Revista de Lenguas para Fines Específicos 23.2 (2017), pp. 111-132
http://dx.doi.org/10.20420/rlfe.2017.175
ISSN: 2340-8561

Servicio de Publicaciones y Difusión Científica
Universidad de Las Palmas de Gran Canaria

Bridging the gap within text-data analytics: a computer environment for data analysis in linguistic research

Carlos Periñán-Pascual[1]
Universitat Politècnica de València

ABSTRACT

Since computer technology became widespread available at universities during the last quarter of the twentieth century, language researchers have been successfully employing software to analyse usage patterns in corpora. However, although there has been a proliferation of software for different disciplines within text-data analytics, e.g. corpus linguistics, statistics, natural language processing and text mining, this article demonstrates that any computer environment intended to support advanced linguistic research more effectively should be grounded on a user-centred approach to holistically integrate cross-disciplinary methods and techniques in a linguist-friendly manner. To this end, I examine not only the tasks that are derived from linguists' needs and goals but also the technologies that appropriately deal with the properties of linguistic data. This research results in the implementation of DAMIEN, an online workbench designed to conduct linguistic experiments on corpora.

Keywords: text-data analytics, corpus linguistics, text mining, software, DAMIEN

1. Introduction

Today most linguists take an empirical approach to research by collecting, analyzing, evaluating and interpreting corpus data. Linguists run experiments to test the validity of linguistic claims on the basis of corpus-based evidence rather than introspective judgments. Corpora must be machine-readable because of their size,

[1] **Corresponding author** – Applied Linguistics Departament – Escuela Politécnica Superior de Gandía, Universitat Politècnica de València, Calle Paranimf, 1, 46730 Gandía, Valencia (Spain).

Email: jopepas3@upv.es

so that computer software can rapidly search for relevant evidence. Therefore, empirical linguists usually employ software that makes use of statistics to analyze large collections of text data. Both corpus linguistics and statistics have long contributed to the development of text-data analytics, but we should keep in mind that methods and techniques from other disciplines such as natural language processing (NLP) and text mining also play a significant role in this field of research. Whereas NLP enables researchers to exploit techniques from computational linguistics —e.g. part-of-speech (POS) tagging, stemming, morphological analysis or syntactic parsing— and from information extraction —e.g. entity or relation extraction, text mining is aimed at extracting "useful information from data sources through the identification and exploration of interesting patterns" (Feldman & Sanger, 2007, p. 1).

In the last few decades, there has been a proliferation of software for practical text-data analytics, whose design has been centred on the most representative tasks of a given discipline, e.g. TextSTAT and AntConc in corpus linguistics, R in statistics, GATE in NLP or WEKA in text mining. The functionalities integrated into this type of software are usually modelled by the discipline itself (i.e. task-oriented design), this being the reason why we talk about software for corpus linguistics, statistics, and so on. Alternatively, software designers can get a deep understanding of users' needs and goals (i.e. user-centred design); thus, only after these needs and goals have been analysed, tasks to meet them can be determined, where a task is regarded as an intermediate step based on state-of-the-art technology:

> Design based solely on understanding activities or tasks runs the risk of trapping the design in a model imposed by an outmoded technology [...]. Looking through the lens of goals allows you to leverage available technology to eliminate irrelevant tasks and to dramatically streamline activities. Understanding users' goals can help designers eliminate the tasks and activities that better technology renders unnecessary for humans to perform. (Cooper, Reimann & Cronin, 2007, p. 16)

Therefore, the utility of a computer application for empirical linguists should be determined by understanding their expectations when conducting corpus-based research.[2] In particular, linguists' needs and goals are mainly aimed at testing hypotheses about the nature of language by using a corpus as a source of objective

[2] Anthony (2013, p. 142) distinguished between "corpus-driven" approach, i.e. where "direct observations of the corpus should be the starting point of analyses", and "corpus-based approach", i.e. where "all corpus analyses are essentially testing pre-existing linguistic theories (a model) against a representative sample of real language (the corpus data)". In this article, both terms are used interchangeably to refer to any empirical approach in which patterns of language use are observed in a collection of real-language texts.

evidence. Consequently, a computer workbench that recognizes usage patterns in corpora is likely to provide linguists with information that can be used to reject or accept their initial hypotheses. In this regard, this article demonstrates that linguists can reach their research goals more effectively by integrating methods and techniques from various fields within text-data analytics. To this end, DAMIEN (DAta MIning ENcountered) was implemented, because today's linguist-friendly software does not have sufficient capability to do so. The remainder of this article is organized as follows: Section 2 briefly describes how software oriented to corpus linguistics and statistics help to develop linguistic research; Sections 3 and 4 explore how NLP and text mining can contribute to support data analysis in corpus linguistics; Section 5 describes the most relevant technologies involved in corpus-based research; Section 6 examines the state-of-the-art software for text-data analytics; Section 7 provides a detailed description of DAMIEN; and finally, Section 8 highlights the main conclusions.

2. Corpora for linguistic research

A critical component of empirical research in theoretical and applied linguistics is the corpus, which is regarded as "a sizeable sample of real-life usage in English or another language under study, compiled and used as a source of evidence for generating or testing hypotheses about the nature of the language" (Sampson, 2001, p. 6). In this context, researchers inevitably become corpus linguists, since corpus linguistics should not be thought of as a branch of linguistics but as "the route into linguistics" (Ibid.). Linguists can explore the corpus from two different but complementary approaches: (i) from a qualitative approach, corpora are used as a test bed or example bank, and (ii) from a quantitative approach, corpora provide statistical information about words and phrases. Regardless of the approach, three major tasks enable corpus linguists, and therefore empirical linguists, to meet their goals (Antworth & Valentine, 1998):

a) Data collection and management: linguists need a database management system, which can provide them with facilities for entering, editing, sorting, searching and retrieving data.

b) Data analysis: linguists need to test their analyses, so they typically rely on tasks such as sorting data according to a criterion, searching data for specific lexical items, presenting concordances in KWIC lists, and producing statistical analysis of the data. With respect to the latter, where corpus linguistics converges with statistics, the focus does not only lie on the frequency of word occurrences but also on:

- descriptive statistics, providing a "picture" of the data through measures of

position (e.g. mean, median and mode), dispersion (e.g. variance, standard deviation and interquartile range) and shape (e.g. skewness and kurtosis), and

- inferential statistics, providing the outcomes of statistical tests and helping researchers decide the degree of significance and reliability of data through measures such as correlation, regression or multivariable analyses.

c) Data presentation: linguists need to transfer text and graphics resulting from data analysis to word processors, so that they can present their research to the scientific community.

