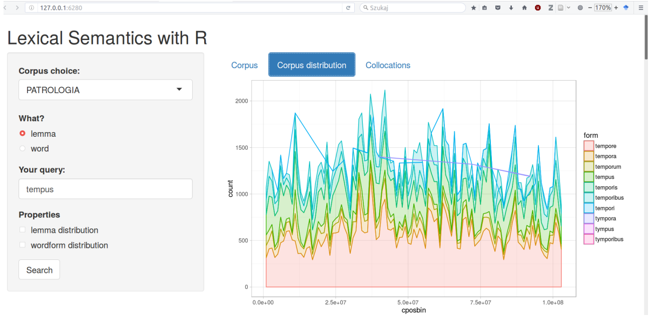
Voces. An R-based Dashboard for Lexical Semantics

Krzysztof Nowak
krzysztof.nowak@ijp-pan.krakow.pl
Institute of Polish Language
Polish Academy of Sciences Krakow, Poland

Introduction

Voces (from Lat. *vox* 'voice', 'word') is an analysis and visualisation dashboard for corpus-based research in lexical semantics. Currently developed as a Shiny application communicating with R session running in the background, *Voces* provides users with possibly exhaustive account of how selected Latin word is distributed across the corpus and what can be told about its meaning. The application is built around a corpus which currently consists of ca. 200M words from texts dating from the Classical era (1 BCE) to the Middle Ages (14th CE). Although *Voces* was originally conceived as a tool of historical semantics research, the application - due to its modular design - may be modified and the code basis can be re-used in new research contexts.



Lexical Semantics with R

Corpus choice: PATROLOGIA

What?
● lemma
○ word

Your query: tempus

Properties
□ lemma distribution
□ wordform distribution

Search

Fig. 1: Voces. User Interface: Word Form Distribution (tempus 'time')

Information computed on a basis of a CWB-indexed corpus is presented to a user through a single-page interface composed of separate widgets arranged in a clear grid layout. Each widget is responsible for displaying in textual or graphical form a clear-cut property of word's distribution or meaning. A heavy use of data visualisation techniques renders *Voces* a convenient tool for exploratory analysis of textual corpora, but the grid layout is also reflection of modular architecture of the application. Each widget is implemented as a separate function which can be extended and adopted by researchers with even limited R programming skills.

Use scenarios

A typical use scenario is triggered when the user specifies a lemma to be looked up. If the search fails, a list of lemmas to choose from is provided. In case of success, neatly separated sections of the dashboard are populated with widgets, each of which corresponds to one sense or distributional property of the word under scrutiny.

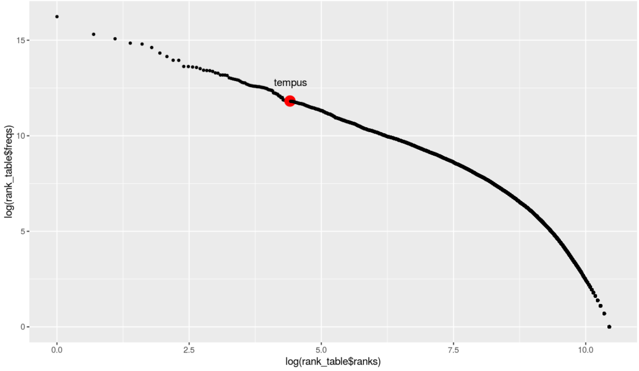


Fig. 2: Voces. User Interface: Frequency Spectrum Plot (Voces. User Interface (tempus 'time')

Word's frequency is summarised as a number of occurrences in the corpus (both raw and p.m.w. counts) and displayed as a highlighted point on a frequency spectrum plot (Baayen 2001). A barplot is provided for investigating change of frequency in subsequent corpus sections. Study of language variation is enabled through widgets presenting word's frequency as a function of such variables as author, work, genre, and – most importantly – time. Users are, therefore, provided with a list of authors who use the word most frequently or a word cloud summarising terms to be found in the titles of works with a particularly frequent use of the word under scrutiny. Genre variation is presented in form of a pie chart, while diachronic dimension - through a bar plot of frequency counts in partitions of the corpus. Diatopic variation study is still to be implemented.

A word's meaning potential can be investigated by means of a set of widgets presenting its contextual properties. The most frequent co-occurrences are enumerated on a simple count list which may be further analysed according to period and genre criteria. A Distributional Semantics Model (Baroni and Lenci 2010) is built from the corpus in order to enable simple meaning computation. Evert's (2014) *wordspace* package and a set of Alain Guerreau's scripts is employed in order to cluster co-occurrences. Similar terms of a looked up word are also computed and then presented in both textual and graphical form.

Users are supported in data and visualisation interpretation through hints which accompany every widget. Their role is to explain not only what the data can mean, but also how the figures were computed, how one can interpret the geometrical properties of a plot, and so on. This, along with the availability of data sets, code snippets, and reports generated on the fly, is what makes *Voces* a tool of reproductive research.

