
Mejora de la Comprensión Lectora mediante Analogías para la Inclusión



Trabajo de Fin de Grado
Curso 2017–2018

Autor

Irene Martín Berlanga
Pablo García Hernández

Director

Virginia Francisco Gilmartín
Gonzalo Rubén Mendez Pozo

Grado en Ingeniería de Software
Facultad de Informática
Universidad Complutense de Madrid

Mejora de la Comprensión Lectora mediante Analogías para la Inclusión

**Trabajo de Fin de Grado en Ingeniería de Software
Departamento de Ingeniería de Software e Inteligencia
Artificial**

Autor

**Irene Martín Berlanga
Pablo García Hernández**

Director

**Virginia Francisco Gilmartín
Gonzalo Rubén Mendez Pozo**

Dirigida por el Doctor

**Virginia Francisco Gilmartín
Gonzalo Rubén Mendez Pozo**

**Grado en Ingeniería de Software
Facultad de Informática
Universidad Complutense de Madrid**

16 de noviembre de 2018

Autorización de difusión

Los abajo firmantes, matriculados en el Grado de Ingeniería de Software de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el presente Trabajo de Fin de Grado: “Mejora de la Comprensión Lectora mediante Analogías para la Inclusión”, realizado durante el curso académico 2018-2019 bajo la dirección de Virginia Francisco Gilmartín y Gonzalo Rubén Mendez Pozo en el Departamento de Ingeniería de Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Nombre Del Alumno

16 de noviembre de 2018

Dedicatoria

Texto de la dedicatoria...

Agradecimientos

Texto de los agradecimientos

Resumen

Resumen en español del trabajo

Palabras clave

Máximo 10 palabras clave separadas por comas

Abstract

Abstract in English.

Keywords

10 keywords max., separated by commas.

Índice

1. Introduction	1
1. Introducción	3
1.1. Motivación	3
1.2. Objetivos	4
2. Estado de la Cuestión	7
2.1. Lectura Fácil	7
2.2. Procesamiento del lenguaje natural	9
2.2.1. ConceptNet	9
2.3. Figuras retóricas	9
2.4. Servicios Web	10
2.4.1. Arquitectura Servicios Web	11
2.4.2. Ventajas de los Servicios Web	12
2.4.3. Desventajas de los Servicios Web	13
3. Trabajo Realizado	15
3.1. Trabajo realizado por Irene	15
3.2. Trabajo realizado por Pablo	15
4. Conclusiones y Trabajo Futuro	17
4. Conclusions and Future Work	19
A. Título	21
B. Título	23

Índice de figuras

Índice de tablas

Chapter 1

Introduction

Introduction to the subject area.

Capítulo 1

Introducción

“Frase célebre dicha por alguien inteligente”

— Autor

RESUMEN: En el capítulo 1 vamos a realizar una introducción a nuestro TFG. Dicha introducción la dividiremos en varias secciones. En el apartado 1.1 hablaremos de la razón de ser nuestro trabajo y en el 1.2 de los objetivos que queremos alcanzar con el mismo

1.1. Motivación

En nuestra sociedad actual, existen ciertos colectivos como pueden ser personas con algún tipo de trastorno cognitivo, inmigrantes, ancianos, analfabetos funcionales, niños, etc... que tienen dificultad para aprender conceptos complejos o no tan complejos. Para ellos, esto supone una serie de limitaciones importantes en su vida cotidiana, como en la forma de relacionarse con otros individuos, en su vida profesional e incluso su vida personal. Por ejemplo en el colectivo de los inmigrantes que tienen que realizar gestiones para la estancia en nuestro país, si no tienen una buena comprensión de nuestro lenguaje pueden ser engañados fácilmente y no pueden defender sus derechos de una forma adecuada. Otro ejemplo de la magnitud de las consecuencias de este problema es lo que ocurrió hace un tiempo con las preferentes que muchas entidades financieras engañaron a personas que pertenecen a este tipo de colectivos (sobre todo personas mayores) para que invirtieran en las preferentes cuando realmente era una estafa.

Para ayudar principalmente a estas personas a que puedan entender el significado de cualquier palabra, y de esta forma superar algunas de sus limitaciones, vamos a desarrollar una aplicación que permita definir palabras complejas mediante comparaciones con otras más fáciles ya conocidas por los usuarios.

