Propuesta recibida el 1 de noviembre 2017 y aceptada para publicacin el 26 de diciembre 2017.

GECO, un Gestor de Corpus colaborativo basado en web

GECO, A Web-based Collaborative Corpus Manager

Gerardo Sierra Martínez
Universidad Nacional Autónoma de México
gsierram@iingen.unam.mx

Julián Solórzano Soto
Universidad Nacional Autónoma de México
jsolorzanos@iingen.unam.mx

Arturo Curiel Díaz
Universidad del Bío-Bío, Chile
me@arturocuriel.com

Resumen

Este artículo presenta GEstor de COrpus (GECO), un software de gestión de corpus en línea que permite a los usuarios subir colecciones de documentos y volverlos corpus digitales. En el sistema, los corpus pueden ser procesados por otras aplicaciones, las cuales están implementadas como módulos integrados a la infraestructura de GECO. En este documento se describen a detalle sus características, así como la funcionalidad del generador de concordancias desarrollado en torno a él.

Palabras clave

gestor de corpus, generador de concordancias, software de anotación

Abstract

This paper presents GEstor de COrpus (GECO), an online corpus management software that lets users upload document collections and turn them into digital corpora. Inside the system, corpora can be further processed by other applications, which are implemented as modules over the GECO framework. In this document, GECO's features are described in detail, as well as the functionality of a concordance generator module developed on top of it.

Keywords

corpus manager, concordance generator, annotation software

1 Introducción

Los gestores de corpus son aplicaciones especializadas que permiten a los usuarios cargar archivos de texto y ejecutar consultas (Ntoulas et al., 2001). Están diseñados para manejar grandes cantidades de información y vienen normalmente con funcionalidades adicionales, tales como el cálculo de estadísticas del corpus.

A grandes rasgos, lo que constituye la parte de gestión de corpus del software es aquella que administra los documentos: permite a los usuarios añadir o eliminar textos de una colección y permite anotar los documentos con diversos metadatos (tal como autor, género, época, tema, etc.). Sin embargo, el aspecto de creación de corpus no suele ser el enfoque principal de los gestores de corpus sino las aplicaciones que proveen, como los generadores de concordancias, que son sistemáticamente incluidos en los sistemas más populares (Kouklakis et al., 2007).

En este artículo se presenta a detalle un sistema de gestión de corpus llamado GECO[1] (GEstor de COrpus). Se pone énfasis en la descripción de los principios de diseño en los que está basado GECO, y como eficientiza el proceso de creación de nuevos corpus.

La sección 2 hace una breve comparación con software existente similar. La sección 3 describe los objetivos de diseño y la filosofía de GECO. En la sección 4 se explican a detalle las funcionalidades del software. La sección 5 presenta un ejemplo detallado de cómo se integra al sistema un módulo aplicativo, un generador de concordancias. En la sección 6 se presenta una descripción más técnica del funcionamiento del sistema. Finalmente, en la sección 7 se presentan algunas conclusiones y trabajo futuro.

2 Software relacionado

Hoy en día existe una gran variedad de software de gestión de corpus en el mercado, cada uno con diferentes capacidades para análisis cuanti-

[1] Está disponible en la página: http://www.corpus.unam.mx/geco/.

DOI: 10.21814/lm.9.2.256

tativos del texto (Manning & Schutze, 1999), como lo son anotación de metadatos, generación de concordancias y cálculo de colocaciones (Kouklakis et al., 2007). Los gestores de corpus pueden proporcionar desde listas de palabras simples (Anthony, 2005), hasta robustos marcos de trabajo de desarrollo capaces de usar el procesamiento del lenguaje natural (NLP) para aplicaciones concretas (Kilgarriff et al., 2015). Por ejemplo, el proyecto Corpógrafo (Sarmento et al., 2006) es un gestor de corpus que utiliza técnicas de procesamiento de lenguaje para extracción de términos y extracción de relaciones léxicas. Asimismo, LinguaKit es una herramient multilingüe para el análisis, la extracción, antotación y corrección lingüística (Gamallo & Garcia, 2017).

En los siguientes párrafos se describen las características de dos gestores de corpus bien conocidos y sobre los cuales están construidos muchos otros sistemas: el IMS Open Corpus Workbench (CWB) y Manatee. Ambas herramientas ofrecen poderosos lenguajes de consulta e implementan una arquitectura similar a la que otros gestores de corpus han usado antes (Christ, 1994) y proporcionan al usuario una interfaz gráfica basada en web. Las siguientes descripciones se dan con fines comparativos.

El IMS Open COrpus Workbench (CWB)

El CWB es una colección de herramientas de código abierto para la gestión de corpus y anotación lingüística (Evert & Hardie, 2011). Está diseñado para manejar grandes cantidades de información eficientemente. El CWB puede codificar e indexar corpus de cualquier tamaño. Para poder ser procesados adecuadamente, los archivos de entrada deben ya estar segmentados y anotados. Para esto, el CWB proporciona dos tipos de anotaciones: los atributos posicionales (atributos-p) y los atributos estructurales (atributos-s), ambos expresados como etiquetas XML. A grandes rasgos, un atributo-p es un atributo a nivel palabra: ligan una posición del corpus a un valor. Por ejemplo, las palabras en sí son atributos-p que corresponden a valores que aparecen en una posición única dentro del corpus. Por el otro lado, los atributos-s son atributos ligados a rangos de posiciones: permiten asociar etiquetas a secuencias de palabras en cualquier parte del documento. Por ejemplo, las colocaciones son anotadas como atributos-s.

La función más importante de CWB es el procesador de consultas de corpus (CQP, por sus siglas en inglés), un sistema de generación de concordancias con un lenguaje de consulta muy flexible que permite ingresar complejos patrones de búsqueda de palabras o frases. Soporta expresiones regulares y es capaz de encontrar atributos de palabras (por ejemplo, etiquetas de parte de la oración) así como elementos entre etiquetas estructurales.

Existe un paquete separado llamado CQPWeb (Hardie, 2012) que proporciona una interfaz web para el software, permitiendo al usuario instalar nuevos corpus y utilizar el CQP desde el navegador web. La Figura 1 muestra un ejemplo de la interfaz usando el Brown Corpus (Francis, 1965).

CQPWeb ofrece otras funcionalidades adicionales no incluidas en CWB, tales como colocaciones y ordenamiento de los resultados de búsqueda, metadatos y otros. Puede ser instalado tanto localmente (en computadoras individuales) como en un servidor público donde los usuarios pueden registrarse y acceder a los corpus instalados usando una cuenta. Para ligar la interfaz con los corpus, el administrador puede apuntar el sistema a un recurso ya existente o bien subir los archivos directamente desde la interfaz web, lo cual creará inmediatamente el índice. La interfaz solicita la descripción de la estructura del corpus, dada a partir de los atributos-p y atributos-s. Una vez que los corpus están cargados en el sistema, el administrador puede dar permisos específicos a cada usuario dependiendo de a qué corpus éste tendrá acceso. Los desarrolladores planean en el futuro permitir que los usuarios puedan subir sus propios documentos, aunque al momento de este escrito esta funcionalidad aún no está implementada.

Como ejemplo de aplicaciones construidas sobre el CWB tenemos el Bwananet de la Universidad Pompeu Fabra (Vivaldi, 2009) y el proyecto Gramateca de la Linguateca (Simões & Santos, 2014).

Manatee + Bonito / SketchEngine

Manatee (Rychlý, 2007) es un software de gestión de corpus de propósito general. Ofrece las mismas funcionalidades que CWB: procesamiento del texto (codificación, etc), administración de metadatos, segmentación del texto, generación de concordancias, anotación (tanto atributos-p como atributos-s) y el cálculo de estadísticas del corpus.

También proporciona un lenguaje de consulta propio, el cual es una extensión de CQP.

El sistema fue diseñado modularmente, proporcionando módulos para la compresión, el indexado y la evaluación de consultas, entre otros.