Architecture

Voces was built as a Shiny application (Chang et al. 2016). Its development was greatly facilitated by the availability of a decent documentation and community support (both particularly useful when dealing with framework's complex reactivity model). It turned out soon, however, that it may not be the best choice for web application which has to combine heterogeneous data and non-R code as well. Hence, other solutions are being tested at the moment, those in particular which would provide, for example, more flexible integration of external APIs. The most promising seems to be OpenCPU (Ooms 2014), an application which exposes R session through a RESTful API. This approach allows any application written in some of the less or more popular web development frameworks to easily communicate with an R server instance.

As for the architecture, *Voces* depends on a CQP server instance running in the background which requires corpora to be indexed with the CWB. Communication of the R server with the CWB is assured through the rcqp package (Desgraupes and Loiseau 2012) which offers a set of useful functions providing access to both positional (token-level) and structural (document-level) attributes. Unfortunately, development of this very helpful tool seems to be less active recently and thus *Voces* will soon accept also tabular data as input.

Previous research

Nowadays, corpus linguists may chose from a vast array of free, open source and stable corpus query systems (CQS) which not only allow for efficient indexing of large corpora, but also provide a user-friendly concordance interface and offer out-of-the-box a set of such essential functionalities as collocation lists, simple corpus statistics etc. Both web (CQPweb, NoSketchEngine *etc.*) and desktop applications (TXM *etc.*) are also usually equipped with a less or more intuitive corpus management interface. *Voces*, a dashboard for vocabulary research, is not yet another CQS and has no intention to supersede well-established tools which cannot be easily combated in terms of either robustness or speed. Quite the contrary, the application communicates with the CWB engine and adapts some of the design choices and features of the popular CQS, while hopefully does not inherit their drawbacks.

Unlike the case of the well-known CQS, more emphasis has been put on quick access to multifaceted information rather than on close analysis of occurrences. *Voces* does not attempt, then, to implement some of the features which are traditionally considered an important part of the corpus analytical toolbox, such as concordance sampling, sorting etc. Undoubtedly, the strength of popular CQS lies in their wide applicability: by default, they do not preclude any research scenario. Although agnostic of linguistic theory, *Voces* was originally built for more specific purposes and focuses on semantic properties of the word and its distribution.

What is believed to be one of the main advantages of the present application is that - thanks to its modular architecture - it can be easily extended or adopted by a researcher with even moderate programming skills. In that *Voces* attempts to fill the gap that exists between, on the one hand, fully-blown CQS, which are normally quite conservative when it comes to adding new features, and, on the other hand, single-purpose research workflows built *ad hoc* by researchers. What also distinguishes *Voces* from other CQS is its emphasis on helping users to interpret data. A system of visual and textual hints keeps a researcher informed about where does the data come from, how have they been computed *etc*.

The grid layout is well-known from analytical environment and is especially popular in finances or engineering (Few 2013); in humanities it was adopted, among others, in the Voyant Tools project. It offers a quick insight into otherwise dispersed data and a coherent account of word's properties.

Further research

Voces is currently in an early stage of development. The work focuses on adding new functionalities and plotting types which may sometimes affect application's efficiency. Future work will focus on: (1) optimising user experience; (2) implementing tools for (a) comparative (ie. two-lemma) research and (b) tracking language change; (3) better processing user input (multi-word search).

Bibliography

Baayen, R. H. (2001). *Word Frequency Distributions*. Dordrecht: Kluwer.

Baroni, M., and Lenci, A. (2010). Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics* 36 (4): 673–721.

Chang, W., Cheng, J., Allaire, J. J., Xie, Y., and McPherson, J. (2016). *Shiny: Web Application Framework for R*. https://CRAN.R-project.org/package=shiny.

Desgraupes, B., and Loiseau, S. (2012). *Rcqp: Interface to the Corpus Query Protocol*. http://CRAN.R-project.org/package=rcqp.

Evert, S. (2014). Distributional Semantics in R with the Wordspace Package. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, 110–114. Dublin, Ireland: Dublin City University and Association for Computational Linguistics.

Few, S. (2013). *Information Dashboard Design: Displaying Data for at-a-Glance Monitoring*. Burlingame, CA: Analytics Press.

Nowak, K., and Bon, B. (2015). *Medialatinitas.eu*. Towards Shallow Integration of Lexical, Textual and Encyclopaedic Resources for Latin. In *Electronic Lexicography in the 21st Century: Linking Lexical Data in the Digital Age. Proceedings of the eLex 2015 Conference*, edited by Iztok Kosem, Miloš Jakubíček, Jelena Kallas, and Simon Krek, 152–69. Ljubljana-Brighton: Trojina, Institute for Applied Slovene Studies - Lexical Computing Ltd.

Ooms, J. (2014). The OpenCPU System: Towards a Universal Interface for Scientific Computing through Separation of Concerns. *ArXiv E-Prints*, June.