ET: A Workstation for Querying, Editing and Evaluating Annotated Corpora

Elvis de Souza
Department of Letters
PUC-Rio, Brazil
elvis.desouza99@gmail.com

Cláudia Freitas
Department of Letters
PUC-Rio, Brazil
claudiafreitas@puc-rio.br

Abstract

In this paper we explore the functionalities of ET, a suite designed to support linguistic research and natural language processing tasks using corpora annotated in the CoNLL-U format. These goals are achieved by two integrated environments – Interrogatório, an environment for querying and editing annotated corpora, and Julgamento, an environment for assessing their quality. ET is open-source, built on different Python Web technologies and has Web demonstrations available on-line. ET has been intensively used in our research group for over two years, being the chosen framework for several linguistic and NLP-related studies conducted by its researchers.

1 Introduction

Annotated corpora are the basis for several natural language processing (NLP) tasks, serving as material from which machine learning systems learn how to perform linguistic analysis and as evaluating material against which system analyses can be confronted. However, manipulating annotated corpora is a costly activity for humans alone.

In this context, we present ET: a workstation for querying, editing and evaluating annotated corpora[1][2]. The underlying idea is to facilitate linguistic research using annotated corpora in the CoNLL-U format[3]. ET is composed of two integrated web-browser-based interfaces that are easily manipulated by non-developers at the same time that it provides tools to explore annotated texts driven by simple-to-build yet linguistically complex queries. The system main language is Portuguese but comes with English translation for the majority of its modules, it was built using different Python Web technologies and has Web demonstrations available online. One can use the available demonstration pages for working with small corpora or install a local copy[4].

ET has been intensively used in our research group for over two years, being the chosen framework for several linguistic and NLP-related studies conducted by its researchers. In section 2, we discuss other tools that inspired ET and how different it is from them. In section 3, we explore the interface for querying and editing annotated corpora, and in section 4 we demonstrate how to assess their quality using the workstation. Finally, in section 5, we outline some future perspectives for the tool.

2 Related tools

ET is a workstation that focuses on building quality corpora for Natural Language Processing, but not from scratch. Thus, it should not be confused with tools aimed at corpus annotation from raw pieces of text, such as Arborator (Gerdes, 2013), ConlluEditor (Heinecke, 2019) and UD Annotatrix (Tyers et al., 2017), nor with tools aimed at Corpus Linguistics studies, such as AntConc (Anthony, 2005) and CQPweb (Hardie, 2012).

From the point of view of text analysis tools, although ET query environment was largely inspired by AC/DC (Santos and Bick, 2000), from Linguateca (Santos, 2011) – one of the most important repositories for NLP in Portuguese –, AC/DC is based on an early version of CWB (Evert and Hardie, 2011), a robust and widely used processor capable of quickly and reliably processing corpora of millions of tokens, a task which ET does not propose to do with such high quality. The query-

[1]The workstation GitHub page and its live demonstration are available at http://comcorhd.letras.puc-rio.br/ET.

[2]The Portuguese term for workstation is "Estação de Trabalho" (lit. workstation), the reason why the suite was named "ET".

[3]The CoNLL-U format is used by the Universal Dependencies (Nivre et al., 2016) project. The format is described at: https://universaldependencies.org/format.html. Accessed on 5 jan. 2021.

[4]Only tested on Ubuntu distributions.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 35–41
November 7–11, 2021.

ing module from ET allows for querying syntactic dependencies in the Universal Dependencies format, sorting the results based on their annotation distribution, such as part-of-speech, morphological features and dependency labels distribution, and affords the addition of filters in order to further specify the query. In addition to the search tools, ET comprises both a manual and a rule-based treebank editing system that was inspired by AC/DC's *Corte e costura* (Mota and Santos, 2009), where the user codes linguistic rules that will search and correct annotation mistakes.

From the point of view of corpora annotation, tools like Arborator will suit better as they allow the editing of trees with features such as graphical editing and user management for collaborative annotation, which facilitate the process for annotators and project coordinators. What is different about ET, in turn, is the integration of the querying environment with evaluation methods to assess corpora that were previously annotated by humans or NLP systems. With the application of linguistic rules and the verification of inconsistent patterns, the tool makes it possible to linguistically guide the work of reviewing annotated corpora for NLP.