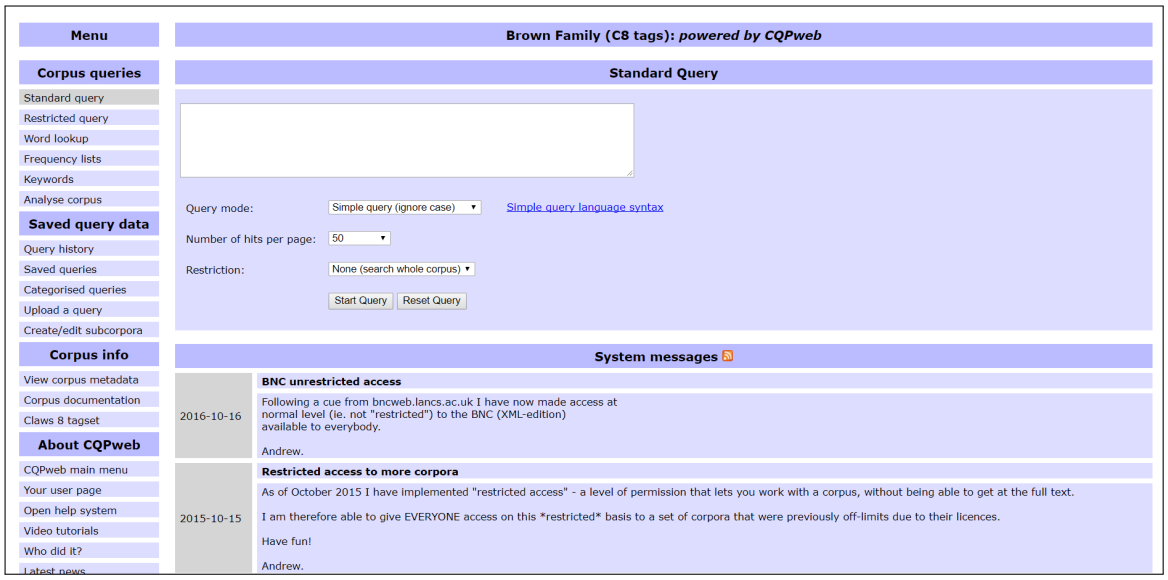

Figura 1: La página de bienvenida del CQPWeb para el Brown Corpus.

Incluye una interfaz de línea de comandos con la cual los corpus son creados y mantenidos. También tiene dos interfaces gráficas: Bonito y Bonito2.

Bonito es un módulo de Manatee que proporciona una interfaz amigable al usuario. Funciona con una arquitectura cliente-servidor, siendo Bonito un cliente de Manatee. Esto significa que Bonito puede ser instalado independientemente de una instancia de Manatee, -incluso en diferentes computadoras- en tanto el primero pueda acceder localmente o vía Internet al segundo. Una versión web de Bonito, llamada Bonito2, también es distribuida junto con Manatee. La mayor diferencia entre los dos es que Bonito2 es accesible vía el navegador web, por lo cual los usuarios pueden evitar instalar el cliente y simplemente dirigir el navegador a la dirección en donde Manatee está funcionando. La Figura 2 muestra un ejemplo de la interfaz web (Rychlý, 2007).

Gracias a su diseño modular, otros sistemas de gestión de corpus pueden ser construidos con base en Manatee + Bonito para agregar funcionalidad adicional. Un ejemplo sería la aplicación comercial SketchEngine. Este software añade nuevas funcionalidades a Manatee + Bonito, ofreciendo bosquejos de palabras (word sketch) además de las herramientas de análisis de corpus tradicionales. A grandes rasgos, los bosquejos de palabras son resúmenes derivados del corpus del comportamiento gramatical y colocacional de una cierta palabra. Se da un ejemplo en la Figura 3.

SketchEngine además tiene el módulo "Arquitecto de COrpus", el cual es una extensión enfocada a la construcción de corpus. Este módulo permite a los usuarios cargar archivos en diversos formatos para crear corpus. Los archivos pueden convertirse en lo que se conoce como un archivo vertical, en el cual cada línea es una palabra junto con sus respectivas anotaciones (por ejemplo, etiquetas POS). Cuando los archivos son cargados, se puede compilar e indexar el corpus vía la interfaz. La funcionalidad del "Arquitecto de COrpus" es además complementada por el módulo "WebBootCaT" (Baroni et al., 2006), el cual está diseñado para construir corpus a partir de la web. Funciona por medio de consultas a la web vía un motor de búsqueda existente (por ejemplo Google) y los documentos que se recuperan se integran en un corpus. También, el usuario puede proporcionar URLs específicos para recuperar sus contenidos.

3 Objetivos de diseño de GECO

A diferencia de los gestores presentados anteriormente, GECO fue concebido en su totalidad como una aplicación web, enfocada en la construcción colaborativa de corpus. En ese sentido, tiene la finalidad de ser un repositorio central de documentos para una variedad de aplicaciones orientadas al PLN, que pueden ser integradas a GECO por desarrolladores con herramientas de código abierto. Además de esto, tiene un enfoque muy marcado hacia la construcción y publicación de corpus como fin en sí mismo, por lo cual también provee herramientas para dar a conocer los corpus creados a través de un portal web personalizado, el cual puede desplegar el nombre de los participantes del proyecto, agradecimientos, referencias, etc.

Figura 2: Generador de concordancias de Bonito2.

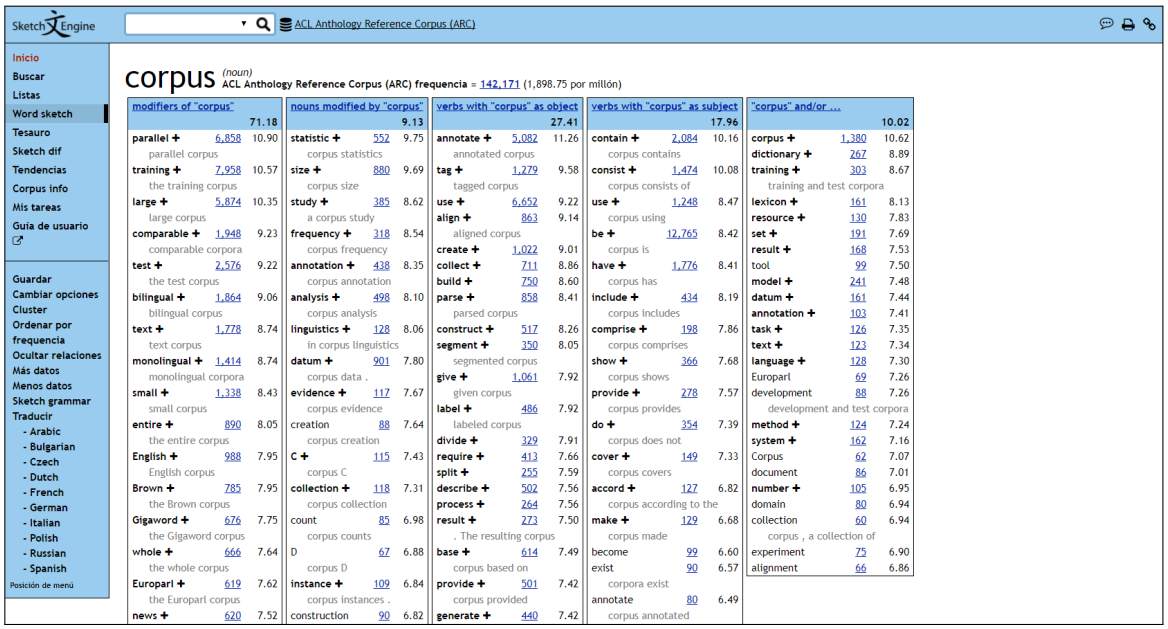

Figura 3: Word Sketch de la palabra corpus, obtenida del ACL Antology Reference COrpus, (Birda et al., 2008), generado con SketchEngine.

Colaboración

Para GECO el proceso de administración de documentos es una tarea colaborativa: varios usuarios pueden participar en la creación de un corpus. Para ello el software organiza archivos similar a como lo haría un sistema de archivos tradicional (Arpaci-Dusseau & Arpaci-Dusseau, 2016, p. 478). Esto permite a los usuarios agrupar documentos en carpetas y compartirlas en línea si así lo desean, dejando que ellos selectivamente decidan quién puede y quién no puede acceder a sus archivos. De manera similar, el administrador del sistema puede controlar el nivel

de acceso que los usuarios tienen de las carpetas. Por ejemplo, solo los usuarios con el permiso de escritura pueden subir documentos a la carpeta y modificar sus metadatos.

Diseño modular

Uno de los aspectos más importantes de GECO es su diseño modular. Fue desarrollado para integrarse con otras aplicaciones de software. Los corpus creados con GECO son visibles para módulos externos vía una Interfaz de Programación de Aplicación (API) (Reddy, 2011). Esto permite a dichas aplicaciones consultar la información acerca de los documentos, recuperando tanto su contenido como sus metadatos. De esta manera, la API facilita la implementación de aplicaciones de PLN, haciendo transparente para los desarrolladores las tareas de preprocesamiento de los textos.

Los módulos de GECO pueden comunicarse unos con los otros, ya que comparten la misma base de datos. Esta comunicación interna permite a los usuarios crear flujos de procesamiento, direccionando los resultados de cualquiera de las aplicaciones existentes para fungir como entrada de otra. Una base de datos compartida también permite a los administradores controlar el acceso individual a cada uno de los módulos, decidiendo quien puede ejecutar qué funcionalidad, dependiendo de las necesidades de cada usuario. Por defecto, los módulos respetan los permisos de archivos: los usuarios pueden ejecutar la funcionalidad de un módulo sólo sobre los datos a los que tienen acceso. De esta manera, los usuarios que tienen una sesión iniciada en GECO pueden visualizar únicamente los módulos y recursos compartidos sobre los cuales tienen permiso, y no los datos privados creados por otros usuarios.

