Sistema de Recomendación

Miguel Ángel Porras Naranjo

6 de abril de 2017

Descripción del problema.

Una empresa ha creado un Software Libre gratuito para resolver problemas de Big Data. Además ha creado un foro de ayuda y sugerencias para los usuarios se ayuden entre sí.

Si antes de abrir una pregunta, hubiera una lista de posibles preguntas ya resueltas que sean parecidas o iguales a la indicada por el usuario, ofrecería un mejor servicio al usuario ya que no tendria que esperar a una respuesta.

Elementos necesarios.

Para ello crearemos una base de datos de datos con las preguntas que se consideren resueltas. En la primera columna tendremos el `id` de la pregunta. Para el resto de columnasm elaboraremos mediante una lista de palabras más frecuentes quitando las palabras que no tengan significado por sí mismas, por ejemplo determiantes, preposiciones, etc. Si aparece en la pregunta valdrá 1, y 0 en caso contrario. Por ejemplo, tenemos las preguntas:

- ¿Quiero ordenar los elementos de una lista y no encuentro ningun ejemplo AYUDAAAA!!!!!!?
- AMIGOS COMO HAGO PARA ORDENAR ELEMENTOS DEJO 5 ESTRELLAS
- Tengo problemas para representar los puntos en un mapa. ¿Algún código de ejemplo?.
- Me encantaria representar una estrella con elementos de una lista, es para mi amiga la Yoli

La base de datos tendría la siguiente forma.

id	ordenar	elementos	lista	estrella	respresentar	puntos	mapa	código	ejemplo
1	1	1	1	0	0	0	0	0	0
2	1	1	0	1	0	0	0	0	0
3	0	0	0	0	0	1	1	1	1

Montando el sistema de recomendación.

Para ver que preguntas mostrar en la caja, montaremos un algoritmo de Machine Learning. Usaremos el popular `XGBoost` para este caso. Nos encargamos de entrenar el modelo con la base de datos que ya tenemos.

Una vez montado el modelo, podríamos usar alguna herramienta como Docker para la implementación en la página web. En la caja de sugerencias, mostraremos el top 7 de las preguntas que el modelo considera que son más parecidas a la pregunta actual del usuario.

Posibles mejoras al modelo.

Un problema fundamental de este modelo es que es muy sensible a los sinónimos y faltas de ortografía. Esto se podría mejorar de varias maneras:

- Implementar un autocorrector para las posibles palabras clave. Si `elemento` es una palabra clave, entonces si el usuario escribe `elmento` conseguir que el sistema sea capaz de corregirlo.
- Que sea capaz de comprobar si las palabras son parecidas mediante alguna red de palabras que exista en la red y mejorar las predicciones del modelo.