COLACIÓN ASISTIDA POR ORDENADOR: ESTADO DE LA CUESTIÓN Y RETOS

COMPUTER-ASSISTED COLLATION: STATE OF THE ART AND CHALLENGES

Helena Bermúdez Sabel
Universidad de Santiago de Compostela
helena.bermudez@usc.es

Resumen

Como advierte el profesor Blecua (1983: 43) en su célebre Manual de Crítica Textual, "la *collatio codicum* es la fase más ingrata y una de las más delicadas de todo el proceso editorial". Tradicionalmente, esta ardua tarea se realizaba cotejando visualmente los diferentes testimonios y anotando las divergencias. A pesar de las ventajas que los métodos computacionales nos ofrecen (entre otras, mayor rapidez y, sobre todo, menos margen para el error) su implementación para esta tarea aún es escasa. En este artículo revisamos los recursos ya existentes, analizando sus ventajas e inconvenientes, para finalizar haciendo una propuesta alternativa a las herramientas que no requieren conocimientos sobre lenguajes de programación.

Palabras clave: Colación. Crítica textual. Variantes textuales. XSLT.

Abstract

The *collatio codicum* is the most unpleasant labor of the editorial process, but it is also one of its most sensitive tasks (Blecua, 1983: 43). Traditionally, this arduous work was carried out by visually comparing the different witnesses and writing down the discrepancies. Although

computational methods offer us multiple advantages (among them, greater efficiency and a tinier margin of error), their application in this area is still scarce. This paper reviews the most commonly used collation tools, analyzing their advantages and shortcomings, and ends up proposing an alternative to *out of the box* resources.

Keywords: Collation. Textual Criticism. Textual Variants. XSLT.

1. LA COLACIÓN, FASE INDISPENSABLE DEL TRABAJO ECDÓTICO

Con una historia que podemos contar en milenios, la Filología es una disciplina heterogénea, compleja, cambiante y difícil de condensar en unas pocas líneas. La disparidad entre las diferentes corrientes filológicas existentes nos imposibilita hablar de una única metodología filológica, pero sí podemos postular el concepto de *edición filológica*, cuya elaboración, forma y objetivos puede variar profundamente, pero que requerirá una fase inicial, tradicionalmente denominada *recensio*, cuya finalidad es la de determinar las relaciones entre los diversos testimonios escritos que transmitan la obra objeto de estudio (Blecua, 1983: 33).

Quizás nuestro trabajo se marque como objetivo la fijación del texto a través de una reconstrucción basada en el examen de las variantes (Tavani, 1983: 9-16); puede que sea la elección del *bon manuscrit* lo que determine nuestra metodología (Bédier, 1922); o tal vez la finalidad de nuestro estudio sea la reconstrucción del *avant-texte* (Bellemin-Noël y Milosz, 1972). A pesar de las diferencias intrínsecas a estos métodos de trabajo, todos ellos parten, necesariamente, del acopio y análisis histórico de los testimonios (*fontes criticae*), así como de la colación o cotejo de esos testimonios entre sí para determinar las variantes y establecer las relaciones existentes entre ellos (*collatio codicum*) (Blecua, 1983: 33-34).

El proceso de colación comienza con la transcripción de todos los testimonios textuales de la misma obra y consiste en la alineación y comparación de estos. Este procedimiento se ha dividido tradicionalmente en las siguientes fases (Blecua, 1983: 43-46; Pérez Priego, 2011: 125-129):

- Atribución de una sigla a cada uno de los testimonios.
- Selección de uno de ellos como base del cotejo (*codex optimus*)[1].

[1] Este texto base también puede denominarse *texto de colación*: "Conosciuti nella loro individualità i testimoni, si procede oltre nella recensio con la collatio, cioè li si mette a confronto parola per parola per quanto riguarda il testo

- La conjunción de diferentes criterios condiciona esta elección: su mayor accesibilidad, legibilidad, su grado de corrección o el nivel de integridad del ejemplar, además de las indicaciones que nos puedan aportar ediciones anteriores.
- Transcripción de este testimonio.
- Numeración de los versos o líneas.
- Cotejo de este testimonio con las demás transcripciones.
- Recolección de todas las variantes, las cuales deben ir acompañadas de la sigla correspondiente, y precedidas del número de verso o línea del texto.

La existencia de ediciones críticas anteriores no disculpa la realización de la colación personal de todos los testimonios. Asimismo, y de manera especial si trabajamos con tradiciones manuscritas, la colación debe ser completa para poder proceder a una eliminación de copias directas (*eliminatio codicum descriptorum*) y detectar los casos de contaminación.

Además de laborioso, este procedimiento es propenso al error, y cualquier equivocación durante esta tarea afectará las consecuentes fases del estudio y los resultados que de ellas se deriven. La informatización de este proceso, es decir, la sistematización del cotejo de formas diferenciando el texto común de las variantes, no solo incrementa exponencialmente la eficiencia en lo que respecta a la rapidez en la obtención de resultados, sino que también nos proporciona la seguridad de que todas las variantes sean detectadas.

