Blocket Price Recommendation System

Search Engines and Information Retrieval Systems, DD2476

Andreas Rosenback
rosenba@kth.se

Dušan Viktor Hrstić
hrstic@kth.se

Sandra Pico
sandrapo@kth.se

Suhas Sheshadri
suhass@kth.se

October 16, 2018

Abstract

The following price recommendation system is able to provide users useful insight regarding a product that they would like to sell / buy in a single place. The system is based on the advertisements available on Blockets website [1]. It fetches all the advertisements from `www.blocket.se`, indexes it in the local system and keep it ready for users to get useful information from it. This is done by typing the query entered by the user which describes the product, during their search operation. The users are provided with an option to select appropriate categories and their attributes (if any present, as seen on Blocket). The final information, based on the search query and the categories / attributes selected are displayed on the screen. The information presented to the user shows a range of prices for which a user could sell the product, like the highest, lowest and average prices.

Contents

Introduction

The system works in a simple yet novel way to provide price recommendations. It utilizes the Blocket website, which is one of Swedens most popular online market for buying and selling where around 98 % of Swedish population knows about Blocket and 25% of what is posted on Blocket is sold within a day, 60% is sold within one week [2]. The total value of what was announced in 2016 was around 616 billion SEK which corresponds to 14% of Sweden's GDP and every day, there are about half a million advertisements on Blocket [2]. Since Blocket has so many users visiting each day and a large number of advertisements put up every day, when combined with the above-mentioned facts, can be used in an innovative way to provide price recommendations.

The system first fetches all the available advertisements present on Blocket by crawling the data from its website [1]. It is then indexed and stored in the local system. This data is used as a reference for future price recommendations. When a user wants to sell anything, he/she can inquire about the product on the system. The system, based on the search query provided by the user and the data present in our system, along with the use of elastic search will recommend a price. The elastic search takes into account not only the data from Blocket, but also the categories and/or their attributes the user has selected during the initial query. Based on this, a score is given to the initial query and then it is used to find same / similar products from the data. Finally,it recommends the minimum, maximum and average price for a particular product.

The recommendation systems can be of several types such as Collaborative Recommender system, Content-based recommender system, Demographic based recommender system, Utility based recommender system, Knowledge based recommender system and Hybrid recommender system [3]. The system in consideration is a hybrid system comprised of Content-based, Demographic and a custom scoring system. The advertisements and their attributes represent the content while the regions represent the demographic systems. A custom two-layer scoring system is implemented over and above these systems to build a novel and effective price recommendation system.

Considering the amount of advertisements, current market trends and the scoring system for the query it may be concluded that the system is robust and complex which requires a good division of subproblems and later their correct assembly which is explained in the Method part of this report.

Related Work

There are various types of recommendation systems which can be used to provide price recommendations [4]. These can be based on specified categories, ratings given by fellow users and the results that are the most similar results to the query to name a few. But, when it comes to selling of used item on the Internet, there is no predefined way to provide any price range for an item to be sold. One possibility may be to consider the original price of

the goods and then depreciate its value based on different aspects like date of manufacture, working condition (in case of electronics), present aesthetic conditions, ease of maintenance, warranty / guarantee remaining, emotional value attached with the item etc.

Method

For the ease of understanding, the recommendation system is segregated into parts and their roles are explained in detail.

Crawler

The system crawls the Blocket [1] website for advertisements. Blocket can be accessed based on the regions interested. The crawling is done keeping this in mind. The system crawls for a specified category for a specified region. This method of breaking up of categories is not just done to reflect what Blocket has, but also to provide more relevant recommendations based on regions. Also, a one-time crawling is done to fetch all the regions and also all the categories and their respective attributes. This information is later used in the UI to represent custom screens and values.

UI

The UI has been designed in a tidy and user friendly way, at the same time reflecting the Blocket website keeping in mind the users ease of relating to it. First, an option is provided to the user to select a region which they are interested in.

