Embracing the Concept of Data Interoperability in Corpus Tools Development

Laurence Anthony, Waseda University, Japan
Stefan Evert, FAU Erlangen-Nürnberg, Germany

1. Introduction

The adoption of new and sophisticated quantitative techniques in the field of corpus linguistics is often very slow. One reason for this is that most corpus linguists work with an integrated concordancer software package, such as AntConc (Anthony 2018), CQPweb (Hardie 2012), SketchEngine (Kilgarriff et al. 2015), or WordSmith Tools (Scott 2017), and rely on the query and analysis functionality provided by the software package. As a result, the addition of new quantitative techniques depends heavily on the available time of the lead developer. Also, the types of functions added to these tools will be guided by the particular interests of developers and their core user groups. Another reason for the slow adoption of new and sophisticated quantitative techniques is that novel techniques are often published with the inclusion of raw program code written in R or Python, which cannot be integrated easily into the standard concordancers. As a result, only corpus linguists with fairly well-developed programming skills can apply these new techniques. For example, even in the minimal case, they would need to extract data from a concordancer and restructure it into a format required by the analysis algorithm. In more common cases, they would also need to work directly with the raw corpus data in various ways, which can be a cumbersome process even for experienced programmers.

Our goal in this paper is to propose an easy approach for improving the interoperability of corpus tools in order to facilitate the broad and systematic introduction of new and sophisticated methods into the field of corpus linguistics. Interoperability is a term used in computer science to describe the ability of systems to share data and resources with other systems. If data interoperability is to be embraced more thoroughly within corpus linguistics, it must be possible to integrate new techniques with a minimal amount of "investment" from all parties involved: (1) developers of concordancers, (2) developers of analysis tools/algorithms, and (3) end users (corpus linguists). We believe that putting a large share of the burden on one of these groups will never be successful. Rather, we propose encouraging the use of a standardized tabular data format that facilitates data interoperability across a wide range of use cases. In this paper, we focus on an interoperability model that allows a single concordancer software tool to share data with several analysis and visualization tools. However, our proposed model also allows for interoperability between multiple concordancers, allowing for asymmetry in the design.

2. Levels of interoperability

One level of interoperability is the adoption of a common corpus format. Today, various formats exist within the corpus linguistics field, such as TEI and XCES, allowing for the same corpus data to be processed easily with a wide range of tools. However, while these established standards can be useful in enabling users to apply different tools to the same data set, they do not create true interoperability and combine the strengths of different tools. Moreover, an increasing number of corpora are not available for public download in source format at all.

Another level of interoperability is offered by concordancer software tools that provide a plugin API architecture, with novel algorithms implemented as plugins (often simply wrappers around reference implementations). These plugins offer the most convenient and effective workflow for end users and have the potential to allow for a "marketplace" of analysis/visualization plugins (e.g. Rüdiger 2018); a concept often embraced by tools in the field of computer science. On the other hand, plugin development requires substantial commitment from developers of analysis algorithms and a "buy in" from the point of view of the end user. Also, in the context of corpus tool development, different plugins would need to be developed for each of the major concordancers, increasing the workload of developers. In addition, these plugins would probably need to be written in the same programming language as the host concordancer, again limiting some of the functionality that could be introduced.

A third form of interoperability inverts the roles of plugin and host. The concordancer application offers (more or less) standardized Web APIs that allow analysis and visualization tools not only to run queries, but also to obtain quantitative data such as frequency tables and type-token distributions (Kupietz et al. 2018). However, this approach will require a lengthy standardization process to agree on a set of suitable APIs and is only applicable to Web-based concordancers running on public servers. Kupietz et al. (2018) list requirements, possible use cases and a long range of existing standards to build on (JSON, XML, CSV, OpenSearch, Web Language Model API, etc.), but do not propose a concrete API design. A more fully developed solution is the canonical text services API (CTS; Tiepmar & Heyer 2017). However, CTS itself only provides full-text access and should rather be seen as an online version of a common corpus format, albeit without the added value of linguistic annotation. The text mining extension CTS-TM (Tiepmar 2016) offers a specific set of statistics (single-word frequency counts, n-gram frequencies, topic models, etc.), but with limited flexibility for the client.