Desde el punto de vista epistemológico, el concepto de texto y variante, y por tanto, el proceso de colación, evoluciona dentro del soporte digital. En este sentido, coexisten diferentes modelos textuales (texto como cadena de caracteres, texto como árbol de elementos, etc.), pero es la conceptualización del texto como grafo el que nos ofrece una vía de investigación interesante que explorar en Humanidades (Schmidt y Colomb, 2009: 497-514). Un grafo permite que el usuario pueda modelar las relaciones entre textos o fragmentos textuales con total flexibilidad. Por ejemplo, en lenguaje XML (Extensible Markup Language), el formato más utilizado para la codificación de textos digitales, la información debe ser almacenada como una jerarquía ordenada[2]; sin embargo, un modelo gráfico no exige tal

in esame. Si sceglie un punto di riferimento, testo di collazione, rispetto al quale misurare convergenze e divergenze" (Stussi, 1994: 123).

[2] La estructura en árbol que define el formato XML conlleva el omnipresente problema del solapamiento (Renear *et al*., 1993), que impide la existencia de estructuras compartidas entre las diferentes *ramas* del árbol, es decir, todos los elementos deben estar anidados. A modo de ejemplo, parece *natural* dividir un poemario en poemas que a su vez se componen de estrofas y estas en versos (jerarquía). En este caso, los problemas de solapamiento surgirían al identificar ciertas unidades menores: es habitual que determinados elementos sintácticos no respeten los límites de los versos o, si estamos identificando figuras literarias, el encabalgamiento es un claro ejemplo de una figura definida precisamente por el concepto de solapamiento. Por tanto, esto imposibilitaría la utilización de un elemento que encerrase el contenido de cada verso, pues no podríamos tener esas unidades menores que comenzasen en un elemento *verso* para terminar un elemento *verso* diferente. Evidentemente, existen estrategias para evitar esta limitación, pero acostumbran ir acompañadas de la pérdida de sistematicidad y coherencia con que son identificados los elementos que no presentan problemas de anidamiento. El ejemplo con el que ilustrábamos este problema sería resuelto con facilidad usando un elemento vacío que marcase los saltos de verso,

jerarquía, aunque, al mismo tiempo, permite que esta pueda establecerse si así se desea, convirtiéndolo en un modelo igualmente compatible con XML.

Por otra parte, el modelo de texto como grafo posibilita la distinción entre dos tipos de colación diferentes: uno que utilice un texto base y una segunda tipología en la cual todos los testimonios sean contrastados entre sí sin necesidad de escoger ese texto base. La ventaja de utilizar un texto base es que la ordenación de los testimonios según las concomitancias y especificidades con respecto a ese ejemplar será inmediata. Sin embargo, la colación sin texto base es, conceptualmente, la estrategia de colación más rigurosa cuando se desconoce, aunque sea parcialmente, la historia material de la tradición textual objeto de estudio (Spencer y Howe, 2004: 253-270).

2. COLACIÓN ASISTIDA POR ORDENADOR: PROCEDIMIENTO

En el año 2009 tuvo lugar un taller financiado por los proyectos COST Action 32[3] e Interedition en Gotenburgo (Suecia)[4] en el que participaban los desarrolladores de dos de los programas de colación más utilizados en Humanidades: CollateX[5] y Juxta[6]. El objetivo de este encuentro era el de proporcionar a la comunidad científica una arquitectura de software modular[7] consensuada, con el fin de establecer un marco común en el que estos dos proyectos, así como otros interesados en el software de colación, pudiesen colaborar y trabajar conjuntamente en el desarrollo de las herramientas necesarias. Esta arquitectura es la que se conoce como el modelo Gotenburgo (*Gothenburg model*).

Este modelo identifica cuatro módulos o tareas básicas en las que podemos dividir conceptualmente el proceso de colación: *tokenización*, alineación, análisis y visualización[8].

2.1. Tokenización

La *tokenización* es el proceso mediante el cual, partiendo de archivos digitales que contienen la transcripción de los testimonios, se divide la cadena de caracteres en unidades

pero desde el punto de vista semántico no es lo mismo considerar que un poema está formado por un conjunto de versos, que definir un poema como una estructura que contiene un número determinado de saltos de línea.

[3] Accesible desde http://www.cost-a32.eu/.

[4] "Interedition is a COST Action; our aim is to promote the interoperability of the tools and methodology we use in the field of digital scholarly editing and research". Accesible desde http://www.interedition.eu/.

[5] Accesible desde http://collatex.net/.

[6] Accesible desde http://www.juxtasoftware.org/.