4 Funcionalidades básicas

Desde una perspectiva más técnica, GECO implementa una serie de módulos destinados a cumplir las metas descritas en la sección anterior: un manejador de archivos, un motor PLN de anotación de textos, un sistema de metadatos y varias herramientas de gestión de proyectos. Además, algunos módulos aplicativos basados en la API de GECO son distribuidos junto con el sistema, proporcionando la misma funcionalidad mínima que otros gestores de corpus proporcionan (por ejemplo, generación de concordancias).

La Figura 4 muestra un diagrama de bloques de la organización interna de GECO. De manera breve, El "Núcleo GECO" expone una API web, referido en el diagrama como el "Webservice GECO". Éste se comunica directamente con el módulo "Interfaz Web GECO", construido con web2py (Di Pierro, 2011), el cual funciona como mascarilla del sistema. Las aplicaciones externas comunican la información de sus vistas a la interfaz, mostrado en el diagrama como "Vistas de las Aplicaciones". Los usuarios solo pueden interactuar con el sistema a través de la interfaz web. Sin embargo, la operación y los permisos de acceso son controlados en su totalidad por el núcleo del sistema, a través de la API Web. El "núcleo GECO" se comunica con un servidor de conversión de documentos, el cual transforma documentos en varios formatos a texto plano. También tiene acceso a un servidor FreeLing (Padró & Stanilovsky, 2012), una suite de PLN de código abierto. Finalmente, los datos de configuración del sistema y los archivos del corpus son almacenados en una base de datos relacional y en un sistema de archivos independiente, respectivamente. A continuación se presenta una explicación más a fondo de cómo funcionan estos componentes en conjunto.

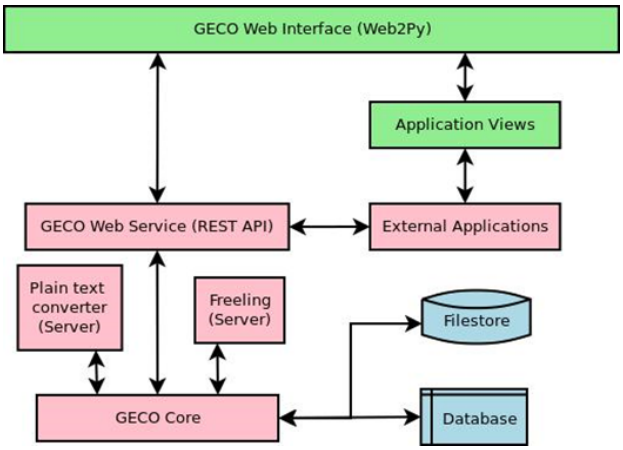

Figura 4: Diagrama de la arquitectura de GECO. Solo la Interfaz web y las Vistas de las Aplicaciones son accesibles a los usuarios. Los documentos son almacenados en un repositorio de archivos central, mientras que la información administrativa del sistema es almacenada en una base de datos relacional.

Manejador de archivos

El "núcleo GECO" es esencialmente un manejador de archivos. Controla la organización de los documentos, incluyendo adición, remoción y modificación. Permite a los usuarios navegar a través de los documentos, organizados en carpetas, como lo harían en cualquier sistema operativo. De esta forma, los usuarios tienen la capacidad de crear nuevas carpetas y llenarlas con

archivos, donde cada carpeta servirá como una posible fuente de documentos para un nuevo proyecto de corpus. La creación de archivos corresponde a una llamada iniciada por el módulo "Interfaz Web GECO".

Como se mencionó anteriormente, los corpus leíbles por máquina están compuestos típicamente de archivos de texto plano, y GECO no es diferente en este aspecto. Sin embargo, a veces los archivos del usuario no son texto plano. El módulo núcleo tiene la responsabilidad de organizar y transformar los diferentes tipos de archivo soportados por GECO como sea necesario. Al recibir un archivo, interpreta y convierte los diferentes tipos de formatos de archivos (por ejemplo, .doc, .pdf, etc) a texto plano UTF-8 (Yergeau, 2003), por medio del módulo "Convertidor de texto plano". Los archivos originales pueden ser conservados por GECO como metadatos multimedia, para no perder información. Para los usuarios, el proceso completo es transparente, ya que no tienen que preocuparse por convertir los archivos antes ellos mismos. La idea detrás de esta conversión es adaptarse a las necesidades del usuario y no viceversa. GECO también puede puede procesar archivos comprimidos en formato ZIP, para cargas masivas.

Cabe mencionar que, por lo menos hasta este momento, GECO no es capaz de importar textos previamente anotados, sea cual fuere su esquema de anotación. Por el momento GECO solo recibe textos sin anotaciones, y es el sistema quien encarga de anotarlo. El archivo anotado resultante es un XML sencillo que puede ser utilizado con otros indexadores de corpus, por ejemplo Manatee. Esta es la razón principal por la cual en esta fase del proyecto no se haya optado por manejar otros esquemas de anotación, como podrían ser los estándares XML-TEI(Areta et al., 2007), aunque bien en el futuro podría beneficiarse de ello.

Preprocesamiento automático del texto

Al insertar documentos de texto plano en el sistema de archivos, el "Núcleo GECO" envía los archivos a un servidor FreeLing, para realizar un procesamiento básico de PLN. Este servicio permite que GECO pueda ejecutar funciones de análisis de lenguaje (segmentación, análisis morfológico, detección de entidades nombradas, etiquetado POS, etc.). El análisis proporcionado por FreeLing permite a GECO segmentar los archivos en tokens, obtener el lema de cada token, y anotarlos con sus respectivas etiquetas POS. Como se verá adelante, este análisis de FreeLing es ampliamente usado en las aplicaciones de GECO.

Por cada archivo, GECO almacena dos versiones: el texto plano UTF-8 (tal cual como se subió), y el texto vertical (con sus respectivas anotaciones tras ser procesado por FreeLing). El sistema permite a los usuarios descargar ambos en cualquier momento. Juntos, el manejo de archivos y el módulo de preprocesamiento actúan como una caja negra, etiquetando el texto plano con información lingüística básica normalmente requerida en los primeros pasos de cualquier tarea de PLN. Adicionalmente, el archivo vertical puede ser enriquecido con metadatos definidos por el usuario. Para esto, el "Núcleo GECO" provee a los usuarios con herramientas para manejar metadatos, como se explica en la siguiente sección.

Manejo de metadatos

Un aspecto importante de la construcción de un corpus es la captura de metadatos. Dependiendo de la aplicación, los usuarios pueden requerir de cualquier número de anotaciones adicionales a nivel documento. En general, el tipo de información a ser añadida depende del uso que se le pretenda dar al corpus y a qué datos puedan ser útiles para obtener conocimiento adicional de esa colección de documentos en específico (Biber et al., 1998). Para este fin, GECO provee funcionalidad para indicar el tipo de campos de anotación que una colección pueda tener. Los campos son definidos como pares (nombre, valor). La herramienta de metadatos funciona por carpeta. Los archivos dentro de la carpeta obtienen el mismo conjunto de campos. Para asignar los valores de cada campo a cada archivo, GECO ofrece una interfaz parecida a una hoja de cálculo. La Figura 5 muestra un ejemplo de edición de metadatos.

En la tabla, cada campo aparece como una columna, con su nombre como encabezado, mientras que cada fila corresponde a un archivo de la colección. El sistema permite a los usuarios agregar o eliminar campos directamente en esta pantalla, es decir, agregar o eliminar columnas. Una vez que los metadatos están capturados, la vista de documentos de GECO permite ordenar y filtrar con base en estos valores. La Figura 6 muestra cómo un campo definido por el usuario puede ser usado para ordenar una colección.

Proyectos

Los documentos que se cargan a una carpeta a GECO no constituyen un corpus automáticamente. Uno de los aspectos flexibles del sistema dentro de un marco de colaboración, es que los usuarios pueden crear un corpus usando archivos

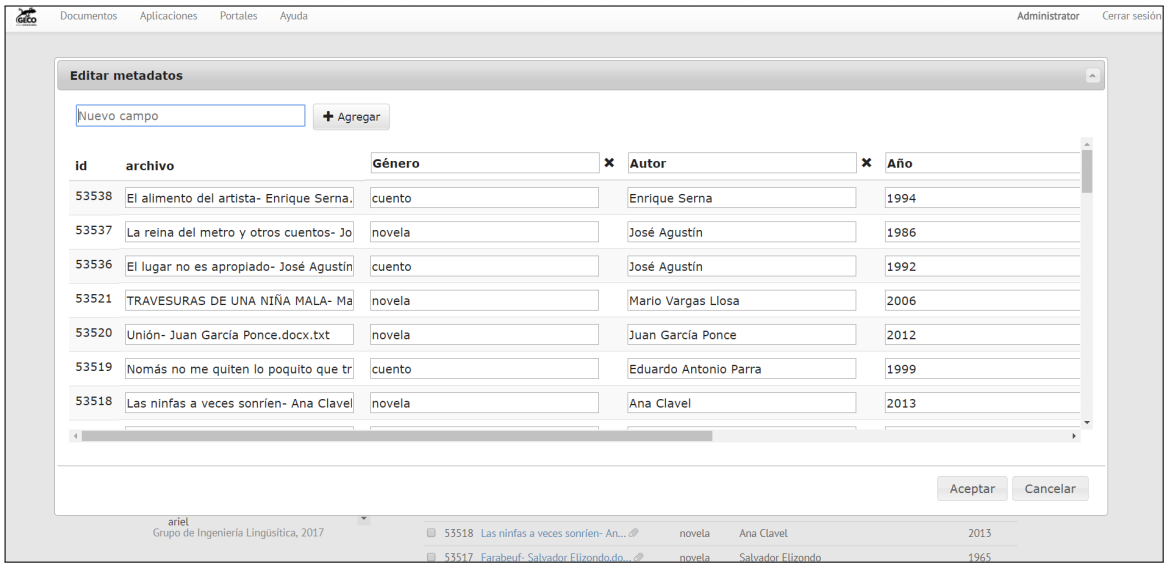

Figura 5: Pantalla de edición de metadatos de GECO.