The following sections describe how NLP and text mining can enhance data analysis in corpus linguistics.

3. Integrating NLP into corpus linguistics

The empirical linguist's concern with respect to corpora lies in recognizing usage patterns, for which it is necessary to build a representational model of the corpus in the form of a dataset. In text-data analytics, the dataset is viewed as a data matrix, i.e. a collection of data that can be arranged in columns (i.e. attributes) and rows (i.e. tuples). Particularly, in corpus linguistics, a corpus dataset typically takes the form of a set of tuples where at least one of the elements in each tuple corresponds to a text feature, that is, an instance of the attribute that denotes the unit of analysis (e.g. word, phrase, sentence, etc.) in the corpus under study. For example, a corpus dataset can hold a set of tuples containing a word and the number of occurrences in the corpus:

(1) D ={(the, 937), (of, 396), (and, 246), …. (winding, 1)}

In this context, the main contribution of NLP to corpus-based research is found in the selection and extraction of text features, which can be simple—through tasks such as stemming or lemmatization, or complex—through tasks such as phrase chunking (shallow parsing) or sentence segmentation. As in text mining, efforts to process unrestricted text "consciously shun the deeper, cognitive, aspects of classic natural language processing in favour of shallower techniques more akin to those used in practical information retrieval" (Witten, 2005, p. 2).

It is noteworthy that the extraction of complex features is typically based on pattern recognition methods, which can be grounded in regular expressions (regexps). Indeed, regexps give researchers a powerful and flexible method for pattern-based information extraction from annotated or non-annotated corpora. For example, suppose that you have a grammatically tagged corpus, where the words have been assigned a POS label:

(2) a/DT typical/JJ single/JJ phase/NN bridge/NN rectifier/NN

In this case, POS-tagged texts can be parsed with regexps for extracting specific types of phrases. To illustrate, the following regexp can be used to recognize noun phrases that are composed of zero or one determiner (DT) followed by zero or more adjectives (JJ) plus one or more singular or plural nouns (NN(S?):

(3) \b(\w+/DT\s)?(\w+/JJ\s)*(\w+/NN\s)+

4. Integrating text mining into corpus linguistics

Text mining is loosely characterized as the process of analysing large quantities of text and detecting usage patterns to extract useful information (Sebastiani, 2002), so the goal of this research field is clearly in line with that of corpus linguistics. Text mining typically exploits machine learning techniques, where learning involves the use of algorithms for the discovery of knowledge. Therefore, text-mining techniques can be applied to the corpus dataset to compute predictions on new data. According to Witten (2005), the applications of text mining can be grouped into three main categories: (i) extracting information for human consumption (e.g. text summarization, document retrieval or information retrieval), (ii) assessing document similarity (e.g. text categorization, document clustering, language identification, ascribing authorship or identifying key-phrases), or (iii) extracting structured information (e.g. entity extraction, information extraction or learning rules from text). For example, in the digital humanities, the use of text mining in literary study is not only aimed at assisting in the stylistic analysis of texts (e.g. Luyckx, Daelemans & Vanhoutte, 2006) but also at providing scholars with new insights in their interpretation of literary works (e.g. Horton et al., 2006; Plaisant et al., 2006). It is clear that text mining goes beyond the superficial counts of character strings, i.e. the statistics gathering that occurs in corpus linguistics, to focus on the search for and discovery of new information:

> [...] an application may be described as text-data mining if and only if novel information is retrieved, information that tells us something about the world rather than simply telling us something about the textual data (Tonkin, 2016: p. 10)

In this way, text mining contributes to integrate quantitative and qualitative approaches to corpus-based analysis.

After outlining the impact of text mining on the linguist's empirical research, the relevant tasks can be determined, most prominent among which are classification and clustering. On the one hand, classification is a task that involves a supervised

learning method, which consists of two main steps. First, the computer is provided with a labelled dataset (or training dataset), where each tuple belongs to a predefined class. To illustrate, suppose the dataset P:

(4) P ={(table, contain, data, cost, call, #list), (periodic, table, show, number, electron, #list), (husband, book, table, dinner, week, #furniture), (raise, book, floor, table, energy, #furniture), (large, book, need, table, content, #list)}

where the class value is #furniture or #list, which correspond to two distinct meanings of *table*, and the other elements in each tuple are the word *table* together with some of the neighbouring lexemes extracted from a random selection of documents on the Web. Second, on the basis of the labelled dataset, the computer predicts the class value of an unlabelled tuple (or test instance), e.g. that which contains the words *table, show, cost, paperback* and *book*. In this case, for example, the multinomial Naïve Bayes classifier calculates that the probability for #furniture is -22.68645 and for #list is -21.38338, so the new tuple is eventually categorized as #list.

On the other hand, clustering is a task that involves an unsupervised learning method. Thus, clustering explores an unlabelled dataset to discover groups of similar tuples. Whereas any classification method requires a training dataset, a test instance and a class attribute, clustering only requires an untrained dataset and the number of clusters.