[7] La arquitectura de software comprende los componentes del software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos. Cuando hablamos de arquitectura modular hacemos referencia a que los problemas complejos de esta estructura han sido descompuestos en problemas más sencillos con el fin de facilitar el desarrollo de dicho sistema.

[8] TEIWiki: http://wiki.tei-c.org/index.php/Textual_Variance.

que serán posteriormente alineadas. Denominaremos estas unidades *tokens*. El procedimiento más habitual consiste en utilizar los espacios en blanco como delimitadores de *tokens*, lo que significa que, en la mayor parte de las tradiciones textuales, esta división es equivalente a la segmentación del texto en palabras. No obstante, cualquier otra unidad puede actuar como delimitador y nuestra elección dependerá del grado de detalle que procuramos: carácter, sílabas, frases, versos, líneas, etc.

A pesar de las apariencias, la segmentación en unidades menores puede presentar determinadas complicaciones y ciertas decisiones deben ser tomadas antes de comenzar el proceso. Dependiendo de la tradición textual objeto de estudio, la puntuación, por ejemplo, puede alterar los resultados de la colación. Las estrategias más habituales para lidiar con este problema implican la eliminación de estos elementos gráficos a efectos de la *collatio*, o la desaglutinación de todos los signos de puntuación que se presenten unidos a las palabras, con el fin de convertir estos en *tokens* independientes.

Más dificultades presentan otros elementos como las contracciones, en caso de que estas sean opcionales y los testimonios no coincidan en su uso. Si los archivos que contienen la transcripción de los testimonios están en formato XML, también debemos decidir antes de iniciar la *tokenización* qué papel cumplirá el marcado durante el proceso de colación.

2.2. Regularización y alineación

Una vez que los testimonios se dividen en estas unidades menores, su alineación consiste en encontrar los *tokens* coincidentes e introducir espacios en blanco (*gap tokens*) en caso de omisión, para que las secuencias queden así perfectamente alineadas. A efectos de la colación, tiene que ser el investigador o investigadora quien decida qué criterios se deben cumplir para que dos o más *tokens* coincidan. El concepto de variante juega aquí un papel fundamental y está íntimamente ligado a la tradición textual objeto de análisis, así como a la finalidad del estudio global.

La taxonomía clásica basada en el contenido diferencia cuatro categorías, todas ellas en relación a un texto base: inserción, omisión, mutación y transposición. Otra de las clasificaciones más habituales reconoce dos categorías principales: variante significativa y variante no significativa. Se acostumbra a considerar como significativos los errores, las variantes de lengua y las variantes en el contenido, y serían no significativas las variantes gráficas (alográficas y/u ortográficas). Esta tipología tiene como objetivo identificar los *loci critici* que permitan la clasificación de los testimonios, además de facilitar la elaboración de un aparato crítico.

Así pues, el grado de complejidad de la tradición textual puede alterar profundamente la concepción de variante a efectos de la colación. Por ejemplo, para el análisis de una

tradición profundamente intrincada y con múltiples testimonios, es posible que las variantes no significativas entorpezcan el análisis, por lo que puede considerarse conveniente su neutralización previa a la colación. Este procedimiento sería seguramente impensable, sin embargo, de compararse dos manuscritos de autor, en el que cada divergencia es substancial.

Consecuentemente, con el fin de garantizar que los resultados de la alineación serán los deseados, el investigador debe decidir si neutralizarán ciertas diferencias con anterioridad para que determinados *tokens* sean considerados equivalentes. En este sentido, es fundamental decidir el grado de exactitud de la comparación. Un ejemplo muy claro es el que presentan los alógrafos, pues la variación que afecta a las diferentes formas que puede adoptar un grafema suele carecer de importancia a este nivel. De igual modo, puede ser de nuestro interés ignorar otros tipos de variantes no significativas como la diferencia entre mayúsculas y minúsculas, abreviaturas y formas plenas, variantes ortográficas y puntuación.

2.3. Análisis

La tercera tarea en que dividíamos el proceso de colación es el análisis. El análisis llevado a cabo a esta altura del proceso, es decir, como fase posterior a la alineación, concierne, principalmente, al reconocimiento de transposiciones y repeticiones. Si bien la detección de omisiones o adiciones es relativamente *fácil*, otras casuísticas son computacionalmente más costosas. Consecuentemente, aún hoy en día se invierten recursos académicos con el fin de incrementar la eficiencia de los algoritmos de comparación y alineación ya existentes[9].

Aunque se hayan establecido previamente criterios para implementar una serie de correcciones automáticas, en ocasiones, la información que aporta el conocimiento de la tradición es difícil de sistematizar. Es por eso que, además de la identificación de transposiciones y repeticiones, dentro del módulo analítico se debería incluir la posibilidad de intervención humana para que el investigador pueda modificar los resultados de la alineación directamente.