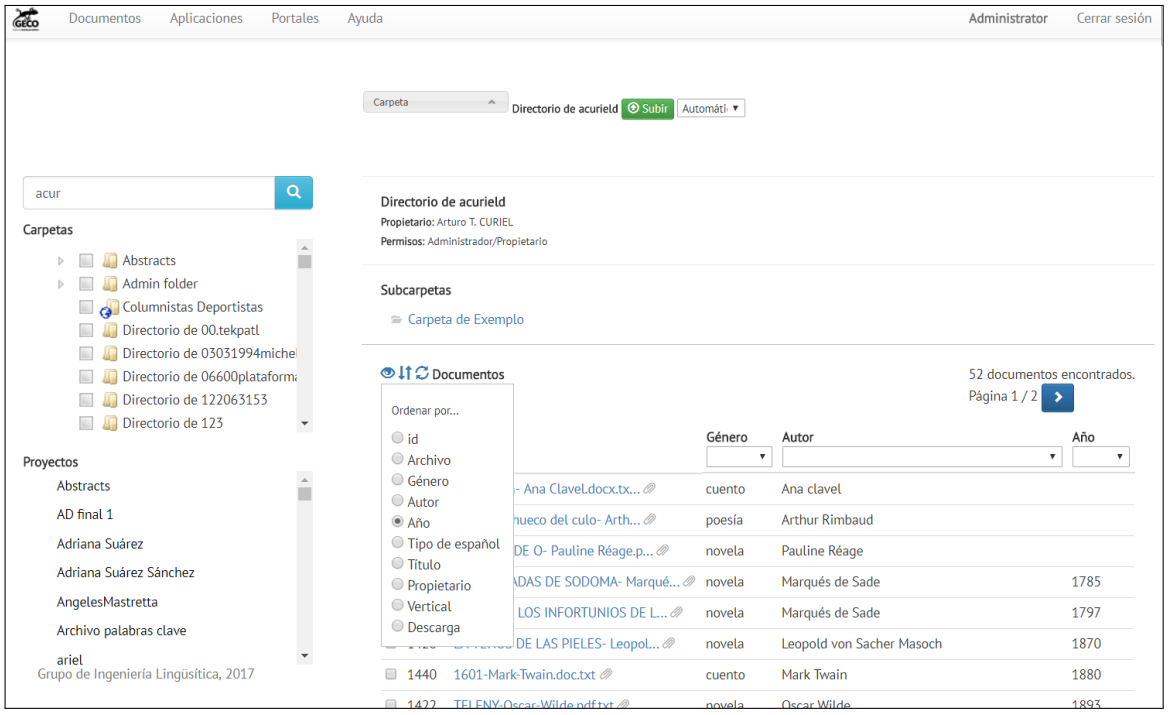

Figura 6: Ordenando una colección por año en GECO.

de varias carpetas. En un caso de uso común, varios usuarios colaborando en la construcción de un solo corpus, pueden tener acceso a diferentes carpetas individuales, donde recolectan sólo un subconjunto del material. No obstante, el corpus requiere ser utilizado como una unidad, aún cuando está contenido en más de una ubicación.

Geco soluciona este problema usando el concepto de "Proyectos", que permite a los usuarios manejar sus corpus como "proyectos". Para el sistema, un proyecto es una colección de documentos, incluso contenidos en diferentes carpetas, unificados bajo un mismo nombre y descripción. Para construir un corpus, el sistema permite a los usuarios escoger archivos individualmente de varias carpetas y encapsularlos en un proyecto, y asignar a un nombre a dicha selección: el nombre del corpus.

Cuando se crea un proyecto, el sistema permite al propietario crear un portal web para el mismo, en el cual los internautas pueden consultar información acerca del corpus, tal como los participantes del proyecto, agradecimientos, publicaciones relacionadas, entre otros. Este portal básico ofrece una estructura genérica para todos los proyectos, que consiste en una serie de pes-

tañas de menú para navegar. Los usuarios pueden elegir la combinación de colores para el portal, la imagen de fondo y el logo que aparecerá en el encabezado de la página. Un ejemplo de la interfaz de configuración del portal se muestra en la Figura 7.

Los proyectos quedan catalogados en el sistema y pueden ser listados mediante el "Web Service GECO". Esto permite a aplicaciones de terceros acceder a colecciones existentes en GECO, siempre y cuando el usuario tenga las credenciales adecuadas para acceder los datos solicitados.

Políticas de seguridad

El manejo de permisos es una de las características más importantes de GECO. Se refiere a qué información los usuarios pueden ver y escribir. Los permisos de lectura y escritura son asignados con base en los tres objetos principales que son: documentos, carpetas y proyectos. Entre estos objetos y cada usuario existirá un "rol", el cual determinará cómo cada usuario puede interactuar con cada objeto.

Los dos principales roles que existen en el sistema son Propietario y Usuario. Los usuarios que tienen el rol de propietario sobre un objeto, pueden otorgar a otros usuarios permisos de acceso sobre ese objeto. Esto es, los Propietarios pueden cambiar las relaciones que existen entre los objetos de los cuales son propietarios, y los demás usuarios. A continuación se describe cómo operan los diferentes permisos más específicamente sobre cada objeto.

Sobre documentos: Los documentos heredan los permisos del contenedor en el que fueron creados: si la carpeta que lo contiene es leíble por un conjunto de usuarios U_f, entonces todo usuario $u \in U_f$ será capaz de leer el documento. Lo mismo pasa con los permisos de un proyecto: si el documento es incluído en el proyecto P, y un grupo de usuarios U_p pueden leer P, entonces todo usuario $u \in U_p$ podrá leer el documento. Lo mismo ocurre con los permisos de creación y escritura. El usuario que crea el archivo (lo carga a la carpeta), obtiene el rol de Propietario. Los propietarios tienen control total sobre sus documentos sin importar la carpeta o proyecto en el que se encuentren. Un documento solo puede ser eliminado por su propietario. Finalmente, un usuario con permisos de lectura sobre el documento lo puede descargar. Es importante que los usuarios verifiquen quién tiene acceso a una carpeta o proyecto antes de incluir un documento en él, para evitar problemas de derechos de autor.

Sobre carpetas: En el caso de las carpetas existe el rol de Colaborador. Adicionalmente, las carpetas tienen dos tipos de nivel de acceso: público y privado. Las carpetas públicas proporcionan a todos los usuarios permiso de lectura sobre las carpetas. Por otro lado, las carpetas privadas solo son visibles a los Propietarios y Colaboradores. Por defecto, toda nueva carpeta es creada con el nivel Privado. El rol de Colaborador proporciona a los usuarios permiso de lectura y permiso de escritura limitado: los colaboradores pueden subir archivos a la carpeta, crear subcarpetas y editar los metadatos de los documentos. Al igual que en el caso de los documentos, solo los Propietarios pueden eliminar la carpeta.

Sobre proyectos: Tienen un comportamiento similar a las carpetas. Como éstas, también pueden ser privados y públicos. Globalmente, los proyectos públicos pueden ser leídos por cualquier usuario, incluyendo usuarios anónimos (no tienen sesión iniciada). Esto significa que cualquier persona puede recuperar los contenidos de un proyecto público, ya sea a través de la interfaz o haciendo una llamada a la API. De manera similar, los proyectos privados solo son visibles para los usuarios con rol Propietario o aquellos que el propietario haya otorgado acceso. Los usuarios con el rol de colaborador pueden agregar o quitar documentos del proyecto (solo quitar del proyecto, no borrarlos completamente). Al igual que en los otros dos tipos de objetos, solo los propietarios pueden eliminar los proyectos. Finalmente, aunque los portales no son un tipo de objeto como tal, son tratados como carpetas en términos de su visibilidad. Hay dos niveles de acceso a los portales: públicos y privados. Los públicos pueden ser visitados por cualquier persona que tenga la URL mientras que las páginas privadas solo pueden ser visitadas por aquellos que tengan permiso explícito, que en el caso de los portales puede ser un usuario en particular o "cualquier usuario con sesión iniciada". Estas políticas pueden ser implantadas a nivel página, de tal manera que la página principal del portal sea accesible a todo el público mientras que otras secciones requieren que el usuario inicie sesión para verlas.