1.2. Objetivos

Nuestro objetivo final es crear una aplicación que dada una palabra compleja, genere símiles, analogías o metáforas que expliquen dicha palabra utilizando conceptos más simples, preferiblemente que estén entre las 1000 palabras más utilizadas de la RAE. La aplicación se apoyará en varios servicios web donde el usuario podrá buscar cualquier palabra, y obtendrá una descripción lo más sencilla posible. Por ejemplo, si buscamos la palabra *piraña*, la aplicación devolverá la siguiente descripción: *Nada como un pez, es pequeño como un ratón y es agresivo como un león*. De esta manera, cualquier persona que no sepa lo que es una *piraña*, puede hacerse una idea de como es dicho animal y asimilar el nuevo concepto. Además, utilizaremos técnicas centradas en el usuario para diseñar una interfaz lo más usable posible y que así el usuario tenga una experiencia de uso satisfactoria.

Por otra parte, queremos que nuestro producto se encuentre al alcance de todas aquellas personas que lo necesiten, que puedan sentir que es realmente útil porque obtienen resultados satisfactorios. Para que todo ello ocurra, los objetivos principales a alcanzar son:

- Desarrollar una aplicación que esté al alcance de todos los usuarios.
- Aplicaciones del proyecto: Estará enfocado en ser utilizado por personas con algún tipo de dificultad de aprendizaje para la mejora de la comprensión de palabras complejas
- Desarrollar servicios web que puedan ser utilizados por cualquier programador
- Construir la aplicación de manera incremental, añadiéndole valor poco a poco
- Construir una aplicación que esté entrada en el usuario, que solucione sus problemas de manera usable
- Realizar un producto que sea eficiente
- Que la aplicación cumpla con unos estándares de calidad
- Un diseño de Interfaz que cumpla con las Ocho Reglas de Oro del diseño de interfaces:
 - Consistencia: La funcionalidad de la interfaz sea similar a otras aplicaciones que el usuario utiliza normalmente.
 - Usabilidad Universal: Cualquier usuario sea capaz de utilizar nuestra aplicación.
 - Retroalimentación activa: Informar al usuario de cada acción que realiza.

- Diálogos para conducir la finalización: El usuario sepa en que etapa está cuando quiere realizar una acción que requiera varios pasos.
- Prevención de errores: Diseñar la interfaz para que el usuario cometa el mínimos de errores posibles.
- Deshacer acciones fácilmente: Cualquier acción pueda ser deshecha
- Sensación de control: Dar al usuario la sensación de que tiene el control en todo momento de la aplicación.
- Reducir la carga de memoria a corto plazo: Minimizar la cantidad de elementos que tiene que memorizar el usuario.

Y no nos tenemos que olvidar de los objetivos académicos. Poner en práctica los conocimientos adquiridos durante el Grado, y ampliar nuestros conocimientos gracias a la utilización de herramientas, lenguajes y metodologías nuevas.

Capítulo 2

Estado de la Cuestión

RESUMEN: En este capítulo se van a tratar aspectos importantes dentro del ámbito de la retórica así como una explicación detallada de Lectura Fácil, sin olvidar aquellas herramientas y tecnologías que se van a utilizar. En la sección 2.1 se explica el concepto de Servicio Web, su funcionalidad, ventajas y desventajas y tipos que existentes. En la sección 2.2 se definirán las Figuras retóricas.

2.1. Lectura Fácil

La lectura fácil surgió en Suecia. En 1968 se editó el primer libro en la Agencia de Educación en el marco de un proyecto experimental. A continuación, en 1976 se creó en el Ministerio de Justicia un grupo de trabajo para conseguir textos legales más claros.

El Ministerio de Educación comenzó en 1984 la edición de un libro llamado "8 Sidor", que tres años más tarde, en 1987, se publica de forma permanente en papel hasta que empezó a editarse en la web. Este mismo año la Fundación Lectura Fácil asume la publicación del semanario y de los libros escritos en lectura fácil.

