Sistema Simple de Pregunta y Respuesta en Español

Cristóbal Donoso Oliva

Ciencias de la Computación, Universidad de Concepción

Objetivos

Los objetivos de este proyecto son

- Extraer información desde la web del tipo *Quien* y *Donde*
- Entregar respuestas coherentes a consultas realizadas en lenguaje natural
- Jerarquizar calidad de respuestas

Introducción

Internet es un gran banco de información la cual podemos capturar realizando lectura de cada uno de los documentos que allí se alojan. Estos documentos son páginas HTML cuyo contenido (texto plano) está delimitado por marcadores sintácticos propios del lenguaje.**Dada una consulta, es necesario utilizar métodos de recuperación y extracción de información con el objetivo de contestar cada pregunta**. A continuación, se muestra el desarrollo de un sistema simple de pregunta y respuesta para documentos virtuales en español. Las preguntas serán del tipo *Quién* y *Dónde*. Finalmente, se mostrarán los resultados obtenidos.

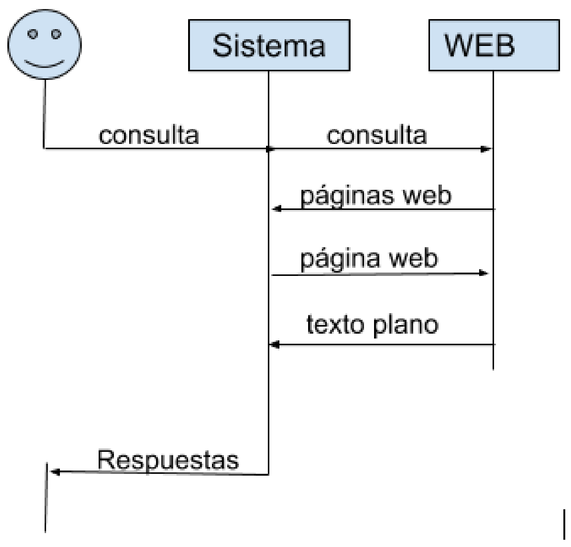

Figure 1:Diagrama de comunicación entre el usuario, el sistema y la web

Ingresando la Pregunta

El sistema cuenta con 7 partes:

1. **Ingreso de la pregunta en lenguaje natural:** El usuario interactúa con el sistema escribiendo una consulta en español.
2. **Limpieza de la pregunta:** Una vez realizada la pregunta se procede a eliminar signos de interrogación. Luego se corta el Token que identifica el tipo de pregunta; por ej.

   ¿Quién es John Lennon?

   quedaría como

   Quien es John Lennon

   y el identificador de pregunta sería **Quién**
3. **Búsqueda de páginas en la web:** Para realizar la búsqueda se añade un string en una url de google.

```
print 'Ingrese busqueda...'
question = raw_input()
questions = re.split(r'[^\w\s\á\ó\í\é\ú]', question, re.U|re.I)
quest = ' '
for q in questions:
    if q != '':
        quest += q

formal_question = quest.replace(' ','+')
print '\nBuscando',quest,'...\n'
text = nltk.tokenize.word_tokenize(quest)
my_url = 'https://www.google.ru/search?q='+formal_question
print my_url
```

Figure 2:Segmento de codigo donde se ingresa la pregunta a un link generico de google

4. **Extracción de respuestas:** Para extraer las respuesta se ingresa a cada una de las páginas utilizando los **request** de python. Luego con **BeautifulSoup** se extrae el texto plano (eliminando los identificadores sintácticos de HTML). Finalmente, utilizando **expresiones regulares**, extraemos las oraciones candidatas para ser respuesta.

Entregando Respuesta

1. **Clustering de respuestas:** Una vez extraída las oraciones realizamos una representación vectorial de cada una de las sentencias. Utilizamos **Agglomerative Clustering** [1] con métrica de **distancia coseno**[2]. Los vectores ingresan como input en el modelo. Luego obtenemos los clusters; en este caso utilizamos **4 clusters**. Cada cluster representa un tipo de respuesta
2. **Selección de respuestas:** Seleccionamos la primera oración de cada cluster asumiendo que los primeros párrafos en un texto tienen más información global. Esto, porque la estructura de un corpus se torna más detallada a medida que avanzamos en los párrafos.
3. **Evaluación:** Para evaluar utilizamos Mean Reciprocal Rank.

$$MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i} \qquad (1)$$

Donde Q son la cantidad de palabras y $rank_i$ la cantidad de palabras que coinciden con la raiz. Por ej.

Query	Proposed Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
tori	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

Dependiendo del tipo de pregunta (Quien o Donde) buscamos apariciones de palabras claves en un diccionario. Por ejemplo, si buscamos un lugar, entonces deberá aparecer el nombre de un país o ciudad.

Resultados

Para medir los resultados, se hizo uso solo del Rank. Lo anterior debido a que solo se contaba con un diccionario de palabras clave para cada pregunta. No se descarta la posibilidad de a futuro enriquecer estos diccionarios de manera de realizar el *Mean Reciprocal Rank*. La figura 3 muestra los clusters generados y la figura 4 el ranking generado con las principales respuestas.

	answer	rank
1	fue un artista, músico multiinstrumentista, po...	3
3	Es difícil imaginar cómo sería hoy este artist...	1
0	Fue asesinado tres semanas después de su lanza...	0
2	es responsable de veinticinco sencillos número...	0

Figure 4:ranks asociados a las ocurrencias de palabras clave en la consulta *Quien fue John Lennon*

Conclusion

Se ha realizado un sistema simple de pregunta y respuesta basado principalmente en expresiones regulares. El uso de Clusters permite separar las respuestas en grupos cuyo contenido es similar. A futuro, en vez de considerar clustering podríamos utilizar un modelo de clasificación de respuestas basado en Redes Neuronales Recurrentes.

Important Result

Fue asesinado tres semanas ... fue asesinado por un ...	fue un artista, músico multi instrumentista, poeta, dibujante, actor,...
es responsable de veinticinco sencillos número uno en el Billboard Hot 100.	Es difícil imaginar cómo sería ...

Figure 3:4 cluster para la consulta "Quien es John Lennon"

Pregunta	Mejor Respuesta
¿Quien fue Gabriela Mistral?	fue una poetisa, diplomática y pedagoga chilena.
¿Quien fue Alan Turing?	fue un pionero de los campos de la computaci...
¿Donde está Lota?	ubicada en la provincia de Concepción, región del Biobío
¿Donde está la Muralla China?	ubicada a menos de 80 kilómetros de Pekín...

References

[1] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

[2] John R Smith. *Integrated spatial and feature image systems: Retrieval, analysis and compression.* PhD thesis, Columbia University, 1997.

Contact Information

- Email: cridonoso@inf.udec.cl
- Phone: +569 774 658 09