Aplicaciones

El paso final de GECO es el más versátil: implementar aplicaciones sobre el sistema. A grandes rasgos, las aplicaciones son herramientas de software que proveen funcionalidades a los módulos explicados anteriormente. En general, hacen uso de los proyectos GECO expuestos a través de la API, cuyos documentos ya tienen un cierto grado de preprocesamiento, y están listos para ser

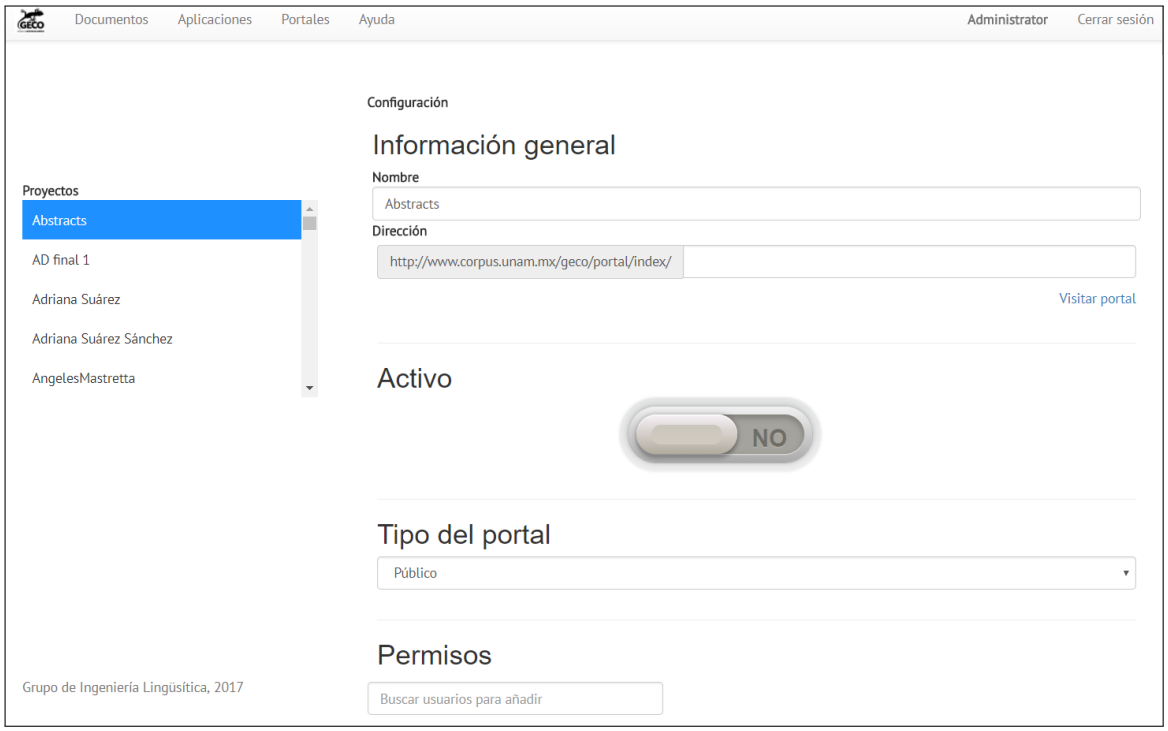

Figura 7: Configuración de portal.

utilizados en otras tareas. Estrictamente hablando, cualquier funcionalidad que estas aplicaciones proveen, no son una funcionalidad de GECO: las aplicaciones son herramientas de software a nivel usuario, diseñadas para aprovechar el contenido de los proyectos GECO.

Como en los proyectos, la API registra y lleva un catálogo de las aplicaciones externas disponibles en GECO y las despliega en la interfaz en el navegador web. La interfaz permite a los usuarios seleccionar cualquier proyecto como entrada a cualquiera de las aplicaciones listadas. Lo contrario también es posible: las aplicaciones pueden ser embebidas en el portal de un proyecto, de tal manera que los internautas pueden hacer uso de las herramientas directamente desde la página del corpus. La Figura 8 muestra un ejemplo de cuatro aplicaciones registradas en GECO, que pueden ser lanzadas con cualquier proyecto de los listados a la izquierda.

Todas las aplicaciones integradas a GECO son expuestas por la API como parte de los recursos disponibles. Esto permite a nuevas aplicaciones llamar programáticamente a las existentes para crear un flujo de trabajo: los nuevos módulos puedes recuperar resultados de otras aplicaciones para usarlos en su propio ciclo de procesamiento. GECO incluye cuatro aplicaciones preregistradas en el catálogo de recursos. Sus funcionalidades se describen brevemente en las siguientes subsecciones.

SAUTEE

El Sistema Automático para Estudios Estilométricos (SAUTEE) es una herramienta que lleva a cabo análisis estilométricos de corpus. Le da al usuario control sobre cuáles marcadores estilométricos utilizar y cómo combinarlos.

A grandes rasgos, los documentos son vectorizados de acuerdo con los marcadores seleccionados y se calcula una distancia entre los vectores resultantes. La salida de esta herramienta es un gráfico de dispersión creado por medio de un escalamiento multidimensional en la matriz de distancias resultante. La imagen producida puede ser usada para inspeccionar cómo los documentos se aglomeran. Para enriquecer el análisis y hacer la visualización más clara al usuario, cada punto del gráfico puede ser coloreado de acuerdo con los metadatos de los documentos. Se presenta un ejemplo en la Figura 9.

TermExt

TermExt es una herramienta de extracción de términos basada en el algoritmo del valor-C (C-value) (Frantzi et al., 2000). La aplicación está diseñada para recibir el contenido de un proyecto registrado en GECO, y extraer de éste una lista de términos, ordenados por puntaje. La Figura 10 muestra un ejemplo de los resultados producidos.

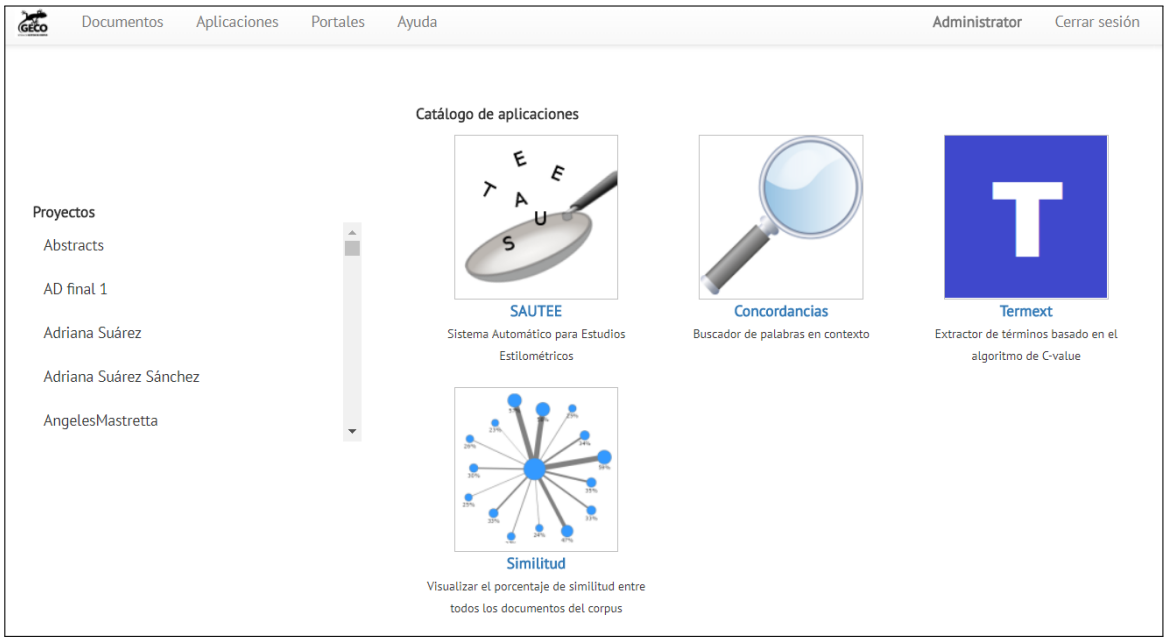

Figura 8: Aplicaciones registradas en GECO, accesibles vía el navegador web.