5. Linguistic data processing

The previous sections described the tasks related to text-data analytics that facilitate corpus-based research in the linguistic realm, which are in turn derived from the needs and goals of linguists. This section describes the most relevant technologies engaged in annotating, managing and analysing text data. First and foremost, this issue directs our attention to the properties of linguistic data, since:

> Good data management includes the use of specific software tools [...], but more importantly centres on an understanding of the nature of linguistic data and the way in which the tools we use can interact with the data. Tools will come and go, but our data must remain accessible into the future. (Thieberger & Berez, 2012, p. 91)

In this regard, Simons (1998) characterized linguistic data as multilingual, sequential, hierarchically structured, multidimensional and highly integrated. Consequently, a computer workbench for linguistic research should be able to (a) process and handle data in many languages, (b) represent the text in proper sequence, (c) build

hierarchical structures of arbitrary depth, (d) attach many kinds of analysis to a single datum, and (e) store and follow associative links between related pieces of data. These requirements will ultimately determine the appropriate types of files and technologies that are involved in the processing of language resources, such as corpora, lexica and ontologies.

One of the technologies that is hailed for its relevance in data exchange is XML, which can actually support the linguistic properties described above. In fact, the best practices for corpus annotation are commonly those which conform to the XML-based standards developed within ISO/TC 37/SC 4 (ISO, 2012a, 2012b, among others), TEI—Text Encoding Initiative (Burnard, 2014), and XCES—the XML version of the Corpus Encoding Standard (Ide, Bonhomme & Romary, 2000). Moreover, there are other widely accepted XML-based standards for lexical-data encoding, such as LMF—Lexical Markup Framework (ISO, 2008) and OLIF—Open Lexicon Interchange Format (McCormick, Lieske & Culum, 2004), and for ontology exchange, such as XOL—XML-based Ontology exchange Language (Karp, 1999) and OIL—Ontology Interchange Language (Fensel et al., 2000). It should also be noted that one of the most powerful XML technologies is XSL (eXtensible Stylesheet Language Transformations), which allows researchers to transform XML documents into other formats, that is, transduce a representation optimized for computer processing (i.e. XML) into a representation intended for human consumption (e.g. HTML or plain text). Even more importantly, XSL can be used as a query language capable of accessing linguistic data from various XML documents and integrating the information into a single dataset for further processing.

However, despite the adequacy of XML and related technologies to represent and exchange language resources, "the text-based and verbose nature of XML, and the fact that it includes metadata (element and attribute names), means that it is not a compact data format" (Meier et al., 2004, p. 1), so it should be considered using another type of data store for the management and analysis of large volumes of data. Therefore, whereas the input for corpus-based research usually takes the form of a collection of weakly structured (e.g. TXT), semi-structured (e.g. HTML) or structured (e.g. XML) documents, this input has to be converted into an analysable dataset during data processing in the search for valuable patterns of knowledge, and here is where the need for databases comes in.

The relational database model, which enables us to store data "in a number of separate data tables that are linked by means of *keys* that identify particular records" (Baker, Hardie & McEnery, 2006, p. 138), can fully support the properties of linguistic data. First, database management systems enable multilingual content; for example, Aguado de Cea, Montiel Ponsoda & Ramos Gargantilla (2007) outlined the technical implications of different metamodels to represent multilinguality in

knowledge bases. Second, sequentiality can be modelled by using fields in the records that store the position or sequence number of the word, sentence, paragraph, etc. in the corpus. Third, the storage and management of hierarchical data in a relational database can be performed by means of adjacency-list or nested-set models (cf. Celko, 2004); in this manner, relational databases can provide an efficient method to process ontological information (Martínez Cruz, Blanco & Vila, 2012). Fourth, a database record (or tuple) can be used to represent a single object of data (e.g. word, phrase, sentence, and so on), and the fields of the record can be used to represent the multiple dimensions of linguistic information. Finally, the relational database model is capable of integrating the various components of linguistic analysis by means of pointers to and from the tables where corpus, lexicon and ontology data are stored. Moreover, just as XSL can be used to find and extract elements from XML data, SQL (Structured Query Language) can be used to store, manipulate and retrieve data held in a relational database. As concluded by Pitti (2004):

> Database and markup technologies represent the predominant technologies available for textual information. [...] Though there is some overlap in functionality, the two technologies are best described as complementary rather than competitive.

6. Software for text-data analytics

Improving the software that supports empirical research in linguistics requires to deploy user-centred (or human-centred) design, which is "an approach to systems design and development that aims to make interactive systems more usable", where usability is defined as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO, 2010, pp. 2-3). This section examines free-access GUI software for text-data analytics (i.e. AntConc, GATE Developer, R Commander, TextSTAT, and WEKA Explorer) in terms of the core aspects of usability, namely utility and learnability.

On the one hand, utility refers to “the extent to which the system provides the right kind of functionality so that users can do what they need or want to do” (Preece, Rogers & Sharp, 2002, p. 16). Therefore, the focus is on the set of functions that meet linguists' needs and goals, as described in sections 2, 3 and 4. Table 1 displays the comparative analysis of the five programs with respect to the tasks described in the previous sections.

Field	Tasks	TextSTAT	AntConc	R Commander	WEKA Explorer	GATE Developer
Corpus linguis-tics	- XML processing (XSL)					✔
	- Database management (SQL)					
	- Regexp-based searching	✔	✔			✔
	- Concordancing	✔	✔			✔
	- Word-frequency listing	✔	✔		✔	✔
Statistics	- Descriptive statistics			✔	✔	✔
	- Inferential statistics		✔	✔	✔	✔
	- Graphical data presentation			✔	✔	✔
NLP	- Ngram extraction		✔		✔	✔
	- Stemming				✔	✔
	- Lemmatization		✔			✔
	- POS tagging					✔
	- Phrase chunking					✔
	- Sentence segmentation					✔
Text mining	- Classification				✔	✔
	- Clustering				✔	✔

***Table 1**. Text-data analytics software.*

TextSTAT is a program developed by Matthias Hüning for a simple analysis of texts.[3] The software essentially produces word-frequency lists and concordances from a collection of TXT, DOC, DOCX, ODT, SXW or HTML files. Queries can take the form of literal strings or regular expressions, and concordances are shown in KWIC format. In short, TextSTAT "is geared towards quick and direct corpus queries and easy structuring and surveying of the search results rather than a full-fledged quantitative analysis" (Wiechmann & Fuhs, 2006, p. 125).