2.4. Visualización

La ya aludida diversidad de modelos ecdóticos y objetivos del análisis determinará la disposición de los resultados de la colación. En este sentido, dentro de los tipos de visualización para el examen y análisis de variantes conviene diferenciar entre visualizaciones textuales y visualizaciones gráficas.

[9] Para profundizar sobre los algoritmos de comparación y alineación más utilizados véase Nassourou (2013: 28-33) y, de manera especial, Dekker y Middlell (2011).

Con mucha probabilidad, como fase previa a la elaboración de una edición crítica, se requerirá una tabla de variantes que facilite la identificación de errores conjuntivos y separativos. Sin embargo, si nuestro objetivo es una edición sinóptica, buscaremos que el resultado final presente un modelo similar a este modelo ecdótico: las lecciones de todos los manuscritos confrontadas.

Además de la disposición visual, también es importante el formato de salida (*output*). En la elaboración de una edición digital es fundamental que el formato de los resultados sea compatible con el soporte digital del que dependa el desarrollo de la misma (JSON, TXT, LaTeX, XML o HTML, entre otros). En esta línea, es muy probable que requiramos que los resultados estén convenientemente marcados para continuar con el procesamiento de los mismos. De hecho, conscientes de esta necesidad, algunos de los softwares de colación que abordaremos en el siguiente apartado ofrecen como posible *output* el método de segmentación en paralelo de la *Text Encoding Initiative* (TEI Consortium, 2016b).

Las visualizaciones gráficas, concretamente, el denominado grafo de variantes, se presenta como un formato especialmente adecuado para la ilustración de resultados cuando la variación es en sí misma el objeto de estudio.

Un grafo de variantes es una manera de representar la variación textual a través de un *grafo acíclico dirigido* (Schmidt y Colomb, 2009: 497-514; Andrews y Mace, 2013: 504-521). Un grafo de este tipo tiene un vértice (o nodo) inicial y un vértice final. Cada testimonio se representa mediante un único trayecto que une esos dos vértices y a través del cual se adhieren secuencialmente las lecciones que conforman ese testimonio.

Podemos reducir a dos los modelos de grafo de variantes utilizados en Humanidades. El primero de ellos, propuesto por Schmidt y Colomb (2009: 497-514) a quien debemos la primera teorización sobre este tipo de visualización para la representación de la variación textual, coloca los identificadores de los testimonios, así como el texto común en las aristas o enlaces (es decir, las líneas que unen los vértices). Por su parte, los diferentes vértices del grafo representan los puntos del texto en que comienza o termina la divergencia.

El segundo modelo (véase Figura 1) presenta las lecciones como los vértices del grafo, mientras que los identificadores de los textos son representados en las aristas, siendo este el trayecto que hay que seguir para leer la secuencia textual de cada testimonio (Andrews y Mace, 2013: 504-521). Este modelo es, actualmente, el más utilizado, pues de él parten los grafos de variantes generados con herramientas como CollateX o Stemmaweb[10], de las que hablaremos a continuación.

[10] Accesible desde https://stemmaweb.net/.

3. COLACIÓN ASISTIDA POR ORDENADOR: SOFTWARE[11]

Si nuestro objetivo es ilustrar gráficamente los resultados de la colación, las dos herramientas más adecuadas para esta finalidad son Stemmaweb y TRAViz[12], aunque CollateX también genera este tipo de *output*.

Stemmaweb se presenta como un conjunto de herramientas y recursos para una *estematología empírica*. Parte del texto previamente cotejado, pero admite varios formatos: texto alineado en una hoja de cálculo (pudiendo estar esta en los formatos CSV, TSV con extensión .TXT, XLS o XSLX), TEI-XML siguiendo el modelo de segmentación en paralelo (TEI Consortium, 2016b) o el denominado "double end-point attachment method" (TEI Consortium, 2016a), o un archivo GraphML (formato XML para grafos)[13].

Si bien no es fácil justificar la presencia de Stemmaweb en este artículo por no tratarse específicamente de una herramienta de colación, difícilmente podemos hablar de grafos de variantes sin hacer mención a este recurso, dada la calidad filológica de las visualizaciones por él producidas (Andrews y Mace, 2013: 504-521). Stemmaweb va un paso más allá que otras herramientas permitiendo la anotación y la diferenciación del tipo de variante. La Figura 1 es un pequeño fragmento de uno de los grafos de variantes disponibles en la página del proyecto. Este grafo representa una tradición textual conformada por quince testimonios distintos. Como se puede observar, las diferentes variantes se relacionan entre sí para informar de su tipología (léxica entre *desire* y *disease* y gramatical entre las otras)[14].