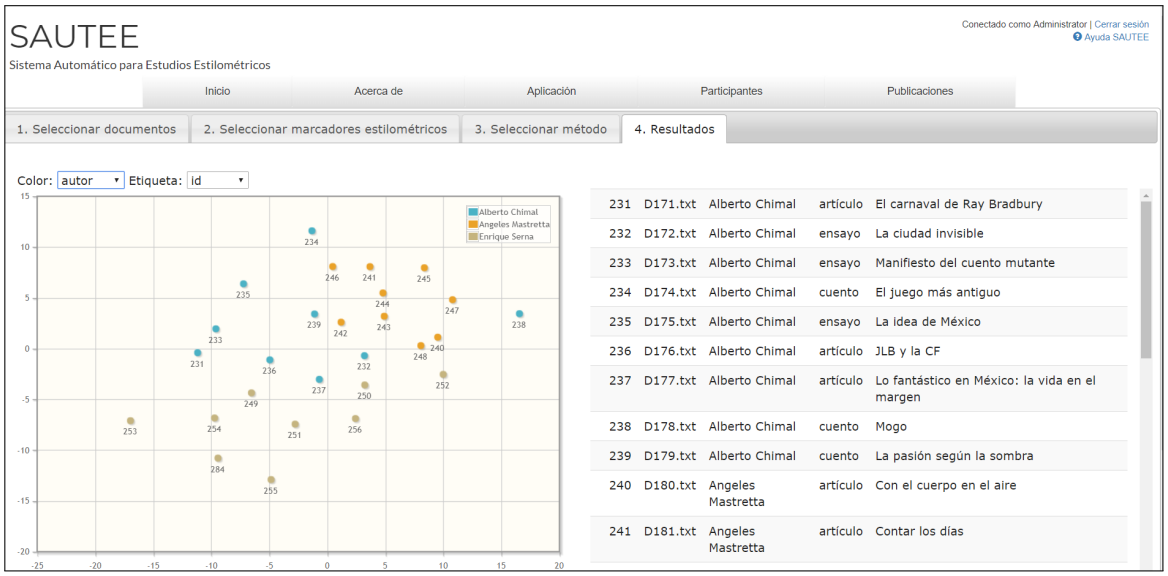

Figura 9: Ejemplo de gráfica generada por SAUTEE.

Similitud

Similitud es una aplicación web que, de manera similar a SAUTEE, calcula una medida de similitud entre todos los documentos de un corpus, solo que la medida está basada en el contenido textual de los documentos, y no en características estilísticas. Su salida es un diagrama que muestra qué tan similar es un documento con respecto a todos los demás. Los valores de similitud son expresados como porcentajes y son desplegados en los nodos de un gráfico de similitud generado por la aplicación. La Figura 11 muestra un ejemplo de los gráficos resultantes.

Concordancias

Concordancias es una aplicación clásica de generación de concordancias que soporta la recuperación tanto de palabras como frases en contexto. Ya que los documentos de GECO están segmentados, lematizados y etiquetados con el etiquetado-POS de Freeling, esta herramienta es capaz de recuperar concordancias en cualquiera de estas tres formas. Adicionalmente, la aplicación puede dividir el corpus en subcorpus más pequeños, filtrando los documentos por sus metadatos. La siguiente sección presenta un análisis más a fondo de está aplicación, así como todos los detalles de su integración con el gestor de corpus.

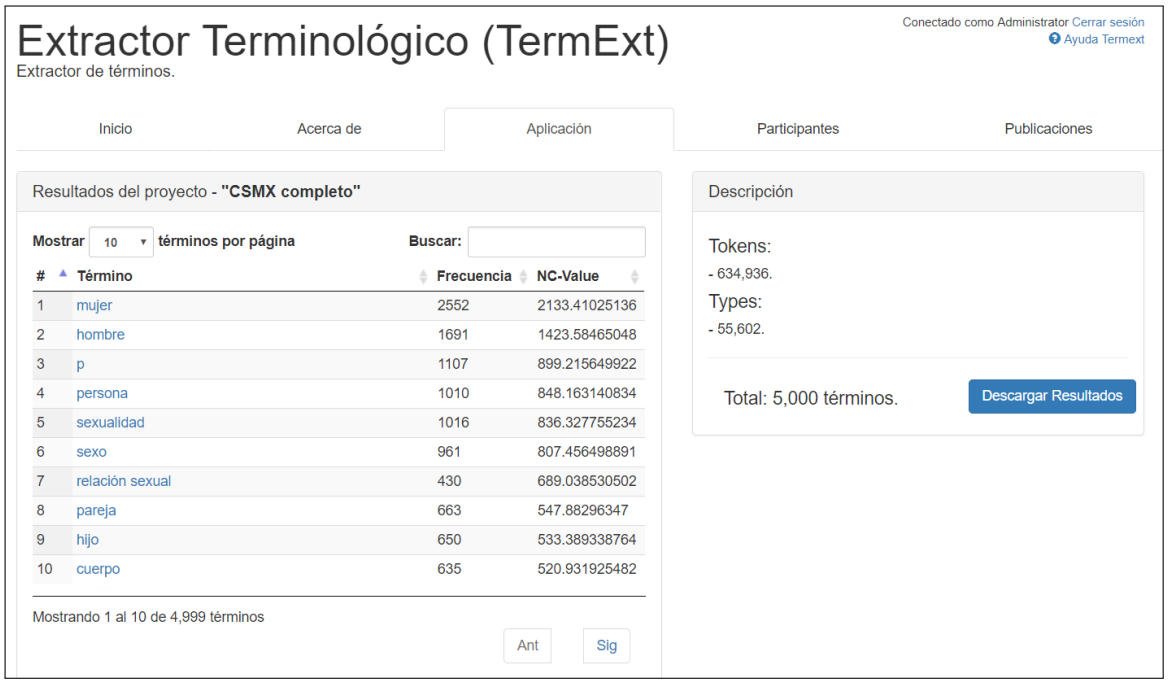

Figura 10: Ejemplo de resultados obtenidos con TermExt.

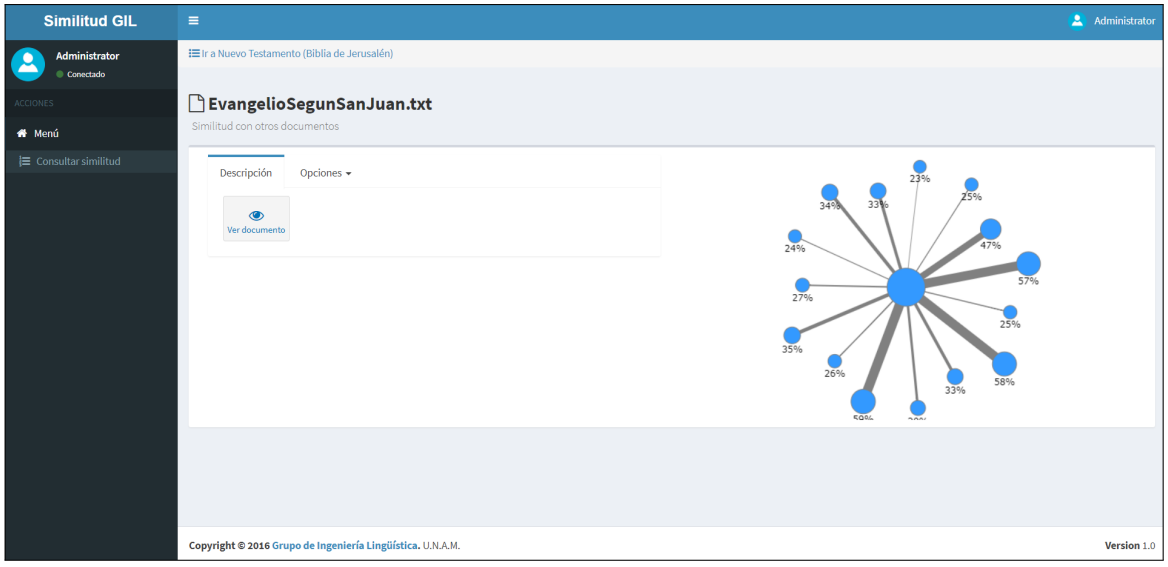

Figura 11: Ejemplo de la gráfica obtenido con Similitud.

5 Aplicación de ejemplo: concordancias

Las concordancias son fragmentos de texto extraídos de un conjunto de documentos, resultado de consultas hechas por un usuario (Manning & Schutze, 1999). Tradicionalmente se muestran en el formato de tres columnas de "Palabra clave en contexto" (Keyword in context o KWIC): la columna de en medio muestra el término buscado (la palabra clave), y las columnas de los lados muestran las palabras que aparecen a la izquierda y a la derecha (el contexto). Los contextos tienen una longitud determinada (tamaño de la ventana), la cual en GECO es un parámetro definido por el usuario. La Figura 12 muestra la pantalla principal de la aplicación.

Selección del corpus

Las concordancias son calculadas a partir de los proyectos registrados en el catálogo de GECO. La aplicación hace una solicitud a la API de GECO, quien retorna una lista de proyectos disponibles y a los que el usuario tenga permiso de accesar según las políticas de seguridad manejadas por GECO, a fin de que pueda seleccionar qué corpus utilizar.