AntConc is a toolkit developed by Laurence Anthony for corpus linguistics research.[4] Unlike TextSTAT, AntConc provides a wider range of functionalities, such as:

- extracting ngrams and their frequencies from a collection of TXT, HTML or XML files,
- building a keyword list by comparing the frequency of the words in the custom-made corpus with the frequency of the words in a reference corpus by means of log likelihood or chi-squared,

[3] http://neon.niederlandistik.fu-berlin.de/en/textstat/

[4] http://www.laurenceanthony.net/software/antconc/

- recognizing the collocates of a search term on the basis of mutual information or T-score, and
- making queries with literal strings or regular expressions, where search results can be shown in KWIC format or plotted in a barcode chart.

The strength of this program "lies in its sophisticated text analysis that surpasses creating simple concordances" (Wiechmann & Fuhs, 2006, p. 120).

R is a software environment for statistical computing and graphs display.[5] It is a command-driven system where you can type your programs with a scripting language. In addition to the twelve base packages that make up the R Core, it also has over 7,000 recommended or contributed packages for extending its functionalities.[6] For example, the text-mining framework is provided by the *tm* package (Feinerer, Hornik & Meyer, 2008), which allows for not only pre-processing tasks, such as data import (XML parsing), stemming, stopword removal or POS tagging, but also methods for classification and clustering. The R language is not very intuitive to use, since being command-driven implies a steep learning curve. However, there are also some free graphical-front ends to avoid having to type commands. One of the most stable full-blown alternatives for users with no experience in R and/or programming is R Commander (Fox, 2005). The motivation of implementing this graphical user interface (GUI) was originally to cover the content of Moore's textbook (2000), although now it is much more extensive than required for undergraduate statistics courses.

WEKA (Waikato Environment for Knowledge Analysis) (Hall et al., 2009) is a popular data-mining workbench where researchers can access state-of-the-art techniques in machine learning.[7] WEKA Explorer is the major GUI application in this environment, which provides users with a wide variety of algorithms for classification, clustering and association-rule mining. Datasets can be loaded from files (ARFF and CSV) and databases (through Java Database Connectivity), where SQL queries can only be run against the latter. It is worth noting that, in the case of document categorization, the *StringToWordVector* filter can be applied for pre-processing purposes, where the string attributes in the dataset are converted into a set of features with Boolean, word frequency or TF-IDF values; this filter also offers a number of options to be configured, including tokenization, stopword removal or stemming.

GATE (General Architecture for Text Engineering) (Cunningham et al., 2014) is a suite

5 https://www.r-project.org

6 Contributed packages are available for download from the CRAN (Comprehensive R Archive Network) at http://cran.r-project.org.

7 http://www.cs.waikato.ac.nz/ml/weka/

of tools for developing and deploying software components that process human language.[8] My interest is focused on GATE Developer, a visual integrated development environment designed to support researchers in building, executing and analysing language-engineering applications. To this end, the system is bundled with a set of several hundred plugins, i.e. "prefabricated software building blocks that language engineers can use, extend and customise for their specific needs" (Cunningham et al., 2002, p. 169). In particular, there are three types of components in GATE Developer: language resources (e.g. lexicon, corpus and ontology), processing resources (e.g. tokenizer, tagger, chunker and parser) and visual resources (i.e. GUI). GATE Developer allows users to construct their applications visually by integrating one or more language resources into a sequence of processing resources, where a visual resource is intended to present the output. Therefore, GATE applications have a modular structure, i.e. "when the application is run, the modules are executed in the specified order on the given data" (Bontcheva et al., 2002, p. 226). The set of plugins included in the GATE distribution is known as CREOLE (Collection of REusable Objects for Language Engineering), including processing resources for corpus linguistics (e.g. Alignment, JAPE and XCES), NLP (e.g. ANNIE, RASP, Stanford and SUPPLE) and machine learning (e.g. MAXENT and WEKA). To illustrate, a pipeline can be constructed with the following processing resources: ANNIE Sentence Splitter + ANNIE Tokenizer + RASP2 POS Tagger + RASP2 Morphological Analyser + RASP2 Parser.

On the other hand, learnability refers to the fact that "the system should be easy to learn so that the user can rapidly start getting some work done with the system" (Nielsen, 1993, p. 26). A relevant aspect of learnability is that the user should "not only have the required domain knowledge, but also a general understanding of what tools and functions will be available" (Grossman, Fitzmaurice & Attar, 2009, p. 650). In this regard, the analysis of this dimension is best understood when the programs are viewed along a continuum. At one end, we find a plethora of concordancing programs, such as TextSTAT and AntConc, whose user-friendly GUIs have been tailored to a very limited number of tasks that are not sufficient for complex linguistic research. At the opposite end, we find a few powerful and flexible development environments that are "hard to use because they have one-of-a-kind user interfaces that have a steep learning curve and are easy to forget if not used regularly" (Simons, 1998, p. 10). This is the case of GATE Developer. Although it was designed to make NLP available to a wider linguistic community, this environment is so complex that it is seldom used by corpus linguists; as stated by de Kok, de Kok & Hinrichs (2014), "learning how to use GATE may be prohibitive for novice users". In between there are a number of applications, e.g. R Commander and WEKA Explorer,

8 https://gate.ac.uk

where the effort of learning is proportional to the complexity of the system.

It can be assumed that this divergence between utility and learnability is not a matter of computer illiteracy, since linguists can easily operate point-and-click software. Instead, the problem results from the design itself of this type of software. On the one hand, concordancing software is typically focused on a few tasks oriented to corpus linguistics in isolation, only permeated by some basic statistics, but ignoring the fact that text-data analytics is multidisciplinary. On the other hand, text-mining and NLP programs expose linguists to a large number of unnecessary tasks for their goals, making them experience a cognitive overload that hinders decision making. Therefore, the computerized analysis of texts can become more effective if and only if the software meets the linguist's needs by means of relevant cross-disciplinary tasks. The remainder of this section demonstrates that software design plays a critical role in the pursuit of this objective.