¹

Los documentos escritos en Lectura Fácil son documentos de todo tipo que siguen las directrices internacionales de la IFLA (International Federation of Library Associations and Institutions) y de Inclusion Europe en cuanto al contenido y la forma.

Está dirigida a colectivos que tengan dificultades lectoras como inmigrantes o personas con trastornos de aprendizaje o problemas cognitivos. Algunos ejemplos de como se debe de escribir un texto en Lectura Fácil son:

¹<https://dilo Facil.wordpress.com/2013/12/04/el-origen-de-la-lectura-facil/>

- Evitar mayúsculas fuera de la norma, es decir, escribir en mayúsculas sólo cuando toca según las reglas ortográficas
- Limitar el uso de la coma y evitar signos menos usados de puntuación
- Evitar tiempos verbales complejos
- Utilizar oraciones simples
- Utilizar palabras cortas y de sílabas poco complejas
- Incluir imágenes y pictogramas a la izquierda y su texto vinculado a la derecha
- Utilizar uno o dos tipos de letra como mucho
- Tamaño de letra entre 12 y 16 puntos
- Evitar frases superiores a 60 caracteres
- Transmitir una idea por línea y cada línea se rompe en un punto natural del discurso
- Alineación a la izquierda
- Evitar la saturación de texto e imágenes
- Incluir el logo de lectura fácil para que se reconozca este tipo de obras.
- Si el documento está paginado, incluir la paginación claramente y reforzar el mensaje de que la información continúa en la página siguiente. (*Lectura fácil: Métodos de redacción y evaluación. Óscar García Muñoz.*)

A continuación vamos a hablar del lenguaje natural, y de que relación tiene con la lectura fácil:

El lenguaje natural es la forma en la que los humanos nos comunicamos día a día ya sea de forma escrita u oral ²

Actualmente, el Procesamiento del lenguaje natural, que es un campo de la Inteligencia Artificial que investiga la manera de comunicar a las máquinas con el ser humano, se está aplicando a la lectura fácil. Por ejemplo, el proyecto "Simplext" desarrollado por una empresa de la Fundación ONCE que tiene como objetivo conseguir un adaptador automático de texto normal a texto en lectural fácil ^{3 4 5}

²<https://sistemas.com/lenguaje-natural.php/>

³<http://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>

⁴<https://dilofacil.wordpress.com/2013/11/07/procesamiento-del-lenguaje-natural-el-futuro-para-la-lectura-facil/>

⁵<https://www.fundaciononce.es/es/noticia/el-proyecto-simplext-de-technosite-finalista-de-los-premios-bdigital-la-innovacion/>

2.2. Procesamiento del lenguaje natural

2.2.1. ConceptNet

Es una red semántica diseñada para ayudar a los ordenadores a entender los significados de las palabras utilizadas por la gente lanzada por el MIT en 1999. Tiene un buscador de palabras en el que al introducir una palabra, se selecciona el idioma y devuelve sinónimos y términos relacionados.⁶

Por ejemplo, si introducimos la palabra "dinero", especificando que el lenguaje en el que está dicha palabra, en ese caso español, ConceptNet te devuelve como sinónimos:

- lana
- pasta
- billete
- plata

y como términos relacionados:

- dineral
- moneda

Entre otros, también te devuelve tanto sinónimos como términos relacionados en otros idiomas, pero nosotros solo nos centraremos en los que estén en castellano.

Dispone de una API (<http://api.conceptnet.io/>) que devuelve los datos en formato JSON, que utilizaremos para obtener los términos relacionados y sinónimos que necesitamos.

2.3. Figuras retóricas

Las figuras literarias(o retóricas) son formas no convencionales de utilizar las palabras, de manera que, aunque se emplean con sus acepciones habituales, se acompañan de algunas particularidades fónicas, gramaticales o semánticas, que las alejan de ese uso habitual, por lo que terminan por resultar especialmente expresivas. (*Introducción al análisis retórico: tropos, figuras y sintaxis del estilo*)⁷

En este trabajo vamos a trabajar con tres tipos de figuras retóricas: Analogía, metáfora y símil.

⁶<http://conceptnet.io/>

⁷Azaustre Galiana, Antonio y Juan Casas Rigall, *Introducción al análisis retórico: tropos, figuras y sintaxis del estilo*, Universidad de Santiago de Compostela, Santiago de Compostela, 1994.