Figura 12: Pantalla de Concordancias.

Por cada proyecto, la aplicación debe hacer un procesamiento propio. Específicamente, se debe crear un índice de los documentos (Zobel et al., 1998). Para esto se recurre a un motor ya mencionado anteriormente en este trabajo: Manatee. Los documentos verticales creados al cargar los documentos a GECO, combinados con los metadatos que hayan sido capturados en los documentos, son enviados como entrada a Manatee para su indexado y compresión. La aplicación de Concordancias guarda este índice localmente, así como la fecha de último indexado, en el caso de que se requiera actualizar el índice si se añadieran nuevos documentos o nuevos metadatos.

Consultas y filtrados

Una vez que los documentos se han indexado, se hace posible ejecutar consultas eficientes. Se propone un lenguaje de consulta especializado que permite buscar por palabra, lema o etiqueta POS. Por ejemplo, uno puede buscar "`perro`" o bien encerrar el término entre corchetes para buscar el lema "`[niño]`" (lo cual traerá resultados para niño, niña, niños, niñas). Se pueden usar diples para buscar una etiqueta POS; por ejemplo, "`<V>`" busca todos los verbos. Los comodines `*` y `?` buscarán cualquier subcadena y cualquier caracter, respectivamente. Estos elementos se pueden combinar; por ejemplo, "`[niño] <V>`" buscará el lema niño seguido de cualquier verbo.

Finalmente, el lenguaje permite búsquedas de proximidad (Goldman et al., 1998), especificando la distancia de una cadena con respecto a otra. Para ello se escribe un número entre llaves, correspondiente a la distancia (en palabras) deseada. Por ejemplo, la consulta "`el niño {3} ayer`" traerá todos los resultados que correspondan a "el niño" seguido entre cero y tres palabras cualquiera más la palabra "ayer".

Además, los metadatos capturados en GECO quedan registrados por manatee como atributos de una etiqueta que envuelve a todo el documento, por lo cual es posible filtrar documentos basados en sus valores. Esto permite efectivamente crear subcorpus en el momento. Para esta clase de filtrado la interfaz presenta selectores de pares campo-valor para restringir el dominio de búsqueda.

Esta sección ha descrito a un alto nivel, cómo es que aplicaciones externas interactúan con los módulos de GECO, mostrando así cómo los principios de diseño de éste promueven la integración. En la siguiente sección se presenta una explicación más técnica del sistema para dar una idea más concreta de su funcionamiento interno.

6 Resumen técnico de la arquitectura de GECO

Las secciones anteriores han presentado el funcionamiento en general de GECO y los principios bajo los que fue diseñado. En esta sección, damos un recorrido más técnico de la funcionalidad del software, dando algunos detalles de implementación de bajo nivel. En particular, los siguientes párrafos describen cómo se guardan los archivos internamente y qué tecnologías son usadas para transportar la información.

Almacenamiento de archivos

Todas las operaciones relacionadas con el almacenamiento de archivos son ejecutadas por Odoo (Reis, 2015), el backend administrativo de GECO y motor del módulo "Núcleo GECO" de la Figura 4.

Odoo es un sistema de gestión empresarial que ofrece varias funcionalidades pre-hechas que permitieron la aceleración del desarrollo. Ejem-

plos de estas funcionalidades por las cuales se incluyó en la arquitectura son: manejo de usuarios (registro y autenticación), visualización de gráficos a partir de cubos de datos (para análisis de estadísticas de uso del sistema), creación de páginas web, y manejo de archivos por medio de una base de datos, lo que se detalla a continuación.

Para insertar nuevos archivos, Odoo asigna un Identificador Único Universal (UUID) a cada archivo cargado, el cual se vuelve su nombre interno. Esto ayuda al sistema a evitar conflictos por nombre, ya que tendrán un nombre único sin importar el nombre del archivo original. Los metadatos de los documentos son almacenados directamente en la base de datos. Con este esquema, solo se leen archivos del sistema de archivos cuando se requieren para procesamiento, mientras que otras operaciones pueden trabajar más rápidamente únicamente manipulando registros de la base de datos.

Base de datos

GECO usa PostgreSQL (Stonebraker & Rowe, 1986), un conocido software de bases de datos relacional. Su función principal es almacenar la información acerca de los usuarios, aplicaciones y permisos de acceso que tienen los objetos en el sistema. También contiene información de los documentos, aunque, como se explicó anteriormente, no los documentos en sí.

Odoo simula un sistema de archivos por medio de los registros de la base de datos. La estructura de las carpetas que se muestra en realidad no es la estructura físicamente en el disco. Odoo construye una representación externa valiéndose únicamente de relaciones entre registros, para evitar operaciones constantes de entrada/salida (lo cual es más lento).

El sistema crea entradas en la base de datos para documentos, carpetas y proyectos, y los relaciona entre sí usando referencias en sus registros. También se guardan referencias que apuntan a la ubicación real del archivo dentro del sistema de archivos.

API

GECO se comunica con aplicaciones externas por medio de una API HTTP mediante la cual provee acceso a todos los recursos y funciones del sistema. La funcionalidad de la API y el web service son provistas por Odoo, el cual recibe peticiones en formato JSON (Bray, 2014). La API permite a aplicaciones externas enviar peticiones JSON al sistema, indicando las operaciones que se desean realizar. Asimismo, permite a las aplicaciones cargar documentos, crear carpetas, integrar proyectos, descargar corpus y modificar metadatos. Está enfocada a los desarrolladores que deseen extender GECO con su propia funcionalidad, en forma de módulos de aplicaciones. La interfaz gráfica de GECO, por ejemplo, está implementada por medio de esta API.

Interfaz gráfica de usuario (GUI)

La GUI de GECO es la manera en que los usuarios finales se comunican con el sistema. Está diseñada para mostrar solo los elementos a los cuales el usuario tiene permiso de accesar. Es accesible a través de cualquier navegador web moderno. La Figura 13 muestra un ejemplo del GUI, en la pantalla de administración de proyectos.

En la figura se puede apreciar que la interfaz muestra una lista de todas las carpetas disponibles en el lado izquierdo de la pantalla. Al hacer click sobre un elemento, se despliegan al usuario sus archivos y subcarpetas. Las carpetas solo se muestran si el usuario tiene permiso de por lo menos lectura. Dentro de las carpetas, los documentos se pueden seleccionar para formar un proyecto o agregarlos a uno existente. Además del control de recursos básico, los menús de navegación de la GUI dan acceso al catálogo de aplicaciones (módulo externos registrados en GECO) y al listado de portales, como se muestra en la Figura 8.

Catálogo de recursos

El catálogo de recursos es donde GECO lista todos los corpus y aplicaciones disponibles, aquellos que pueden ser llamados por medio de la API. Sirve como un índice global por medio del cual las aplicaciones externas que se conecten pueden obtener un listado de todos los corpus y aplicaciones disponibles. Las aplicaciones deben autenticarse para poder acceder a los catálogos, de tal modo que el sistema sabrá qué recursos mostrarle y cuáles no, dependiendo de los permisos de acceso.

Finalmente, el catálogo permite que las aplicaciones se ejecuten de tres modos diferentes:

Normal: Usado cuando el usuario visita el url de la aplicación directamente. En este caso la aplicación debe mostrar un catálogo de corpus para que el usuario pueda seleccionar con cuál proyecto quiere trabajar.