In the subsequent screen, the user is given with a search box and a drop-down menu. The menu contains all the categories offered by Blocket. The system currently ignores the categories such as Jobs and Bostad, since these are implemented in a different way and also as a fact that they are not useful to the user in the current context. The category field is implemented as a mandatory field, keeping in mind the plethora of advertisements present on Blocket. Choosing a category, in turn shows all the respective attributes associated with it. This is implemented to make the user experience easy. Having all the attributes for a particular category, user has the power to chose the attributes which he/ she is sure about. Selecting more attributes helps in providing a better price recommendation.

Scoring

The system is implemented as a two-layer scoring, based on the search query and the attributes. First, using the query terms itself, which is split into separate words and these are checked in the title of the advertisements for a match. Higher the number of matching terms, higher the score. Secondly, using the attributes selected. The system compares the user selected attributes with the attributes of the advertisements and give a score corresponding

to the number of attributes which match.

The score function can be represented as:

Score = W1*(matching terms/total terms) + W2*(matching attributes/total attributes)

where,

- W1: Weight given to the terms scoring
- W2: Weight given to the attribute scoring
- Note: W1 + W2 = 1

The system is currently using the values of 0.7 and 0.3 as the weights W1 and W2 respectively. This means that more weightage is given to the query terms than to the attributes. The attributes are used in conjuncture to the query terms and also help to filter the results so as to get more relevant ones. It is implemented in this way due to the fact that the attributes are definite and provides less information about the specific product in comparison to the query terms.

Elastic Search

In order to be able to index and search different advertisements that previously have been fetched from Blocket the Elastic Search engine [5] have been used. In order to implement it two functional parts have been programmed using the Elastic Search API [6], i.e Indexing and Searching.

Indexing

Once all the advertisements are crawled from the Blocket website, it is passed into the Elastic Search as a list of Ad objects for indexing. Each Ad has a JSON representation (Appendix A) that is sent and indexed into the Elastic Search. The indexed data is stored in the local system for easier and faster access. This is done to overcome any delays which can be caused if the crawling is done online (at the time of the search) since there can be more than half a million of advertisements at any given moment on the website. The advertisements are indexed based on the categories as seen on Blocket.

The indexed advertisement can be checked on the browser with a link similar to:

http://localhost:9200/bilar/ad/15000

where,

- bilar is the category
- ad is the name of the type indexed

- 15000 is the id of the current advertisement in consideration

This provides a detailed index as shown in (Appendix B)

Here, details like category, type, id, price, title, region and attributes like Modellar, Tillverkningsar, Bransle, Vaxellada can be seen. These attributes are the ones which are present for a particular advertisement considered.

Searching

As and when a search is carried out by the user, the UI will send the search query and the category along with its attributes to the Elastic Search. Based on the attributes selected by the user, the query can either be a Range query, a Term query or a Boolean query or a combination of these. The Elastic Search using the above information, finds the relevant results from the indexed data and communicate it back to the UI.

Integrated System

After implementing the above parts, a Model View Controller [7] architecture is implemented to consolidate these parts to form the current system as shown in Figure 1.