3 Querying and editing annotated corpora

Interrogatório (Portuguese word for "Interrogatory") is the name of the first of the two environments that compose ET. Its purpose is to make it easy for anyone to query and revise annotated corpora.

The system was built using Python CGI technology in the back-end and JQuery in the front-end. A server and a client machine are needed, but the same machine can work as both server and client, although installing and running a server-side will require intermediate technology skills. Installation steps and requirements are available in the workstation GitHub page.

Managing corpora The "Manage corpora" hub can be accessed from Interrogatório main menu on the top. This is where a new file is uploaded to the workstation. This file can be either a *.conllu* file, which already carries annotation, or a raw text in a *.txt* file. In the latter case, the text will be tokenized, tagged and parsed by UDPipe (Straka et al., 2016) using the models for Portuguese or English[5]. Once there are files in the repository, this page will present the number of sentences and tokens in each corpus.

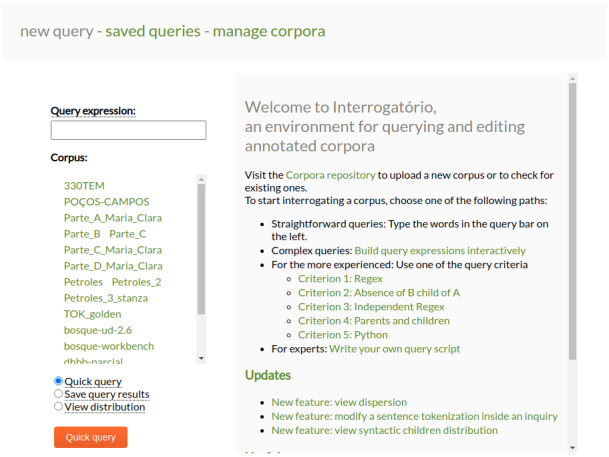

Figure 1: Interrogatório homepage. The main menu is on the top, a list of corpora is on the left, and the user guide is on the right.

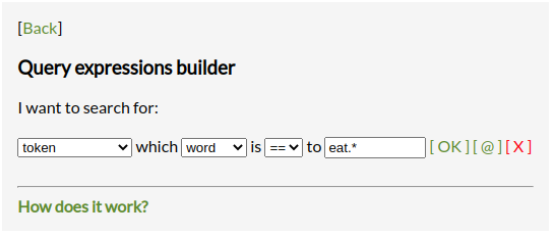

Figure 2: Query expression builder available on the Interrogatório homepage

Interrogating a corpus There are multiple ways one can interrogate a corpus and find sentences with specific annotation. The search criteria have been developed attending the needs of researchers – linguists – who used the workstation. Whenever a new search must be done and can not be easily achieved, a new query system can be built and documented by code. The current five search criteria are explained and exemplified in the homepage user guide (on the right, in Figure 1).

Since querying treebanks can be a complex task for beginners, Interrogatório comes with a "Query expressions builder", a GUI that helps the user when building query expressions by showing them the tags and relations between them in natural language (Figure 2). The intended search is then translated into the query syntax.

As a last resort, when a query is too complex and

[5]Such models were trained on Rademaker et al. (2017) and Silveira et al. (2014) for Portuguese and English, respectively

