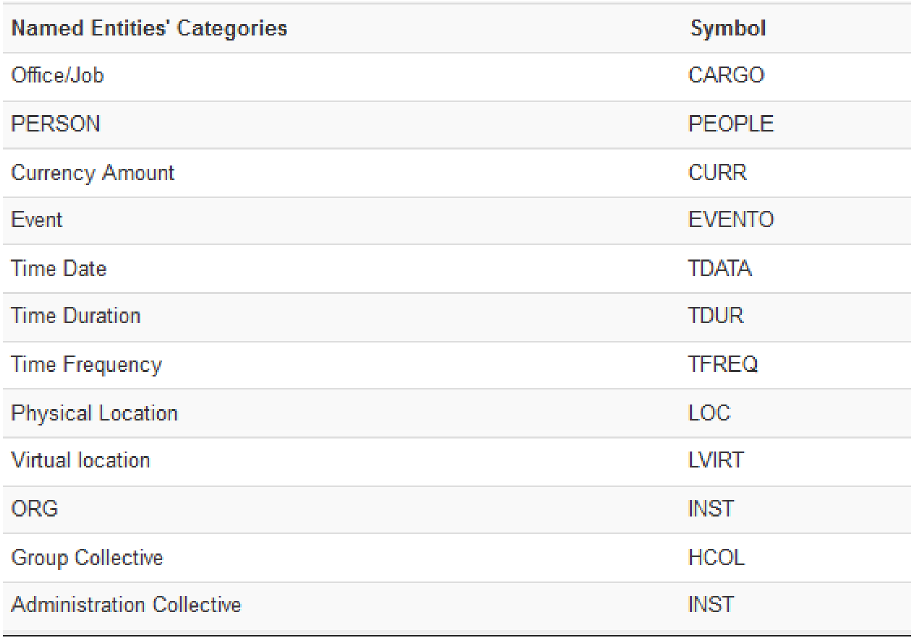
  `QUANTD(pacote,um) QUANTD(açúcar,pacote) QUANTD(maçãs,cinco);`

- `CLASSD`: links a nominal head and a nominal classifier.

  Essa espécie de peixe alimenta-se deste tipo de plâncton. (That species of fish feeds on that type of plankton.)

  `CLASSD(peixe,espécie) CLASSD(plâncton,tipo).`

Named Entities are captured by the XIP parser with a dependency, which delimits and assigns them to one of several general categories. The categories used in this system are presented in the following table. DeepString - Syntax Deep Explorer collapses the instances of all the entities within the same category for the statistic calculation of lex-grams.



Named Entities' Categories	Symbol
Office/Job	CARGO
PERSON	PEOPLE
Currency Amount	CURR
Event	EVENTO
Time Date	TDATA
Time Duration	TDUR
Time Frequency	TFREQ
Physical Location	LOC
Virtual location	LVIRT
ORG	INST
Group Collective	HCOL
Administration Collective	INST

The web application provides users with an interface that allows them to exploit these co-occurrence patterns in the context of selected corpora.

The corpora available at the moment are:

- **CETEMPúblico**: A corpus composed by texts from the Portuguese daily national newspaper Público, from 1991 to 1998. This corpus is publicly available and it is distributed by Linguateca. It contains 175,350,145 words;

- **Parlamento**: A corpus composed by minutes from sessions of the Portuguese Parliament, from 1976 to 2018. This corpus contains 123,633,859 words;

- **Desportivo**: A corpus composed by texts from the Portuguese sports newspaper named O Jogo from 1999 to 2005 and from another Portuguese sports newspaper named A Bola from 2000 to 2006. This corpus contains 100,161,374 words.