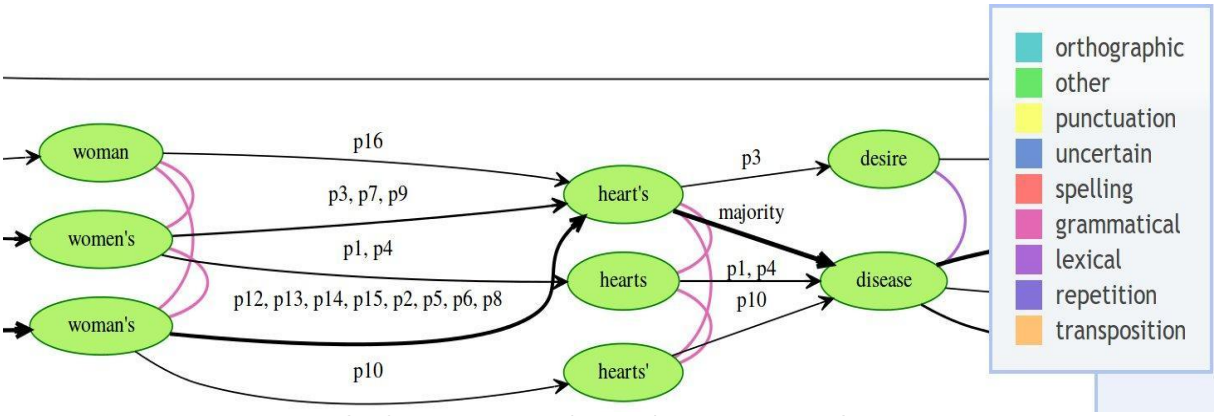

Figura 1. Grafo de variantes (parcial) generado por Stemmaweb.

[11] Nos limitamos a incluir software específicamente concebido para su uso en la investigación filológica. Se excluyen, pues, otros programas como GNU Wdiff (https://www.gnu.org/software/wdiff) de uso más genérico.
[12] Accesible desde http://www.*traviz*.vizcovery.org/.
[13] Este último muy difícilmente puede ser elaborado de manera manual, por lo que presupone el uso previo de otras herramientas de colación como CollateX, la cual también genera este formato.
[14] Además de ofrecer la posibilidad de representar visualmente el tipo de variante, Stemmaweb disponibiliza un diálogo para la incorporación de anotaciones que permitan identificar las variantes con mayor relevancia estematológica.

TRAViz es una librería JavaScript desarrollada dentro del proyecto de Humanidades Digitales *eTraces*[15], que alinea las diferentes versiones de un texto generando un grafo de variantes.

TRAViz se presenta como una herramienta más flexible en cuanto al diseño del grafo, dando la oportunidad al usuario de decidir sobre diferentes aspectos de la visualización. Es este motivo el que lleva a sus creadores a defender la mayor calidad visual de esta herramienta con respecto a los grafos generados por CollateX o Stemmaweb (Jänicke *et al.*, 2015: i85). Sin embargo, desde el punto de vista filológico, TRAViz no soporta el tipo de anotación que Stemmaweb ofrece ni la oportunidad de incluir información sobre la tipología de variante.

Como se puede observar en la Figura 2, en lugar de etiquetar las aristas, TRAViz utiliza diferentes colores para la identificación de los testimonios. Además, el tamaño de los nodos (el texto de la variante) es directamente proporcional a la representatividad de este dentro de la tradición.

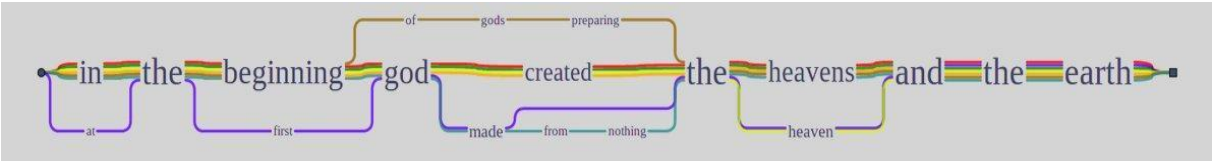

Figura 2. Grafo de variantes generado por TRAViz.

Mencionado en varias ocasiones durante este artículo, CollateX es, seguramente, el software de colación más eficiente del mercado. Cuenta con una versión en Java, además de disponibilizar una aplicación web y un módulo de Python[16]. Concebido específicamente para la investigación filológica, la principal característica que distingue a CollateX de otros softwares es la capacidad de intervención que ofrece al usuario, quien puede controlar directamente cada uno de los aspectos de la colación.