- Una metáfora: Su estudio se remonta a Aristóteles y su Retórica. Una metáfora se refiere a una cosa mencionando otra, utiliza el desplazamiento de características similares entre dos conceptos con fines estéticos o retóricos. Por ejemplo, cuando una persona tiene muy buena memoria, se la dice que tiene memoria de elefante.
- Un símil: Realiza una comparación entre términos. A pesar de que los símiles y las metáforas son similares, los símiles utilizan explícitamente, aunque no necesariamente, conectores (por ejemplo, como, cual, que, o varios verbos tales como se asemejan). Por ejemplo: Cuando nos referimos a una persona que es muy corpulenta, se dice que es como un oso.
- Una analogía: Es la comparación entre varios conceptos, indicando las características que permiten dicha relación. En la retórica, una analogía es una comparación textual que resalta alguna de las similitudes semánticas entre los conceptos protagonistas en dicha comparación. Por ejemplo, sus ojos son azules como el mar.

(⁸)

- Analogía: Según la RAE, es la relación de semejanza entre dos cosas distintas.
- Símil: Según la RAE, es la comparación, semejanza entre dos cosas.
- Metáfora: Una metáfora es una Figura retórica de pensamiento por medio de la cual una realidad o concepto se expresan por medio de una realidad o concepto diferentes con los que lo representado guarda cierta relación de semejanza.

2.4. Servicios Web

Para definir el concepto de servicio Web de la forma más simple posible, se podría decir que es una tecnología que utiliza un conjunto de protocolos para intercambiar datos entre aplicaciones, sin importar el lenguaje de programación en el cual estén programadas o ejecutadas en cualquier tipo de plataforma. Según el W3C(*World Wide Web Consortium*)⁹, un servicio web es un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable.

Las principales características de un servicio web son:

⁸GENERACIÓN DE RECURSOS LINGÜÍSTICOS MEDIANTE LA EXTRACCIÓN DE RELACIONES ENTRE CONCEPTOS

⁹<https://www.w3.org/>

- Debe poder ser accesible a través de la Web. Para ello debe utilizar protocolos de transporte estándares como HTTP, y codificar los mensajes en un lenguaje estándar que pueda ser accesible por cualquier cliente que quiera utilizar el servicio.
- Debe contener una descripción de sí mismo. De esta forma, una aplicación web podrá saber cual es la función de un determinado Servicio Web, y cuál es su interfaz, de manera que pueda ser utilizado de forma automática por cualquier aplicación, sin la intervención del usuario.
- Debe poder ser localizado. Debemos tener algún mecanismo que nos permita encontrar un Servicio Web que realice una determinada función. De esta forma tendremos la posibilidad de que una aplicación localice el servicio que necesite de forma automática, sin tener que conocerlo previamente el usuario.

Los servicios web pueden definirse tanto a nivel conceptual como a nivel técnico, es por ello que mediante este último podemos diferenciar dos tipos distintos de servicio web:

- Servicios Web RESTful: no tienen estado. Cada petición del cliente al servidor debe contener toda la información necesaria para entender la petición, y no puede aprovecharse de ningún contexto almacenado en el servidor.
- Servicios web SOAP (Simple Object Access Protocol): es un protocolo basado en XML para el intercambio de información entre ordenadores. Normalmente utilizaremos SOAP para conectarnos a un servicio e invocar métodos remotos.

2.4.1. Arquitectura Servicios Web

Los servicios web se componen fundamentalmente de tres partes:

- Proveedor: Es la aplicación que implementa el servicio y lo hace accesible mediante Internet
- Solicitante: Cualquier persona que necesite utilizar el servicio web
- Publicador: Se refiere al repositorio centralizado en el que se encuentra la información de la funcionalidad disponible y como se utiliza

A continuación, explicamos más detalladamente las distintas capas que tiene el servicio web:

- Service Discovery. Es el responsable de centralizar los servicios web en un directorio común de esta forma es mas sencillo buscar y publicar.