A final, simpler approach to interoperability is the adoption of a standard *data* format (not *corpus* format) that allows for the interchange of quantitative data. This is the approach we propose in this paper. One of its advantages is that it requires the least amount of commitment from the different groups of software developers, leading to interoperability between a wide range of tools. We also believe that many

applications can be handled successfully with the help of a simple data interchange format (even if the procedure may not be maximally efficient). In addition, since the quantitative data to be exchanged are abstractions, they can legally be provided even for corpora that are not licensed for redistribution. One limitation with this approach is that it provides only limited integration between concordancer and analysis tools. Also, some effort from end users will be required to obtain data from their concordancer in a suitable format before it can be used with other tools. However, the advantages afforded by the adoption of such a scheme hugely outweigh the disadvantages.

3. Our proposal: data interoperability through a tabular data model

Our proposal is to facilitate data interoperability through a standardized tabular data model comprised of multiple tables, which can be serialized into a collection of text files in tab-separated values (TSV) format or stored collectively in a single SQLite database file (Hipp 2019). We call our new data format MTSV (for "multiple TSV"), but envisage SQLite to be the primary format. However, the format also allows for data to be stored in Excel spreadsheets or various other serialization formats.

Many important use cases can easily be represented as single tables, including collocation analysis, multidimensional analysis, and full-vocabulary keyword analysis. However, in other cases a simple table representation can be wasteful even where it is possible. For example, two-way collocation analysis in a single table requires a redundant storage of marginal frequencies (f_1, f_2) and sample size (N) in many table rows. Similarly, keyword analysis requires explicit information on corpus sizes unless frequency counts are provided for all word types. In such cases, the MTSV data format allows for a collection of multiple tables, reducing the memory load and allowing for more elegant data structuring. It should be noted that our proposal is not an entirely new idea. Indeed, Coquery (Kunter 2017) is a tool that relies on such tabular data as an internal format to offer end users more flexibility in their analysis; Kupietz et al. (2018: 22) also note that analysis results can often be serialized as data tables, e.g. in CSV format.

Data types allowed in the MTSV data format are UTF-8 strings, signed 64-bit integers, and IEEE 754 floating-point 64-bit numbers. These data types allow for the modeling of most if not all commonly used data structures for corpus methods, including those needed for Key-Word-In-Context (KWIC) and dispersion plot searches, clustering and n-gram analysis, collocate studies, keyword list creation, as well as network graph visualizations. In order to ensure two-way interoperability, concordancers are expected to provide a link-back API allowing analysis tools to display relevant corpus examples in the concordancer. For this purpose, we introduce a REF data type linking data items to concordance lines; these identifiers are internal to the respective concordancer, but must be valid UTF-8 strings.

To support the MTSV data format, we intend to provide a toolbox for manipulating data sets and converting formats (e.g. converting multiple plain text TSV files and Excel data files to SQLite and vice versa). We also hope to provide tools for common processing tasks, such as data manipulation, filtering, and reformatting, which are written in Python for use in existing tools as well as on the command-line, or as standalone, user-friendly R packages.

References

Anthony, L. (2018). *AntConc* (version 3.5.7) [Computer Software]. Tokyo, Japan: Waseda University. Available from http://www.laurenceanthony.net/software.

Hardie, A (2012). *CQPweb* – combining power, flexibility and usability in a corpus analysis tool. *International Journal of Corpus Linguistics*, 17(3): 380–409.

Hipp, D. R. (2019). *SQLite* (version 3.27.2) [Computer Software]. Charlotte, NC: Hwaci Inc. Available from https://www.sqlite.org/.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V. (2015). The Sketch Engine: ten years on. *Lexicography*, 1, 7-36.

Kunter, G. (2017). Coquery (version 0.10) [Computer Software]. Available from https://www.coquery.org/index.html.

Kupietz, M., Diewald, N., and Fankhauser, P. (2018). How to get the computation near the data: Improving data accessibility to, and reusability of analysis functions in corpus query platforms. In *Proceedings of the LREC 2018 Workshop Challenges in the Management of Large Corpora* (CMLC-6), pages 20–25, Miyazaki, Japan.

Rüdiger, J. O. (2018). CorpusExplorer [Computer Software]. Universität Kassel / Universität Siegen. Available from http://corpusexplorer.de.

Scott, M. (2017). *WordSmith Tools* version 7, Stroud: Lexical Analysis Software.

Tiepmar, J. (2016). CTS text miner – text mining framework based on the canonical text services protocol. In *Proceedings of the 4th Workshop on Challenges in the Management of Large Corpora* (CMLC–4), pages 1–7, Portorož, Slovenia.

Tiepmar, J. and Heyer, G. (2017). An overview of canonical text services. *Linguistics and Literature Studies*, 5(2): 132–148.