DeepString - Syntax Deep Explorer (v. 1.0) was initially developed as part of a dissertation in Information Systems and Computer Engineering by José Correia (2015), and the current version (2.0) was subsequently developed and very much improved and by João Pedro Trindade (2020), who added the comparison features, updated the CETEMPúblico corpus, and built anew the Parlamento and Desportivo corpora, adding them to the system. To quote, use the references below:

José Correia. Deep Syntax Explorer. (MSc thesis) Mestrado em Engenharia Informática e de Computadores, Instituto Superior Técnico, Universidade de Lisboa (2015).

José Correia, Jorge Baptista and Nuno Mamede. Syntax Deep Explorer. Branco, A. et al. (eds). Computational Processing of the Portuguese Language. 12th International Conference PROPOR'2016, Tomar (Portugal), July 13-15, 2016. Proceedings. Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence. Berlin: Springer. [to appear]

The databases of cooccurrence statistics for these corpora can be provided upon request. Contact us for further details.

Also, if you need to parse and use cooccurrence statistics of another corpus of your choosing, please contact us. Last update: April 27, 2020

Appendix C

Evaluation sentences

This Appendix contains the sentences used in the user test described in Section 4.1. This test measured the impact of the highlighting functionality on the user experience. In this test, users would have to find the target word and the co-occurrent word in example sentences taken from the Explorer's interface.

Since the word highlighting algorithm has a varying performance depending on the POS of the target words, the words selected for this test also include balanced examples from the four different POS.

In this test, a group of 10 users were recruited and each person was asked to point both the target and the co-occurrent word in 48 sentences divided into 2 sets of sentences, named Group A and Group B. In Group A, no element of the sentences was highlighted, and in Group B all the sentences had some elements highlighted, following the word highlighting algorithm. Some sentences in Group B contained incorrectly highlighted words as well as missed, non-highlighted words, to better simulate the functioning of the highlighting algorithm.

In the sentences below, the correctly highlighted words appear in green and the incorrectly highlighted words appear in red, in the test prompted to the user all of the colored words appeared in bold instead.

C.1 Short sentences

C.1.1 Noun

- Por seu turno , António Alves Guerreiro , de 54 anos , diz que o RMG " é só para tapar os olhos a as pessoas " .

- A baixa irá dever se principalmente a o arrefecimento de a actividade imobiliária , cujo crescimento exponencial em o ano de 1993 se deveu em grande parte a capitais estrangeiros .

- Disse Francisco Vital , insatisfeito , mas já a frio , que a partida tivera duas partes distintas .

C.1.2 Verb

- António Ramalho , director de o Bingo , desmente as acusações e salienta que os postos de trabalho nunca estiveram em causa .

- " Lavagem a o cérebro " , pressão de o advogado em a altura que lhe deu conta de a inevitável cadeira eléctrica na falta de uma confissão , alega o presumível homicida de Luther King .

- Antes de discursar , Xavier ainda viu o seu colega Paulo Portas pedir a suspensão imediata de os trabalhos , para o PP decidir o que fazer em o momento de a votação.

C.1.3 Adjective

- O Mundial 98 será o maior de sempre .

- Hoje é ele quem toma conta de os irmãos mais novos .

- Se o acordo for cumprido , tal com ele foi escrito , a consequência vai ser mais recessão , maior aperto .

C.1.4 Adverb

- Mas nem só de reconciliações se falará em Nova Iorque .

- Ora , os ricos não estão para isto .

- Depois de terem terminado a fase regular só com uma derrota em os sete jogos realizados , os maiatos não deverão ir muito mais longe .

C.2 Long sentences

C.2.1 Noun

- Em os EUA , o limiar de a pobreza está fixado em 13.914 dólares / ano (cerca de 1700 contos) para uma família de quatro pessoas e em os 6932 dólares / ano (cerca de 860 mil escudos) por pessoa e não considera o património pessoal nem os subsídios de carácter social .

- O processo de adesão de os Camarões , membro a partir de 1 de Novembro , durou seis anos e não quebrou esta regra , na medida em que parte significativa de o território de aquele país esteve sob administração britânica , sendo o inglês um de as suas línguas veiculares .