CollateX trabaja eficazmente con diferentes formatos: texto simple, XML y JavaScript Object Notation (JSON). Este último ofrece múltiples ventajas dada su flexibilidad[17] y su estructura resulta tan intuitiva y legible que fácilmente podemos transformar cualquier otro

[15] Accesible desde http://etraces.e-humanities.net/.
[16] El paquete CollateX de Python está disponible en https://pypi.python.org/pypi/collatex. Para un tutorial que incluye su instalación y uso visítese http://collatex.obdurodon.org/.
[17] Un archivo JSON como *input* en CollateX consiste en un único objeto raíz que contendrá los datos de la transcripción. El objeto raíz requiere una propiedad, llamada *witness*, que englobará las diferentes versiones que se desean cotejar. El valor de esta propiedad es, a su vez, un *array* (lista) de objetos y cada uno de ellos representará una de las versiones, por lo que se requiere que tengan un identificador único dentro de la propiedad *id*, siendo esta obligatoria. Este identificador será usado para referirse a una determinada versión/testimonio en el *output*. Como decíamos, como *input*, JSON es un formato muy flexible y el contenido textual puede representarse de diferentes maneras. En el ejemplo (figura 2) presentamos este contenido *tokenizado*, con un propiedad *n* (opcional) que contiene la versión normalizada. No obstante, además de las dos propiedades definidas, *t* y *n*, los objetos que representan los *tokens* pueden contener el número de propiedades que deseemos, pudiendo ser estas consideradas (o no) a efectos de la colación.

formato a JSON (véase Figura 3). De hecho, en nuestra experiencia, una de las estrategias de trabajo más eficientes implica la *tokenización* de los archivos XML a este formato, incluyendo el marcado como propiedades adicionales que posteriormente serán recuperadas en los resultados de la colación. Es en este nivel, además, donde podemos llevar a cabo cualquier tipo de normalización y regularización de grafías, siendo esta una de las propiedades que hacen de CollateX un programa superior a otros existentes.

CollateX nos proporciona tanto formatos de salida textuales como gráficos (grafo de variantes). Entre los primeros genera una tabla de variantes en formato HTML o texto simple. Otro de los *outputs* textuales disponibles es XML e, incluso, XML-TEI siguiendo el modelo de segmentación en paralelo (TEI Consortium, 2016b). Conviene destacar que la posibilidad de crear un objeto JSON con los resultados conlleva esa misma flexibilidad que admirábamos como *input*. Entre los formatos gráficos posibles podemos escoger entre GraphViz DOT o GraphML, pero destacamos este último pues permite que el grafo de variantes pueda ser importado por otras herramientas específicas para el análisis de grafos que permiten realizar cálculos y visualizaciones más complejas, como Gephi[18] o el ya aludido Stemmaweb.


```
{"witnesses": [
  {
    "id": "A",
    "tokens": [
      {
        "t": "A",
        "n": "a"
      },
      {"t": "mia"},
      {
        "t": "sennor",
        "n": "senhor"
      },
      {"t": "que"},
      {"t": "por"},
      {"t": "mal"},
      {"t": "destes"},
      {"t": "meus"}]
  },
  {
    "id": "B",
    "tokens": [
      {
        "t": "A",
        "n": "a"
      },
      {
        "t": "mha",
        "n": "mia"
      },
      {
        "t": "sẽh˜",
        "n": "senhor"
      },
      {
        "t": "q̃",
        "n": "que"
      },
      {
        "t": "p˜",
        "n": "por"
      },
      {"t": "mal"},
      {"t": "destes"},
      {
        "t": "meg",
        "n": "meus"
      }
    ]}
]}
```


Figura 3. Ejemplo de *input* en JSON incluyendo la normalización de abreviaturas y mayúsculas.

[18] Accesible desde https://gephi.org/.

Juxta es una herramienta de código abierto que también permite comparar y cotejar varios testimonios para crear un único objeto textual. Es una herramienta *out of the box*, es decir, de fácil instalación y uso, que además del cotejo, ofrece varios recursos adicionales con el fin de auxiliar en las tareas relacionadas con la crítica textual.

Juxta trabaja con textos digitales tanto en XML (y TEI) como texto simple, y, al igual que CollateX, también genera diferentes tipos de visualizaciones, todas ellas muy adecuadas para el análisis textual. Así pues, podemos generar fácilmente una edición sinóptica en que las diferentes versiones se muestran paralelamente; un mapa de calor con todas las variantes textuales que permite al usuario localizar la variación con respecto al texto base; un histograma, especialmente útil si trabajamos con documentos largos, pues muestra la densidad de la variación con respecto al texto base y funciona como guía para buscar variantes específicas. También podemos crear un archivo HTML con un aparato crítico que sigue el formato línea, lectura del texto base, variante y testimonio o testimonios que contienen esa variante[19].

Además de estos múltiples formatos de salida, Juxta incluye una interfaz que permite la anotación con el fin de que el investigador registre libremente cualquier tipo de información adicional sobre el cotejo. Otro de los recursos que incorpora esta herramienta es la posibilidad de inclusión de imágenes facsimilares.

Algunas de estas herramientas también están disponibles en la API online Juxta Commons[20], la cual no requiere la instalación de ningún módulo y es altamente intuitiva.
En lo que respecta al número de testimonios, Juxta permite añadir tantas versiones como sean necesarias a un set de comparación, y coteja todos los testimonios entre sí. Lo que esto significa es que, durante el análisis posterior podemos cambiar el texto base a nuestro antojo, generando los nuevos resultados automáticamente.