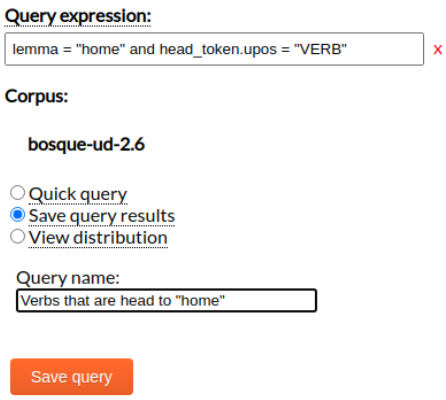

Figure 3: Users can save results for later analyses

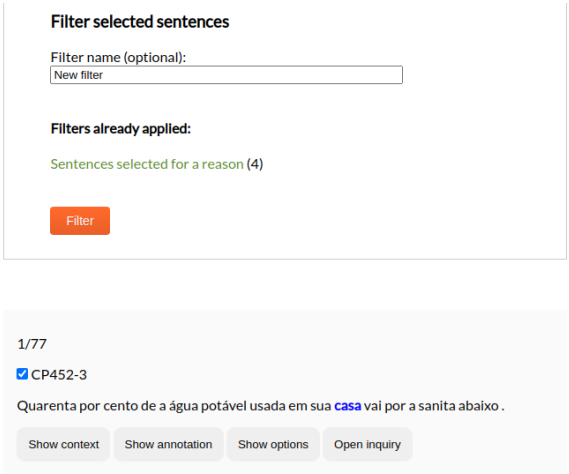

Figure 4: Filtering selected sentences

cannot be done by any of the implemented search criteria, Interrogatório will allow the user to write their own Python code to look for sentences following a given model by opening the menu "Write your own query script".

Saving a query Interrogatório will not make any previous indexation on corpora uploaded to the workstation, which is an architecture decision that makes queries slower – in comparison to systems that perform indexation – but that is what allows the dynamism needed for changing the corpus while being able to make queries in it real-time. The time it takes to complete a query will depend on the search criterion and the amount of results that the query returns. Thus, if the user intends to execute a query that returns too many results, it might be more efficient to save these results for later. They will be found on the page "saved queries".

Finding sentence information Returned sentences within a query are presented on the screen along with some visualization options. By clicking on the "Show context" button one can see the sentences before and after the sentence in focus. Besides, through the "Show annotation" button the user can see the annotation of the sentence, allowing them to understand how it is currently being analyzed and judge its quality.

Editing sentences Whenever a user finds a mistake in the annotation of a sentence, be it sporadic or in cases in which the query expression was meant to lead to errors that need correction, one can click "Open inquiry". An "inquiry" is a new tab with the annotated sentence displayed in a table, one token per line, ready to be edited. Options such as modifying the sentence segmentation are also available, making it possible to add or remove a token, split the sentence or join two different sentences using the GUI, leaving the difficult job of ensuring that the output format is correct to the machine.

Another possibility for changing sentence annotation automatically is by running a correction script. Once in the query results page one can navigate to the "Options" menu and select "Batch correction", where they will be able to download a model script for automatic correction. The script will then be edited by the user maintaining a Python syntax to both find the tokens that need to be changed and to assign them the new annotation. Once the script is uploaded back to Interrogatório, a page will simulate the changes so the user can decide whether the changes are as intended or not before applying them.

Filtering query results Once inside a saved query results page, the user is able to filter the results, separating sentences in different slots. The reasons for applying this are various:

- The same query can lead to different but related linguistic phenomena. Disentangling linguistic phenomena is a common task in research, and it is something that at times can only be done by reading each sentence inside a query results page. Keeping track of those sentences in different slots may facilitate linguists' work.
- The main query expression might not be fine-grained enough to find the phenomenon being studied, in which case there is the need of refining a query with other queries inside it.

Query: 5 "president."
Corpus: bosque-ud-2.6.conllu

Number of occurences: 162
Number of **deprel** diferentes: 11

deprel	frequency	in files
nsubj	56	52
appos	29	27
nmod	29	28
obj	13	13
obl	13	13
obl:agent	7	7
conj	6	6
acl:relcl	3	3
root	3	3
xcomp	2	1
iobj	1	1

Figure 5: Distribution of dependency relations for the query expression "president." in corpus Bosque-UD v2.6

- A user may want to apply a correction script to only a subset of sentences that are the results of a query, so filtering the sentences that should be automatically corrected and separating them from the main query is needed.

