Prototipo 1

Descripción

En inicio de se trató de utilizar herramientas abiertas disponibles en la web para el proceso de enlace y desambiguación de los elementos trascedentes en un texto, así se consumió el servicio ofrecido por DBpedia, spotlight el cual produce el resultado esperado pero con problemas de conexión debido a que el servidor no se encontraba siempre disponible, para lo cual se construyó un cliente REST en python. A partir de este intento y conforme se fue avanzando en la investigación de soluciones para los requerimientos presentados, se decidió desarrollar un propuesta de software propia.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Desambiguación y enlace | * Consumir servicios libres existentes para procesos de * Tiempo excesivos de estará para espera respuesta | * DBpedia Spotlight | 3 días |
| Construcción de cliente para servicios | * Conexión con servidores REST | * Python | 2 días |

Prototipo 2

Descripción

A través de Procesamiento de Lenguaje Natural (PLN), se trabaja con el texto de entrada para obtener las entidades de quien se habla en las oraciones, así como las palabras claves que los acompañan. Este procesamiento se lo realiza especializado en el idioma ingles utilizando la librería Natural Language Toolkit (NLTK) de Python.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Tokenización en sentencias | Procesamiento de lenguaje natural o PLN | NLTK (Natural Language Toolkit) librería de python | 2 semanas |
| Tiokenizacion en palabras | PLN | NLTK | 2 horas |
| Etiquetado (Part of speech) en idioma ingles | PLN | NLTK | 2 horas |
| Extracción de entidades y palabras claves idioma ingles | PLN | NLTK | 1 día |

Prototipo 3

Descripción

Todos lo esfuerzos realizados se concentran en el idioma inglés, a través de NLTK para su procesamiento, para en otros idiomas como español por ejemplo no se puede realizar el mismo proceso que en el idioma ingles debido a lo diferente de su estructura. Para poder trabajar en otros idiomas se debe partir de un etiquetado (Part of Speech) propio del idioma, por lo cual se decidió trabajar con TreeTagger, que permite etiquetado en diferentes idiomas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Etiquetado (Part of speech) en idioma ingles | PLN | TreeTagger | 5 días |
| Extracción de entidades y palabras claves idioma ingles | PLN | NLTK | 2 día |

Prototipo 4

Descripción

Una vez que se ha logrado extraer los datos importantes de un texto, procederemos a enlazar esto con recursos disponibles en BDpedia.org, para esto se realizan consultas Sparql para obtener los recursos que coinciden con los datos extraídos. Una vez obtenidos los recurso de DBpedia, pueden existir más de una recurso que pueda coincidir para un dato del texto, este caso decimos que este término es ambiguo y es necesario romper esta ambigüedad para enlazarlo con un solo recurso de DBpedia, para lo cual utilizamos el algoritmo de Lesk introducido por  Michael E. Lesk en 1986.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Enlazar entidades y palabras claves encontrados con recursos similares en DBpedia | Sparql | Python  Sparql | 4 días |
| Desambiguación entre los recursos encontrados | Algoritmo de desambiguación de Lesk | Python  Sparql | 5 días |

Prototipo 5

Descripción

Se levantan servicio Rest para ofrecer la propuesta y una interfaz para que pueda interactuar con usuarios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Servicio Web REST | Levantar servicios Web Rest | REST  Python | 2 semanas |
| Interfaz de usuario (Web) prototipo inicial | Construcción de cliente Web JavaScript | JavaScript  HTML | 3 días |

Prototipo 6

Descripción

Se levantan servicios diferenciados por los procesos realizado de esta forma el resultado de un servicio o es la entrada de toro servicio y así puedes ser consumido por otros servicios externos a esta propuesta.

Para poder visualizar el resultado de la interacción de los servicios Web de forma agradable para usuarios se mejora la interfaz gráfica en donde se diferencia el resultado de cada servicio.

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad especifica** | **Aprendizaje** | **Herramienta** | **Tiempo** |
| Servicios web diferenciados por Procesos | Levantar servicios Web Rest | REST  Python | 2 semanas |
| Interfaz de usuario final | Mejoramiento de interfaz Web | Css  JavaScript | 7 días |