User-centred design supports users in building a "flow state" that helps them achieve their goals. As explained by Csikszentmihalyi (2008), when a person is engrossed in some complex activity, the person is immersed in a flow state. Therefore, the computer application should be designed in such a way that the linguist's flow state does not dissipate. To this end, perceived simplicity is the key to create a GUI that does not “obstruct” users from their goals. Thus, designers of software for linguistic research should avoid making any task appear so hard that linguists get discouraged. One way to do so is by using the “progressive disclosure” technique to break up the difficulty into multiple stages. This technique, which was first introduced by Keller (1987), entails “providing only the information people need at the moment”, so that, “by giving them a little information at a time, you avoid overwhelming them” (Weinschenk, 2011, p. 62). In other words, starting with a minimal GUI, users are guided through a series of steps by showing more of the GUI as they complete each step:

> The user should be walked through a complex task step by step, perhaps because the task is novel, rarely done, or outside the user’s domain knowledge. [...] When the user sees the task unfolding directly in front of him via a dynamically growing UI, he can form a correct mental model of the task more quickly and easily. (Tidwell, 2010, p. 179)

Progressive disclosure is closely related to Tufte's stacked-in-time approach. Particularly, Tufte (1997) described two approaches to deal with the information displayed in the GUI: adjacent in space (i.e. positioning all the elements of the application on the same screen) or stacked in time (i.e. separating the elements by levels of navigation and interaction). Whereas the adjacent-in-space approach gives more control to users as well as expediting the interaction with them, the stacked-in-time approach reduces the complexity of the GUI as well as providing guidance for

novice or occasional users.

The next section describes DAMIEN, a workbench that enables linguists not only to accomplish all of the tasks described in Table 1 but also to perceive that it has a simple GUI where they can learn to use all of the additional features as they appear.

7. DAMIEN

DAMIEN is an online workbench for corpus-based linguistic research that integrates techniques and methods from various fields within text-data analytics.[9] DAMIEN consists of four workspaces (i.e. Corpus, Statistics, Mining and Evaluation), which are further described below. Figure 2 shows the GUI of DAMIEN.

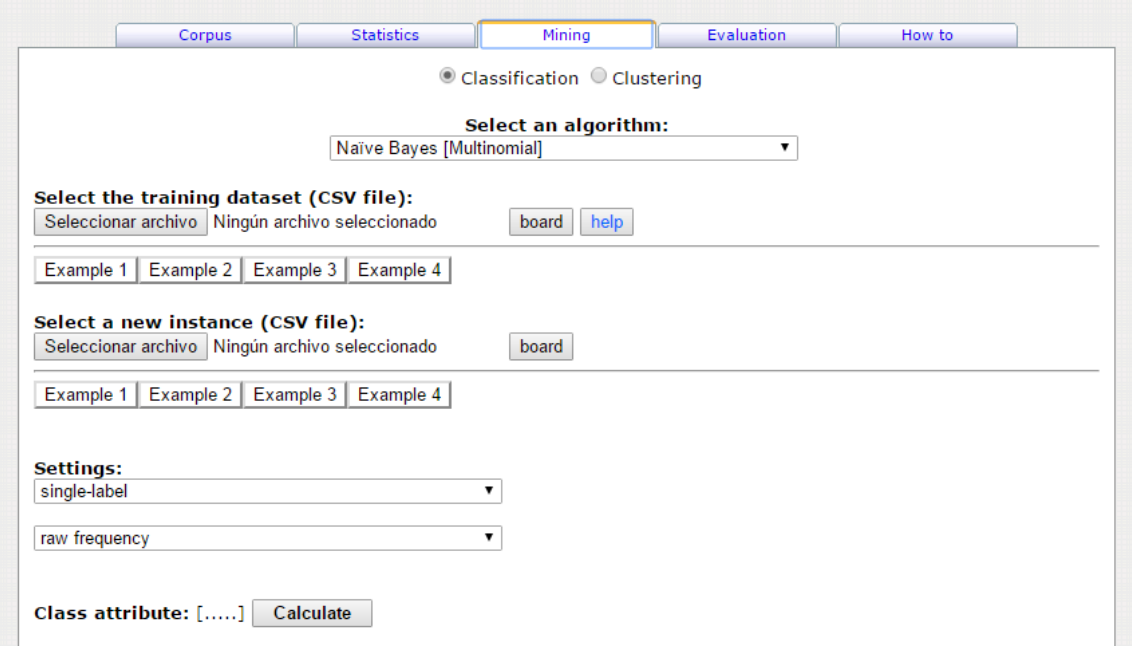

Figure 2. *DAMIEN.*

7.1. Corpus

The Corpus workspace includes four operational modes: Pre-process, Process, Open and Import. In the Pre-process mode, the following tasks can be performed:

- In File Conversion, a collection of PDF or XML files, or of online HTML documents,

[9] DAMIEN, which has been developed in C# with ASP.NET 4.0, is freely accessible from the FunGramKB website (http://www.fungramkb.com/nlp.aspx).

can be converted to plain text; in the case of XML, DAMIEN provides linguists with an XSL editor.

- In File Resizing, a collection of TXT files can be merged into a single file, or a single TXT file can be split into several files, on the basis of a regex-based delimiter or the number of kilobytes per file.
- In Text Modification, DAMIEN searches for regex-based patterns in a corpus and replaces the matches with a user-defined string.
- In Dataset Conversion, datasets in JSON or XML format can be converted to CSV files, or vice versa, which is particularly useful to export DAMIEN datasets to other software.
- In POS tagging, the words in an English corpus are annotated with the Penn Treebank tagset by means of the Brill Tagger.