This feature was developed while looking for omitted subjects in different Portuguese corpora, a research that required complex queries that involved the absence of a tag in the sentence (the "subject" relation) and five further query specifications (filters) to remove sentences such as those without verbs and those whose main verbs express meteorological condition such as "to rain", "to snow" etc. (Freitas and de Souza, 2021)

Distribution of linguistic phenomena When a user executes a query they will see the sentences returned with the tokens being looked for in the query expression in bold. However, if one is not willing to read sentence by sentence but, instead, wants to see the distribution of any annotation for the words in bold, it is possible to view the distribution of their part-of-speech, dependency relation, features, lemma etc., as shown in Figure 5. This page can be accessed by clicking on the "Options" menu on the top of the screen and selecting "View distribution" or before executing a query selecting the same option in the homepage.

4 Evaluating annotated corpora

Julgamento (Portuguese word for "Judgment") is the name of the second of two environments that compose ET. Its purpose is to evaluate the quality of annotated corpora by different methods that search for inconsistencies in the annotation. Currently, Julgamento provides three methods that help to search for annotation inconsistencies – n-grams, linguistic rules and contrastive analysis – presented below.

The system was built using Python Flask framework technology in the back-end and JQuery in the front-end. Its installation process and architecture are the same as Interrogatório: a server and a client machine are needed, one for setting the system up and the other for the user to browse through the interfaces. Installation steps and requirements are available in the workstation GitHub page.

Managing corpora From the top menu in any screen one can have access to the "manage corpora" hub in Julgamento. Interrogatório and Julgamento are integrated[6], which means that whatever corpora are uploaded to one environment will also be accessible through the other, since the corpus file is the same.

Some of Julgamento's evaluation methods are based on the idea of contrastive analysis, which will require the user to upload a corpus with two different annotations (e.g. annotations provided by two different systems, two different human annotators or a gold-standard and a system counterpart), basing the evaluation on the confrontation of both. It is in this page that the user is able to upload the main corpus and its alternative annotation, that is, two different CoNLL-U files with the same sentences but different annotation.

Finding inconsistent n-grams This is a method for detecting inconsistencies in the annotation of a corpus uploaded to Julgamento. It is largely based on de Marneffe et al. (2017) method, although some important changes were applied and are still under test. The general idea is that hardly two dependency pairs with the same context and same lemmas will have different dependency relations, as in Figure 6, in which "António" and "Oliveira" are a dependency pair (*António* being the head) but have a different dependency relation in each sentence (*nmod* and *flat:name*), indicating an inconsistency that needs fixing. The method will display all the n-grams in the corpus that, although similar, have different relations, leaving it to the user

[6]Interrogatório must be installed in the same folder as Julgamento to make it possible to integrate them. To ensure that the integration is working, the "manage corpora" page on Interrogatório should present a large orange strip warning that "Interrogatório is integrated to Julgamento".

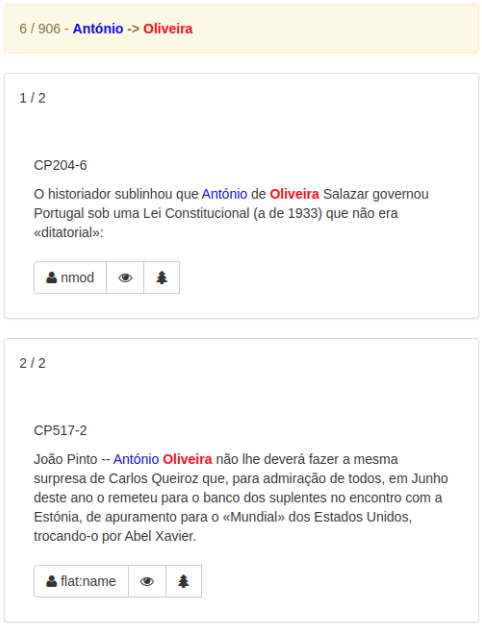

Figure 6: Inconsistent n-gram between two sentences (*António* and *Oliveira* are related by different tags in each sentence)

to judge whether they are wrongly annotated and giving them the ability to correct their annotation.

Checking for validation errors Two methods for checking validation errors in a corpus are available in Julgamento. One is the official Universal Dependencies project script for validating a new corpus[7], in which several rules will be applied to a CoNLL-U file to ensure that the format is correctly encoded and that basic points from UD annotation guidelines have not been skipped while annotating that corpus.