Juxta también permite llevar a cabo un número reducido de regularizaciones: puntuación, mayúsculas y minúsculas y espacios en blanco. Además, si el *input* son archivos XML que incluyen como parte del marcado información sobre revisiones (inserciones o cancelaciones), estas pueden ser tratadas individualmente, por lo que cada una de ellas puede ser aceptada o rechazada como parte del material textual que será cotejado.

Después de haber analizado el control que CollateX ofrece sobre los diferentes módulos en los que se dividía el problema de la colación (*tokenización*, alineación, análisis y visualización), las limitaciones de Juxta se hacen más evidentes. Las regularizaciones que se pueden realizar son mínimas, y pueden resultar especialmente restrictivas si trabajamos con

[19] En nuestra opinión, esta es la visualización menos eficiente que ofrece esta herramienta. Su limitación se debe al hecho de que el lema, la lección principal, es siempre el texto base, y dependiendo de la clase de materiales textuales con que trabajamos esta visualización puede convertirse en completamente inútil.
[20] Accesible desde http://juxtacommons.org/.

lenguas o períodos sin normas ortográficas: en estos casos, la variación gráfica acostumbra a ser el tipo de variante más habitual y puede interferir negativamente en los resultados de la colación.

Además de la posibilidad de controlar las revisiones textuales, no hay mucho que se pueda hacer para conservar el marcado. Si nuestros archivos XML contienen notas, simultaneidad de criterios, u otros elementos que deban ser visualmente diferenciados, Juxta no será la herramienta de publicación más adecuada para la colación y visualización de nuestros textos[21].

Por último, debemos advertir de la dificultad que conlleva extraer los materiales generados por Juxta de su propia interfaz. A pesar de la utilidad de las diferentes visualizaciones que genera, no está concebida su inclusión en un medio diferente (por ejemplo, una página web), por lo que la única opción a nuestro alcance pasa por la utilización de Juxta Commons e incluir un enlace con nuestro proyecto subido en esa API.

4. CONCLUSIÓN: ¿ES XSLT/XQUERY UNA ALTERNATIVA VIABLE?

Computacionalmente, la colación es un problema complejo. Hemos analizado brevemente diferentes herramientas y todas ellas demandan distintos tipos de sacrificio al investigador, sea en la intensidad de la curva de aprendizaje sea en la calidad de los resultados.

Evidentemente, es el usuario quien debe decidir qué sacrificios merecen la pena. Con una tradición compleja y con objetivos muy específicos, quizás el esfuerzo que implica el dominio de una herramienta como CollateX sea recompensado. Bajo otras circunstancias, un software intuitivo que nos ofrece resultados inmediatos puede ser la mejor opción. En nuestro caso particular, la tradición textual objeto de nuestro estudio, la lírica profana gallego-portuguesa, así como los objetivos concretos del análisis, eso es, un estudio muy pormenorizado de la variación lingüística, demandaban un control exhaustivo sobre todaos las fases de procesamiento de los textos. Es por eso que desde un primer momento decidimos basar nuestra metodología en el uso de XSLT y XQuery.

Así pues, utilizamos XSLT para la *tokenización*, alineamiento y análisis y XQuery para un análisis complementario y para la publicación de resultados.

Partimos de la transcripción de los manuscritos en texto simple, y utilizamos expresiones regulares para una marcación automática de los elementos estructurales de las *cantiga*s. Una vez que registramos cada una de estas unidades textuales con su

[21] En este sentido, la *Versioning Machine* (Schreibman, 2016) es más eficiente. Este recurso no ha sido incluido en este artículo por no tratarse específicamente de una herramienta para la colación, pues su funcionamiento depende de que el usuario haya marcado los textos previamente con las variantes perfectamente alineadas.

correspondiente identificador, procedemos a cotejar las diferentes versiones de la misma *cantiga*, tokenizando cada verso según los espacios en blanco presentes y comparando si cada uno de esos *tokens* es igual o no al *token* situado en la misma posición en todos los testimonios. Si el *token* es el mismo, se etiqueta como texto común, si es diferente, elaboramos una serie de cálculos que categorizan (con bastante exactitud) el tipo de variante. Si uno de los testimonios tiene un número de *tokens* superior a los otros, introducimos elementos vacíos que identifican la omisión (<gap/>).

Los puntos más débiles de esta estrategia son evidentes. Un método como el que acabamos de describir sería difícilmente sustentable con tradiciones que presenten mayores divergencias textuales, pues el índice de exactitud más alto se produce cuando la *cantiga* presenta el mismo número de versos y el mismo número de *tokens* en todos los testimonios. Además, un sistema que depende tanto de la posición de los elementos tampoco reconoce transposiciones ni repeticiones.