- **Service Description.** Como ya hemos comentado con anterioridad, los servicios web se pueden definir así mismos, por lo que una vez que los localicemos nos darán suficiente información para saber que operaciones soporta y como activarlo.
- **Service Invocation.** Invocar a un Web Service implica pasar mensajes entre el cliente y el servidor. SOAP (Simple Object Access Protocol) especifica cómo deberíamos formatear los mensajes request para el servidor, y cómo el servidor debería formatear sus mensajes de respuesta.
- **Transport.** Todos estos mensajes han de ser transmitidos de alguna forma entre el servidor y el cliente. El protocolo elegido para ello es HTTP ((HyperText Transfer Protocol)).

2.4.2. Ventajas de los Servicios Web

Estos tienen multitud de ventajas pero las principales son las siguientes:

- **Permiten la integración “justo-a-tiempo”:** Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son auto-configurables, adaptativos y robustos.
- **Reducen la complejidad por medio del encapsulamiento:** Los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementado el servicio por parte del proveedor, y éste a su vez, no sabe cómo utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores. El encapsulamiento es crucial para reducir la complejidad.
- **Promueven la interoperabilidad:** La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje.
- **Abren la puerta a nuevas oportunidades de negocio:** Los servicios web facilitan la interacción con socios de negocios, al poder compartir servicios internos con un alto grado de integración.
- **Disminuyen el tiempo de desarrollo de las aplicaciones:** Pues gracias a la filosofía de orientación a objetos que utilizan, el desarrollo se convierte más bien en una labor de composición.
- **Fomentan los estándares y protocolos basados en texto,** que hacen más fácil acceder a su contenido y entender su funcionamiento.

2.4.3. Desventajas de los Servicios Web

Pero también tiene desventajas, las cuales tenemos que nombrar:

- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear.
- Existe poca información de servicios web para algunos lenguajes de programación .
- Dependencia de la disponibilidad de servidores y comunicaciones.

Capítulo 3

Trabajo Realizado

RESUMEN: En este capítulo vamos a describir que trabajo hemos hecho cada uno

3.1. Trabajo realizado por Irene

Primero investigamos las bibliotecas que utilizaremos para el procesado de las palabras, al principio encontramos una biblioteca para el procesado de texto en Python, que es la `ntlk` pero vimos que las etiquetas que ponía a las palabras no eran del todo correctas por lo que buscamos otra biblioteca y encontramos `Spacy`, con esta ya pudimos etiquetar bien todas las palabras diseñando un programa inicialmente en el `Jupyter`. A continuación, investigamos que tecnologías utilizar para la realización del prototipo(un servicio web) y encontramos como entorno de desarrollo el `Pycharm` y como framework el `Django`. Una vez seleccionadas las tecnologías, investigamos como se utilizaban y nos pusimos a trabajar en el prototipo.

Yo me encargué de conectar las vistas html con la lógica en Python, a continuación vimos como se implementaba un formulario y como se hacia una redirección a vista. Cuando supimos como se hacia todo esto, integramos el código desarrollado en `Jupyter` a nuestro servicio web finalizando el prototipo.

3.2. Trabajo realizado por Pablo

Al igual que mi compañera, lo primero que hicimos fue investigar como podíamos etiquetar las palabras, encontramos la librería `ntlk` de Python para hacerlo, pero tras un primer intento nos dimos cuenta de que muchas palabras no estaban etiquetadas como deberían por lo que decidimos buscar

alternativas, indagando un poco encontramos Spacy, la probamos y obtuvimos unos resultados mucho mejores que con nltk por lo que decidimos utilizar esta última (todo esto lo hicimos desde el Jupyter).

Cuando terminamos de etiquetar las palabras nos pusimos a investigar herramientas para el desarrollo del servicio web prototipo y nos decantamos por utilizar Django como framework integrado en Pycharm, que es el entorno de desarrollo.

A continuación empezamos el desarrollo del servicio web primero investigando como se utilizaban estas tecnologías(implementar formularios, hacer las redirecciones a vista...). Para finalizar migramos lo hecho desde el Jupyter a nuestro servicio web.

Capítulo 4

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Chapter 4

Conclusions and Future Work

Conclusions and future lines of work.

Apéndice	A
----------	----------

Título

Contenido del apéndice

Apéndice	B
----------	----------

Título

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