Another script was built by our team and focuses on Portuguese grammar rules that, when skipped, provide evidence of incorrect annotation or inconsistency. The rules were built using the same syntax from interrogating a corpus in Interrogatório (as discussed earlier) and can be edited[8] to conform to any project annotation guidelines.

Comparing corpora Other way of judging a corpus quality is by comparing two different annotations of the same sentences. These two annotations can be provided by two different systems, by two different human annotators or even by a gold-standard and a system counterpart, in which case the comparison will provide an evaluation of the system output. Once two annotations are uploaded to Julgamento, new methods will unlock:

- **Metrics from conll18_ud_eval.py**: This feature applies the evaluation metrics from the CoNLL 2018 Shared Task (Zeman et al., 2018) on the corpus to compare the second annotation to the first. The metrics encompass precision, recall and F1 of attributes such as tokenization, sentence segmentation, lemmatization, POS-tagging, the attachment of dependency relations etc.[9]

- **Sentences accuracy**: This feature presents how many sentences received exactly the same annotation in both versions of the corpus. Its relevance is based on a point that Manning (2011) makes – we usually assess quality by number of correct tokens, but a perhaps more difficult yet realistic way of assessing quality is by the number of totally correct sentences.

- **Accuracy per morphosyntactic category**: The ”accuracy” for each part-of-speech tag and dependency relation are described through tables, as in Figure 7. Clicking on any dependency head attachment percentage will lead to a page where the user will find sentences in which a token is attached to different heads when comparing both versions of the corpus.

- **Confusion matrix**: CMs facilitate visualizing what are the most usual divergences in POS tags and dependency relations. In Figure 8, the diagonal line shows the number of tokens in which annotation both versions converged, whereas numbers out of this diagonal will show divergent analyses that could suggest inconsistencies in the training data when the alternative annotation was provided by a model trained on the corpus. Clicking any number will open a page listing the sentences with the focused token in bold where the user can judge which is the correct annotation and edit the sentence when needed.

[7]The script is named "validate.py", which can be called from the workstation interface. The returned sentences can be edited from inside it, as well. The source code is available at: https://github.com/UniversalDependencies/tools. Accessed on 5 jan. 2021.

[8]Rules can be edited from the file "validar_UD.txt".

[9]More information on the original evaluation methods can be found at: https://universaldependencies.org/conll18/evaluation.html. Accessed on 5 jan. 2021.

DEPREL	Total	Hits	LAS	DEPHEAD errors
_	744	100.0%	100.0%	0.0%
acl	117	82.05128205128204%	62.39316239316239%	19.65811965811966%
acl:relcl	118	85.593220338983 06%	49.1525423728813 56%	36.440677966101696%
advcl	109	79.81651376146789%	55.04587155963303%	24.770642201834864%
advmod	327	93.88379204892966%	70.9480122324159%	22.935779816513 76%
amod	369	89.159891598915 99%	85.907859078590 79%	3.2520325203252036%
appos	161	73.913043478260 86%	54.0372670807453 45%	19.875776397515526%
aux	49	89.795918367346 94%	89.795918367346 94%	0.0%
aux:pass	60	91.666666666666 66%	91.666666666666 66%	0.0%
case	1476	98.64498644986 45%	97.222222222222 1%	1.4227642276422763%

Figure 7: Accuracy for some of the dependency relations when comparing two annotations for the same corpus

corpus_2	ADJ	ADP	ADV	AUX	CCONJ	DET	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X	_	All
corpus_1																	
ADJ	385	1	1	0	0	0	27	1	0	2	0	0	0	23	0	0	440
ADP	0	1513	4	0	0	5	0	0	0	1	0	6	0	0	0	0	1529
ADV	1	5	340	0	0	0	5	0	0	0	0	3	0	2	0	0	356
AUX	0	0	0	225	0	0	0	0	0	1	0	0	0	7	0	0	233
CCONJ	0	2	1	0	203	0	0	0	0	1	0	0	0	0	0	0	207
DET	1	1	3	0	0	1544	0	2	5	3	0	1	0	0	0	0	1560
NOUN	35	2	3	0	0	1	1982	1	0	25	0	0	0	12	0	0	2061
NUM	2	2	0	0	0	0	2	237	0	5	0	0	0	0	0	0	248
PRON	0	0	2	0	0	5	2	2	332	0	0	6	0	0	0	0	349
PROPN	5	0	2	0	0	0	38	1	0	822	0	0	0	3	0	0	871
PUNCT	0	0	0	0	0	0	0	0	0	0	1343	0	0	0	0	0	1343
SCONJ	0	10	9	0	0	1	0	0	8	0	0	190	0	0	0	0	218
SYM	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	30
VERB	10	0	1	6	0	0	8	0	0	3	0	0	0	883	0	0	911
X	1	1	0	0	0	0	8	1	0	17	0	0	0	0	2	0	30
_	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	744	744
All	440	1537	366	231	203	1556	2072	245	345	880	1343	206	30	930	2	744	11130