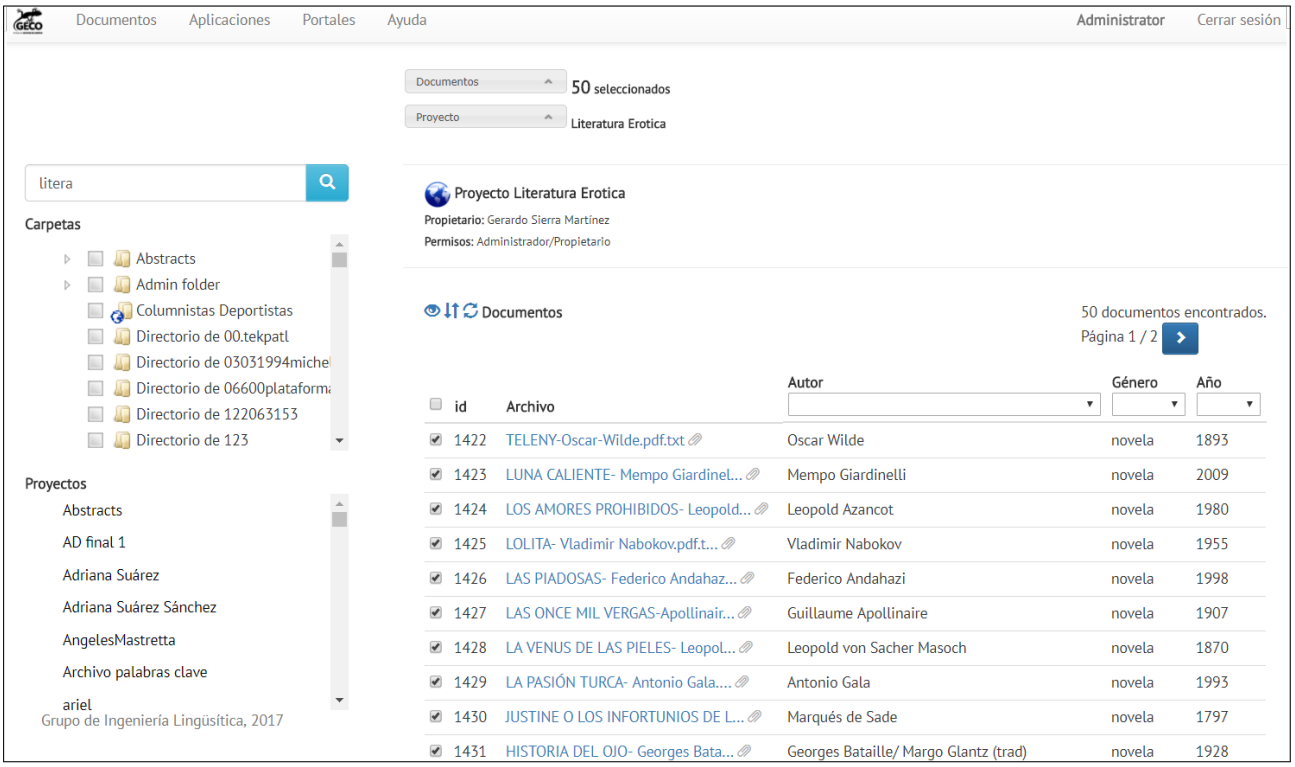

Figura 13: Pantalla de gestión de proyectos en la interfaz de GECO.

Implícito: Usado cuando el usuario lanza una aplicación desde la interfaz de GECO, seleccionando un proyecto de la lista. En este caso la aplicación se abre directamente en el corpus seleccionado.

Embebido: Usado en los portales. Como el implícito, se abre directamente en un corpus preseleccionado. En este caso dicho corpus será aquel que corresponda al proyecto del portal en cual está embebido.

7 Conclusiones y trabajo futuro

En este trabajo se describieron las funcionalidades, principios de diseño y características salientes de un gestor de corpus recientemente desarrollado, GECO, el cual se enfoca en la creación de corpus, y delega la explotación del mismo (por ejemplo, generación de concordancias) por medio de un enfoque modular. También se presentó un módulo para generación de concordancias implementado con Manatee e integrado a GECO por medio de su API.

Las mayores ventajas de GECO sobre otros gestores de corpus es que puede ser usado para construcción de corpus colaborativamente. También tiene la ventaja de que produce corpus anotados, leíbles por máquina, directamente de las fuentes documentales independientemente de su formato, de tal modo que el usuario no tenga que preparar los textos de antemano. Otra funcionalidad que diferencia a GECO de otros softwares existentes es que permite publicar un portal web del corpus, así como embeber aplicaciones como el generador de concordancias en las páginas del portal.

El trabajo futuro incluye agregar soporte para una gran gama de esquemas de anotación. Idealmente GECO debería ser capaz de interpretar cualquier número de atributos-p y atributos-s que pudieran resultar de usar otras herramientas o esquemas de anotación además del análisis básico que actualmente provee Freeling. Esto es un problema abierto en el área, a la vez que propuestas como el modelo de datos Ziggurat (Evert & Hardie, 2015) aún no están estandarizados.

Finalmente, una meta importante para GECO a largo plazo es expandir el catálogo de módulos oficiales. La idea de GECO es serle útil a la comunidad ofreciendo una suite de herramientas de análisis, que de otra manera para un usuario que no sea experto en computadoras le resultarían difíciles de acceder y usar.

Agradecimientos

Este artículo ha sido elaborado gracias a los proyectos PAPIIT IA400117 y Fronteras de la Ciencia 2016-01-2225.

Referencias

Anthony, Laurence. 2005. AntConc: A learner and classroom friendly, multi-platform corpus analysis toolkit. En *An Interactive Workshop on Language e-Learning (IWLeL'2004)*, 7–13.

Areta, Nerea, Antton Gurrutxaga, Igor Leturia, Iñaki Alegria, Xabier Artola, Arantza Diaz de Ilarraza, Nerea Ezeiza & Aitor Sologaistoa. 2007. ZT Corpus: Annotation and tools for Basque corpora. En *Corpus Linguistis Conference*, s.pp.

Arpaci-Dusseau, Remzi & Andrea Arpaci-Dusseau. 2016. Operating systems: Three easy pieces. Electronic Version 0.91.

Baroni, Marco, Adam Kilgarriff, Jan Pomikálek & Pavel Rychlý. 2006. WebBootCat: a web tool for instant corpora. En *EuraLex Conference*, 123–132.

Biber, Douglas, Susan Conrad & Randi Reppen. 1998. *Corpus linguistics: Investigating language structure and use*. Cambridge University Press.

Birda, Steven, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Josepha, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev & Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. En *Language Resources and Evaluation Conference (LREC'2008)*, 1755–1759.

Bray, Tim. 2014. The JavaScript object notation (json) data interchange format. Internet Engineering Task Force. RFC 7159.

Christ, Oliver. 1994. A modular and flexible architecture for an integrated corpus query system. En *3rd Conference on Computational Lexicography and Text Research (COMPLEX'1994)*, 7–10.

Di Pierro, Massimo. 2011. web2py for scientific applications. *Computing in Science & Engineering* 13(2). 64–69.

Evert, Stefan & Andrew Hardie. 2011. Twenty-first century corpus workbench: Updating a query architecture for the new millennium. En *Corpus Linguistics Conference*, s.pp.

Evert, Stefan & Andrew Hardie. 2015. Ziggurat: A new data model and indexing format for large annotated text corpora. En *3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3)*, 21–27.

Francis, W. Nelson. 1965. A standard corpus of edited present-day American English. *College English* 26(4). 267–273.

Frantzi, Katerina, Sophia Ananiadou & Hideki Mima. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries* 3(2). 115–130.

Gamallo, Pablo & Marcos Garcia. 2017. LinguaKit: uma ferramenta multilingue para a análise linguística e a extração de informação. *Linguamática* 9(1). 19–28.

Goldman, Roy, Narayanan Shivakumar, Surech Venkatasubramanian & Hector Garcia-Molina. 1998. Proximity search in databases. En *24rd International Conference on Very Large Data Bases*, 26–37.

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

Kilgarriff, Adam, Fredrik Marcowitz, Simon Smith & James Thomas. 2015. Corpora and language learning with the Sketch Engine and SKELL. *Revue française de linguistique appliquée* XX(1). 61–80.

Kouklakis, George, George Mikros, George Markopoulos & Ilias Koutsis. 2007. Corpus manager: A tool for multilingual corpus analysis. En *Corpus Linguistics Conference*, s.pp.

Manning, Christopher & Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT Press.

Ntoulas, Alexandros, Sofia Stamou, Manolis Tzagarakis, Ioanna Tsakou & Dimitris Christodoulakis. 2001. Viewing web search engines as corpus query systems. En *6th Conference on Computational Lexicography and Corpus Research*, s.pp.

Padró, Lluís & Evgeny Stanilovsky. 2012. FreeLing 3.0: Towards wider multilinguality. En *Language Resources and Evaluation Conference (LREC'2012)*, 2473–2479.

Reddy, Martin. 2011. *API design for C++*. Morgan Kaufmann.

Reis, Daniel. 2015. *Odoo development essentials*. Packt Publishing.

Rychlý, Pavel. 2007. Manatee/bonito: a modular corpus manager. En *1st Workshop on Recent Advances in Slavonic Natural Language Processing*, 65–70.

Sarmento, Luís, Belinda Maia, Diana Santos, Ana Pinto & Luís Cabral. 2006. Corpógrafo V3: From simple word-concordance to semi-automatic knowledge engineering. En *Language Resources and Evaluation Conference (LREC'2006)*, 1502–1505.

Simões, Alberto & Diana Santos. 2014. Nos bastidores da Gramateca: uma série de serviços. En *1st Workshop on Tools and Resources for Automatically Processing Portuguese and Spanish*, 97–104.

Stonebraker, Michael & Lawrence Rowe. 1986. The design of postgres. En *ACM SIGMOD international conference on Management of data*, 340–355.

Vivaldi, Jordi. 2009. Catálogo de herramientas informáticas relacionadas con la creación, gestión y explotación de corpus textuales. *Tradumática* 7. s.pp.

Yergeau, François. 2003. UTF-8, a transformation format of ISO 10646. RFC 3629.

Zobel, Justin, Alistair Moffat & Kotagiri Ramamohanarao. 1998. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems* 23(4). 453–490.