In the Process mode, linguists can extract data from a corpus (i.e. unstructured data) and deposit the data in a dataset of pipe-delimited values (i.e. structured data). DAMIEN, which has been designed for small and medium-sized collections of TXT documents in English, French, Italian or Spanish, can perform two types of corpus processing. On the one hand, in Raw Processing, the resulting corpus dataset can take the form of an ngram-frequency, ngram-ngram or doc-ngram matrix; in the latter two matrices, each text feature is treated as a separate attribute, and each tuple corresponds to a single ngram or document respectively. The construction of this dataset can be customized by a number of settings, e.g. type of ngram (i.e. unigram, bigram or trigram), form of the ngram (i.e. word form, stem or lexeme) and ngram weight (i.e. absolute frequency, relative frequency, normalized entropy or normalized tf-idf), along with the use of a stopword list, start list and/or threshold value. On the other hand, in Regex-based Processing, the corpus is processed on the basis of a user-defined regexp. Since the notational system of regexps is not easy for the layman to understand, DAMIEN provides linguists with a cheat sheet of frequent symbols and the most recurrent examples used in terminology tasks. DAMIEN can also access some online regexp editors to help newcomers learn this codified method of searching, e.g. RegExper,[10] a tool that transforms any regexp into a SVG graphical representation that is easier to read than its corresponding textual form, and RegExr,[11] a tool to devise and test regexps interactively with syntax highlighting and contextual help. In this way, linguists have an ergonomic GUI that shortens the learning curve of regexps.

Once the dataset has been automatically constructed with Raw or Regex-based

[10] http://regexper.com

[11] http://www.regexr.com

Processing, we have the choice to explore the dataset through SQL queries. This is technically feasible because DAMIEN allows datasets to be mapped into database tables; in particular, datasets can be internally stored as tables in a SQLite database. Moreover, a built-in editor enables linguists to create their own datasets and even manage them: adding, updating or deleting records. It should be noted that the potential of SQL in DAMIEN goes beyond text-feature exploration. For example, thanks to the incorporation of mathematical functions such as LOG(), POW() or SQRT() into SQL,[12] linguists can implement their own statistical metrics. In this way, DAMIEN provides more flexibility for linguistic research than pre-configured corpus programs such as TextSTAT and AntConc.

In the Open mode, a zipped archive containing several datasets can be uploaded with a view to manipulate data through SQL queries, as in the Process mode. Indeed, these two operational modes allow researchers to combine their language resources with those of DAMIEN, which are shown in Table 2.

Type	Description	Language
Word list	Functional and common stopwords	English, French, Italian and Spanish
Word list	Leipzing corpora collection[13]	English, French, Italian and Spanish
Corpus	Leipzing corpora collection	English, French, Italian and Spanish
Lexicon	IATE (InterActive Terminology for Europe)[14]	English, French, Italian and Spanish
Lexicon	OPTED (The Online Plain Text English Dictionary)[15]	English
Lexicon	WordNet[16]	English
Lexicon	WordNet Domains[17]	English

12 That is, logarithm, power and square root respectively.

13 The Leipzig Corpora Collection (Quasthoff, Richter & Biemann, 2006) presents corpora that are similar in size (e.g. one million sentences) and content (e.g. newspapers) in different languages.

14 IATE is the multilingual term database of the European Union. IATE results from the compilation of all the terms used in many subject matters (e.g. politics, finance, education, applied sciences and humanities, among many others) by the translators of the various language services of the EU institutions.

15 OPTED is a public domain English dictionary based on the The Project Gutenberg Etext of Webster's Unabridged Dictionary.

16 WordNet 2.0 (Miller, 1995; Fellbaum, 1998) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

Table 2. *Language resources in DAMIEN.*

In the Import mode, researchers can reuse the terminological data derived from a corpus compiled in DEXTER (Periñán-Pascual, 2015; Periñán-Pascual & Mestre-Mestre, 2015, 2016),[18] an online workbench for the extraction of terms from domain-specific corpora in English, French, Italian and Spanish.

7.2. Statistics

The datasets resulting from work in the Process and Import modes can be analysed by means of descriptive and inferential statistics. Statistical analyses available in DAMIEN are more than enough for the topics that are covered in standard textbooks about statistics for linguists (cf. Butler, 1985; Woods, Fletcher & Hughes, 1986), going far beyond the statistical functionalities of current corpus-based software. In particular, DAMIEN enables researchers to carry out the following tasks:

- Applying measures of position, dispersion and shape, and graphically showing the frequency distribution with a histogram.
- Applying probability functions (e.g. probability density or cumulative distribution) for a given discrete or continuous attribute in the dataset.
- Calculating the confidence interval for the mean with respect to a given confidence-level percentage and the standard error of the mean.
- Conducting hypothesis testing for one sample (e.g. T test or Z test) or two samples (e.g. independent T test, Mann-Whitney-Wilcoxon test, McNemar's test, paired T test, Pearson's chi-squared test, Wilcoxon signed-rank test or Z test) to determine the level of significance, the score of the test and the probability value, in addition to finding out whether or not the null hypothesis is rejected.
- Calculating the correlation (e.g. Pearson's or Spearman's coefficient) or simple linear regression between two attributes in the dataset, where the trend between the attributes is shown in a scatterplot. When the dataset takes the form of an ngram-ngram matrix, statistical significance metrics for co-occurrence analysis can be employed, e.g. log-likelihood ratio or (normalized) pointwise mutual information; in this case, the collocates of a given ngram are ranked according to their co-occurrence values.
- Calculating the similarity between two vectors, represented by two attributes in the dataset, through distance functions such as Cosine, Euclidean, Hamming,

17 WordNet Domains (Magnini & Cavaglià, 2000; Bentivogli et al., 2004) is a lexical resource that assigns domain labels to WordNet synsets.

18 http://www.fungramkb.com/nlp.aspx

Jaccard or Manhattan.

7.3. Mining

DAMIEN can apply methods of classification and clustering to datasets. With regard to the former, decision trees (e.g. ID3 and C4.5), k-nearest neighbour and naïve Bayes (e.g. single- or multi-label multinomial) can be used. With regard to the latter, only k-means is available. It should be highlighted, therefore, that linguists can employ four of the five most popular machine-learning algorithms for classification and clustering, according to Wu et al. (2007).