Figure 8: Confusion matrix portraying divergences in POS annotation

5 Concluding remarks

In this paper we presented ET, a Workstation for Querying, Editing and Evaluating Annotated Corpora. Its aims are to facilitate linguistic research and evaluate annotated corpora in the CoNLL-U format, reuniting functionalities that are not new ideas along with innovative ways of characterizing and judging annotated corpora. Both Interrogatório and Julgamento, integrated parts of the workstation, are available on-line in the project GitHub page for download and usage, as well as a live demonstration which does not need previous installation.

Although ET is not to be confused with corpus analysis tools and corpora annotation tools alone, the workstation can benefit from features of both kinds of tools. In the future, thus, it is possible to expand its functionalities so it will work as both kinds of tools as well, increasing the range of tools available to the user without leaving the workstation.

Acknowledgments

We would like to thank Luísa Rocha, who assisted in the conception of the system in its very initial steps, and Aline Silveira, Tatiana Cavalcanti, Maria Clara Castro and Wograine Evelyn, who have chosen the workstation as main framework for their linguistic research and daily work in various projects, challenging the tool to fit their (computational) linguistic needs. Elvis de Souza thanks the National Council for Scientific and Technological Development (CNPq) for the Masters scholarship grant number 130495/2021-2.

References

Laurence Anthony. 2005. Antconc: design and development of a freeware corpus analysis toolkit for the technical writing classroom. In *IPCC 2005. Proceedings. International Professional Communication Conference, 2005.*, pages 729–737. IEEE.

Marie-Catherine de Marneffe, Matias Grioni, Jenna Kanerva, and Filip Ginter. 2017. Assessing the annotation consistency of the universal dependencies corpora. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 108–115.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century corpus workbench: Updating a query architecture for the new millennium.

Cláudia Freitas and Elvis de Souza. 2021. Sujeito oculto às claras: uma abordagem descritivo-computacional/omitted subjects revealed: a quantitative-descriptive approach. *REVISTA DE ESTUDOS DA LINGUAGEM*.

Kim Gerdes. 2013. Collaborative dependency annotation. In *Proceedings of the second international conference on dependency linguistics (DepLing 2013)*, pages 88–97.

Andrew Hardie. 2012. Cqpweb—combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics*, 17(3):380–409.

Johannes Heinecke. 2019. Conllueditor: a fully graphical editor for universal dependencies treebank files. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 87–93.

Christopher D Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer.

Cristina Mota and Diana Santos. 2009. Corte e costura no ac/dc: auxiliando a melhoria da anotação nos corpos.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria de Paiva. 2017. Universal dependencies for portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 197–206.

Diana Santos. 2011. Linguateca's infrastructure for portuguese and how it allows the detailed study of language varieties. *OSLa: Oslo Studies in Language 32 (2011) ISSN: 18909639 Volume edited by J.B.Johannessen*.

Diana Santos and Eckhard Bick. 2000. Providing internet access to portuguese corpora: the ac/dc project. In *Maria Gavrilidou; George Carayannis; Stella Markantonatou; Stelios Piperidis; Gregory Stainhauer (ed) Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)(Athens 31 May-2 June 2000)*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Francis Tyers, Mariya Sheyanova, and Jonathan Washington. 2017. Ud annotatrix: An annotation tool for universal dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 10–17.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.