- A mulher de o desaparecido , ao contrário de o que fez a família de Armando Estudante , não comunicou o caso a a Judiciária , pelo que esta corporação só veio a tomar conhecimento de o caso através de comunicação de a GNR .

C.2.2 Verb

- Por outro lado , a metodologia de análise adoptada , arrisca se a transmitir a os contribuintes a perigosa mensagem : um novo Governo PSD irá limitar se a gerir o " status quo " porque não tem mais margem de manobra (ou pior do que isso porque se dá por satisfeito com a sua obra em áreas tão chave como a Saúde , Educação ou a Segurança Social) e , como admite a a partida , que também não é possível " cortar " em nada e o défice terá mesmo de se reduzir para três por cento de o PIB já a partir de 1997 , qualquer eventual aumento de os gastos , ou em o limite a sua simples manutenção , terá de esta vez (e ao contrário de o que se passou em os últimos dez anos) de se traduzir , em menor ou maior grau , em um maior esforço de os contribuintes (o que sendo de boa política , não é popular) .

- A votação representa também uma vitória significativa para os que se propõem efectuar este tipo de investigação , uma vez que conseguiram convencer muitos senadores antiaborto que defender a utilização de restos fetais não é a mesma coisa que defender a prática que lhes dá origem .

- Mas como fazê lo , se o ministério mantém mal-intencionado secretismo sobre os dados que vai acumulando em cada campanha , mesmo dando de barato que em muitos casos se trate de pura incompetência de a administração pública ?

C.2.3 Adjective

- A formação espanhola de o Chapela venceu ontem o Torneio Internacional Feira de S. Mateus , que decorreu em Viseu , ao derrotar em a final o Valladolid , por 28-27 (13-13 a o intervalo) , em a quinta e última jornada de a prova , em a qual participaram ainda FC Porto , Benfica , Sporting e Madeira SAD .

- O centro de produção pretende estabelecer uma simbiose entre os ex-alunos de a escola , principalmente aqueles que já adquiriram experiência profissional , e os novos estudantes de cinema e video , aumentando , para ambos , as hipóteses de trabalho e a execução de projectos , em os quais a rentabilização económica não é o principal objectivo - - explicou Paulo Santos , director de produção .

- Depois de isso , a Kaos tem prevista a estreia de o LL Project , de o DJ Luís Leite , de um novo projecto de o Porto designado Algo Rítmico , um novo maxi de os Ozone , e álbuns de estes dois últimos projectos .

C.2.4 Adverb

- Os centros de emprego deverão passar a ser recentrados em três coisas em que , actualmente , estão a fazer muito pouco : tratar os desempregados como pessoas , fornecendo lhes alternativas concretas , desde formação adaptada até propostas de reinserção , o que implica que se defina com cada pessoa o seu projecto profissional ; permitir a o centro de emprego ter um papel mais activo junto de as empresas , pondo a a sua disposição de instrumentos de política de emprego que muitas vezes não são conhecidos porque , justamente , os centros de emprego não o fazem ; e tornar os centros em pólos de animação de o desenvolvimento local , capazes de dar vida a uma rede local de educação e formação .

- Assim , em o seu primeiro Porto-Sporting como adjunto , com a rivalidade Pinto de a Costa-João Rocha estava em o auge e com Artur Jorge a não conseguir ganhar em as Antas , Octávio seria acusado de agredir à cabeçada e a murro alguns responsáveis sportinguistas , indo assim mais além do que o próprio treinador principal que , impávido e sereno , comandou a sua equipa sem se envolver em as " guerras " marginais em que Pinto de a Costa e Octávio mostravam estar como " peixes em a água " .

- Depois de a reunião de a Comissão de Acompanhamento de o PDM , que se realiza em a próxima semana , embora ainda sem dia marcado , o instrumento que regulará o ordenamento do território concelhio em os próximos tempos será analisado e votado em Assembleia Municipal , cuja data também ainda não está definida.