7.4. Evaluation

Finally, empirical methods for validating linguistic claims require some evaluation technique before data interpretation. Accordingly, DAMIEN allows researchers to derive a confusion matrix (or contingency table) from a dataset that holds the predicted and expected values of the experiment. Moreover, the scores of a wide range of performance metrics are calculated: true positive rate (recall), true negative rate, positive predictive value (precision), negative predictive value, false positive rate, false discovery rate, accuracy, efficiency, error rate, Euclidean distance, F-score, Matthews correlation coefficient (phi coefficient), prevalence, and standard error. The ROC curve is also shown in a scatterplot with 100 cut-off points, and the AUC value determines the quality of the test.

8. Conclusions

In the past four decades, the corpus-based approach to linguistic research has become common practice in the scholarly community. The exploratory analysis of large collections of text samples is usually aided by appropriate software tools, so that relevant usage patterns can be recognized. However, linguists' expectations when conducting corpus-driven experiments can be more effectively met by using software that fully integrates data storage, access and display techniques from corpus linguistics, pre-processing capabilities from NLP, data analysis methods from statistics, and classification and clustering tasks from text mining. Therefore, empirical linguists should be thought of as text-analytics practitioners capable of crossing over various disciplines as needed. Developing software for theoretical and applied linguists also requires that the properties of linguistic data should be taken into account, since these properties inevitably determine specific technological requirements—for example, XML for data exchange and relational databases for data storage. In terms of utility and learnability, a comparative analysis of computer

applications for text-data analytics actually revealed the need to implement a workbench that enables linguists to accomplish all and only the fundamental tasks that they usually want to do in their corpus-based research. For this purpose, DAMIEN was developed to increase not only the efficiency but also the ease of use of this type of specialized software.

About the author

Carlos Periñán-Pascual studied English Language and Literature at Universitat de València and received his Ph.D. degree in English Philology at UNED in Madrid (Spain). Since his doctoral dissertation on the resolution of word-sense disambiguation in machine translation, his main research interests have included knowledge engineering, natural language understanding and computational linguistics. More particularly, his research has been focused on the cognitive and computational treatment of lexical information, constructional meaning, conceptual representation, and reasoning, among many other tasks. Since 2004, he has been the director of FunGramKB, a lexico-conceptual knowledge base, together with a suite of tools, for the automatic processing of language. His scientific production includes over 50 peer-reviewed publications in the fields of linguistics, natural language processing and artificial intelligence. He has been the principal investigator in four funded research projects as well as the chair of the organizing committee in many scientific events, including international workshops and conferences. He is currently an associate professor in the Applied Linguistics Department at Universitat Politècnica de València, Spain.

Acknowledgements

Financial support for this research has been provided by the DGI, Spanish Ministry of Education and Science, grant FFI2014-53788-C3-1-P.

Article history

Paper received: 14 February 2017
Paper received in revised form and accepted for publication: 27 April 2017

References

Aguado de Cea, G., Montiel Ponsoda, E. & Ramos Gargantilla, J.A. (2007). Multilingualidad en una aplicación basada en el conocimiento. *Procesamiento del Lenguaje Natural, 38*, 77-97.

Anthony, L. (2013). A critical look at software tools in corpus linguistics. *Linguistic Research, 30*(2), 141-161.

Antworth, E.L. & Valentine, J.R. (1998). Software for doing field linguistics. In J. Lawler & H. Aristar Dry (Eds.), *Using computers in linguistics: A practical guide* (pp. 170-196). London-New York: Routledge.

Baker, P., Hardie, A. & McEnery, A. (2006). A glossary of corpus linguistics. Edinburgh: Edinburgh University Press.

Bentivogli, L., Forner, P., Magnini, B. & Pianta, E. (2004). Revising WordNet Domains hierarchy: Semantics, coverage, and balancing. In *Proceedings of the 21st International Conference on Computational Linguistics. Workshop on Multilingual Linguistic Resources* (pp. 101-108). Geneva.

Bontcheva, K., Cunningham, H., Tablan, V., Maynard, D. & Saggion, H. (2002). Developing reusable and robust language processing components for information systems using GATE. In *Proceedings of the 3rd International Workshop on Natural Language and Information Systems* (pp. 223-227). Los Alamitos (Ca.): IEEE Computer Society Press.

Burnard, L. (2014). What is the Text Encoding Initiative? How to add intelligent markup to digital resources. Marseille: OpenEdition Press.

Butler, Ch. (1985). Statistics in linguistics. Oxford: Basil Blackwell.

Celko, J. (2004). Trees and hierarchies in SQL for smarties. San Francisco: Elsevier.

Cooper, A., Reimann, R. & Cronin, D. (2007). About face 3: The essentials of interaction design. Indianapolis: Wiley.

Csikszentmihalyi, M. (2008). Flow: The psychology of optimal experience. New York: Harper and Row.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. (2002). GATE: An architecture for development of robust HLT applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 168–175). Stroudsburg (Pa.): Association for Computational Linguistics.

Cunningham, et al. (2014). Developing language processing components with GATE version 8. <https://gate.ac.uk/sale/tao/tao.pdf> [3/10/2016].

De Kok, D., de Kok, D. & Hinrichs, M. (2014). Build your own treebank. In *Proceedings of the CLARIN Annual Conference*. Soesterberg.

Feinerer, I., Hornik, K. & Meyer, D. (2008). Text mining infrastructure in R. *Journal of Statistical Software, 25*(5): 1-54.

Feldman, R. & Sanger, J. (2007). The text mining handbook: Advanced approaches in

analyzing unstructured data. Cambridge-New York: Cambridge University Press.

Fellbaum, Ch. (1998, ed.). WordNet: An electronic lexical database. Cambridge (Mass.): MIT Press.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M. & Klein, M. (2000). OIL in a nutshell. In R. Dieng (Ed.), *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management* (pp. 1-16). Berlin-New York: Springer.

Fox, J. (2005). The R Commander: A basic-statistics graphical user interface to R. *Journal of Statistical Software, 14*(9): 1-42.

Grossman, T., Fitzmaurice, G. & Attar, R. (2009). A survey of software learnability: Metrics, methodologies and guidelines. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems* (pp. 649-658). New York: ACM.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. (2009). The WEKA data mining software: An Update. *ACM SIGKDD Explorations Newsletter, 11*(1): 10-18.