A pesar de la homogeneidad que presenta la lírica profana gallego-portuguesa, esta no nos evita la realización de una serie de comprobaciones previas a la *tokenización*. Por poner un ejemplo, debemos controlar a qué se deben las divergencias en el número de versos introduciendo elementos vacíos en la posición adecuada para que la equivalencia entre las diferentes versiones sea total.

La gran ventaja de este método es la identificación de variantes hecha a la medida de nuestros datos. La gran desventaja es que no se aprovecha de la eficiencia de los algoritmos de colación ya desarrollados. Volviendo sobre la descripción de las herramientas previamente presentadas que hacen uso de estos algoritmos, parece evidente que la solución pasa por la combinación de CollateX con XSLT. Sin embargo, las librerías para el procesamiento de XML en Python, como LXML[22], funcionan con XSLT 1.0, lo que significa que las funciones más avanzadas de este lenguaje de transformación no estarían a nuestra disposición. Es aquí donde vemos el principal reto para la colación asistida por ordenador de archivos XML que contengan complejas estructuras de marcado. Creemos, pues, que invertir en el desarrollo de métodos que combinen ambas tecnologías con una compatibilidad máxima es imprescindible para consolidar el uso de herramientas informáticas para la colación de textos con objetivos filológicos.

[22] Accesible desde http://lxml.de/.

REFERENCIAS BIBLIOGRÁFICAS

ANDREWS, T.L. y MACE, C. (2013). "Beyond the Tree of Texts: Building an Empirical Model of Scribal Variation through Graph Analysis of Texts and Stemmata". *Literary and Linguistic Computing*, 28.4, 504-521.

BÉDIER, J. (1922). *La Chanson de Roland; publiée d'après le manuscrit d'Oxford et traduite par Joseph Bédier*. Paris: L'Edition d'Art, H. Piazza.

BELLEMIN-NOËL, J. y VLADISLAS DE LUBICZ MILOSZ, O. (1972). *Le texte et l'avant-texte: les brouillons d'un poème de Milosz*. Paris: Larousse.

BLECUA, A. (1983). *Manual de Crítica Textual*. Madrid: Castalia.

DEKKER, R.H. y MIDDELL, G. (2011). *Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements*. Copenhagen: University of Copenhagen. Recuperado de https://www.bibsonomy.org/bibtex/1a080f7c10e92f65e2e807524a389a590/c.schoech el 21/04/2017.

INTEREDITION DEVELOPMENT GROUP (2013). *CollateX.* Recuperado de http://collatex.net/ el 21/04/2017.

JÄNICKE, S. (2014). *TRAViz*. Recuperado de http://www.traviz.vizcovery.org/ el 21/04/2017.

JÄNICKE, S., GESSNER, A., FRANZINI, G., TERRAS, M., MAHONY, S. y SCHEUERMANN, G. (2015). "TRAViz: A Visualization for Variant Graphs". *Digital Scholarship in the Humanities*, 30, suppl 1, i83-i99.

NASSOUROU, M. (2013). *Computer-Supported Textual Criticism: Theory, Automatic Reconstruction of an Archetype.* BoD: Books on Demand.

PÉREZ PRIEGO, M. A. (2011). *La edición de textos*. Madrid: Síntesis.

PERFORMANT SOFTWARE SOLUTIONS LLC (2014). *Juxta*. Recuperado de http://www.juxtasoftware.org/ el 21/04/2017.

RENEAR, A., MYLONAS, E. y DURAND, D. (1993). "*Refining Our Notion of What Text Really Is*". Recuperado de http://cds.library.brown.edu/resources/stg/monographs/ohco.html el 21/04/2017.

SCHMIDT, D. y COLOMB, R. (2009). "A Data Structure for Representing Multi-Version Texts Online". *International Journal of Human-Computer Studies*, 67.6, 497-514.

SCHREIBMAN, S. (2016). *Versioning Machine 5.0*. Recuperado de http://v-machine.org/ el 21/04/2017.

SPENCER, M. y HOWE, C. (2004). "Collating Texts Using Progressive Multiple Alignment". *Computers and the Humanities*, 38.3, 253-270.

STUSSI, A. (1994). *Introduzione agli studi di filologia italiana*. Bologna: Il Mulino.

TAVANI, G. (1983). “Appunti in margine al problema dell’edizione critica”. *Studi di Letteratura Ispano-Americana*, 15-16, 9-16.

TEI CONSORTIUM (2016a). “12.2.2 The Double End-Point Attachment Method”. Última actualización 08/12/2016. En *TEI P5: Guidelines for Electronic Text Encoding and Interchange,* TEI Consortium. Recuperado de http://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html - TCAPDE el 21/04/2017.

TEI CONSORTIUM (2016b). “12.2.3 The Parallel Segmentation Method”. En *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, TEI Consortium. Recuperado de http://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html el 21/04/2017.