Horton, T., Taylor, C., Yu, B. & Xiang, X. (2006). 'Quite right, dear and interesting': Seeking the sentimental in nineteenth century American fiction. Paris-Sorbonne: Digital Humanities.

Ide, N., Bonhomme, P. & Romary, L. (2000). XCES: An XML-based standard for linguistic corpora. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (pp. 825-830).

ISO (2008). Language resource management - Lexical Markup Framework (LMF). ISO 24613:2008, ISO/TC 37/SC 4. Geneva: International Organization for Standardization.

ISO (2010). Ergonomics of Human-System Interaction - Part 210: Human-Centered Design for Interactive Systems. ISO 9241–210. Geneva: International Organization for Standardization.

ISO (2012a). Language resource management - Linguistic annotation framework (LAF). ISO 24612:2012, ISO/TC 37/SC 4. Geneva: International Organization for Standardization.

ISO (2012b). Language resource management - Morpho-syntactic annotation framework (MAF). ISO 24611:2012, ISO/TC 37/SC 4. Geneva: International Organization for Standardization.

Karp, R., Chaudhri, V., & Thomere, J. (1999). XOL: An XML-based ontology exchange language. Technical Report, SRI International. < http://www.ai.sri.com/~pkarp/xol/xol.html>[3/12/2016].

Keller, J.M. (1987). Development and use of the ARCS model of instructional design. *Journal of Instructional Development, 10*(3): 2-10.

Luyckx, K., Daelemans, W. & Vanhoutte, E. (2006). Stylogenetics: Clustering-based stylistic analysis of literary corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 30-35).

Magnini, B. & Cavaglià, G. (2000). Integrating subject field codes into WordNet. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (pp. 1413-1418).

Athens.

Martínez Cruz, C., Blanco, I.J. & Vila, M.A. (2012). Ontologies versus relational databases: Are they so different? A comparison. *Artificial Intelligence Review, 38*(4), 271-290.

McCormick, S., Lieske, Ch. & Culum, A. (2004). OLIF v.2: A flexible language data standard. The OLIF2 Consortium. <http://www.olif.net/documents/OLIF_Term_Journal.pdf>[22/5/2015].

Meier, J.D., Vasireddy, S., Babbar, A. & Mackman, A. (2004). Improving XML performance. In *Improving .NET Application Performance and Scalability*. Microsoft. <http://msdn.microsoft.com/en-us/library/ff647804.aspx>[15/11/2016].

Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM, 38*(11): 39-41.

Moore, D.S. (2000). The basic practice of statistics. New York: Freeman.

Nielsen, J. (1993). Usability engineering. Boston (Mass.): Academic Press.

Periñán-Pascual, C. (2015). The underpinnings of a composite measure for automatic term extraction: The case of SRC. *Terminology, 21*(2), 151-179.

Periñán-Pascual, C. & Mestre-Mestre, E.M. (2015). DEXTER: Automatic extraction of domain-specific glossaries for language teaching. In *Proceedings of the VII Congreso Internacional de Lingüística de Corpus. Procedia - Social and Behavioral Sciences 198* (pp. 377-385).

Periñán-Pascual, C. & Mestre-Mestre, E.M. (2016). A hybrid evaluation procedure for automatic term extraction. In C. Periñán-Pascual & E.M. Mestre-Mestre (Eds.), *Understanding meaning and knowledge representation: From theoretical and cognitive linguistics to natural language processing* (pp. 261-282). Newcastle: Cambridge Scholars Publishing.

Pitti, D.V. (2004). Designing sustainable projects and publications. In S. Schreibman, R. Siemens & J. Unsworth (Eds.), *A companion to digital humanities*. Oxford: Blackwell. <http://www.digitalhumanities.org/companion/>[17/10/2016].

Plaisant, C., Rose, J., Yu, B., Auvil, L., Kirschenbaum, M.G., Smith, M.N., Clement, T. & Lord, G. (2006). Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In *Proceedings of the 6th ACM/IEEE Joint Conference on Digital Libraries* (pp. 141-150). New York: ACM Press.

Preece, J., Rogers, Y. & Sharp, H. (2002). Interaction design: Beyond human-computer interaction. New York: J. Wiley & Sons.

Quasthoff, U., Richter, M. & Biemann, C. (2006). Corpus portal for search in monolingual corpora. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (pp. 1799-1802). Genoa.

Sampson, G. (2001). Empirical linguistics. London-New York: Continuum.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1): 1-47.

Simons, G.F. (1998). The nature of linguistic data and the requirements of a computing

environment for linguistic research. In J. Lawler and H. Aristar Dry (Eds.), *Using computers in linguistics: A practical guide* (pp. 10-25). London-New York: Routledge.

Thieberger, N. & Berez, A.L. (2012). Linguistic data management. In N. Thieberger (Ed.), *The Oxford handbook of linguistic fieldwork* (pp. 90-118). Oxford: Oxford University Press.

Tidwell, J. (2010). Designing interfaces. Sebastopol (Ca.): O'Reilly.

Tonkin, E.L. (2016). Working with text. In E.L. Tonkin & G. Tourte (Eds.), *Working with text: Tools, techniques and approaches for text mining* (pp. 1-22). Cambridge: Chandos.

Tufte, E. (1997). Visual explanations. Cheshire: Graphics Press.

Weinschenk, S. (2011). 100 things every designer needs to know about people. Berkeley: New Riders.

Wiechmann, D. & Fuhs, S. (2006). Concordancing software. *Corpus Linguistics and Linguistic Theory, 2*(1): 109-130.

Witten, I.H. (2005). Text mining. In M.P. Singh (Ed.), *Practical handbook of Internet computing* (pp. 14/1-14/22). Boca Raton: Chapman & Hall/CRC Press.

Woods, A., Fletcher, P. & Hughes, A. (1986). Statistics in Language Studies. Cambridge: Cambridge University Press.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., & Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge Information Systems, 14*(1): 1-37.