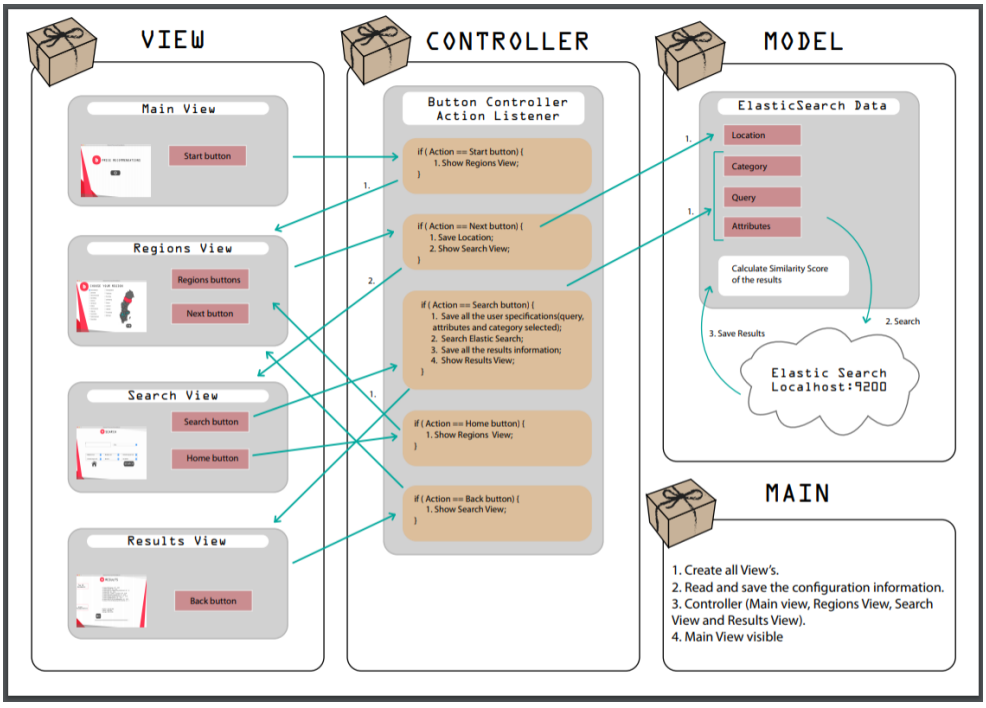

Figure 1: The Model View Controller architecture implemented in the system.

There are two systems which are used one after the other. First, the crawling system crawls the Blocket website, for a particular category, region by region. All the advertisements and their attributes are fetched. This information is then sent to the Elastic Search which indexes and stores it on the local system.

The price recommendation system is then started. The user first selects the region they are interested in. Choosing the region will help the user to get relevant price recommendations based on the region selected. The user, on the next screen, types the search query and selects a category. Upon the selection of category, a set of associated attributes are shown. The user can now select the attributes. Again, selecting as much attributes as possible will improve the results since these attributes are also considered for scoring by the system. When the user makes a search with the query and attributes, this information is then passed on from the UI to the Elastic Search. The Elastic Search will search the already indexed data based on the information it receives. A scoring function inside the Elastic Search will score the advertisements and provide the top results to the UI. The system then represent these results in a graphical manner providing the user with insights such as the least price, the maximum price and the average price what he can expect for the searched item.

Results

There are around 0.1 million advertisements indexed for the category of cars. The system, for a given query provides a price recommendation based on different aspects of the product such as attributes and query terms as described earlier.

Tests

Four different test cases were carried out on the system. All of them was filtered on the region 'Västerbotten'. The total weight of query and attributes are summed to one. More detailed information about each case is given Table 1. Screen-shots for two test cases are shown in (Appendix C).

Table 1: Parameters of the tests

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Query-weight	0.1	0.9	0.5	0.5	0.7	0.7	0.7
Attribute-weight	0.9	0.1	0.5	0.5	0.3	0.3	0.3
Query	volvo s80	volvo s80	volvo s80	volvo v70 11	volvo	volvo s80	volvo s80
Modellar fran	2010	2010	2010	-	-	-	2010
Modellar till	2018	2018	2018	-	-	-	2018
Vaxellada	Manuell	Manuell	Manuell	Automat	-	-	Automat

Discussion

In the Results page, there are two sections. First, the top ten results are displayed along with the title, price and the score calculated for that advertisement. This is calculated based on the top fifty results depending on the scoring done. This is just to give an insight to the user regarding the price recommendations. Secondly, the section where the highest, the lowest and the average prices are shown.

The system was tested using different combinations of weights for the query terms and the attributes. In the case of former, the results were more oriented towards the query and the attributes were used more like a filtering criteria. In the case of latter, the results were more inclined towards the matching of the attributes. Some results with non query terms in the advertisements made their way up the list since they had more attributes matching to the ones selected during the search. This case can be used when the user is not specific about a certain car (brand / model), but more interested in the specifications of the car.

Under both the above sets of cases, the system provided with the respective price recommendations and the relevant top results. It can be concluded that the system is working as intended.

Note that the results may vary from time to time, based on the advertisements present on Blocket during indexing.

Future Work

The system does not consider the regions 'Norrbotten' and 'Skåne', as these regions are made up of several sub-regions. It would be good to implement the recommendation system considering these regions as well.

The current system is programmed to work only on the category Bilar (Cars). This is done due to the fact that this category has a good number of attributes associated with it and many other categories do not have any attributes at all. Also, attributes of certain categories are maintained as links in Blocket. This was a problem during indexing these categories.

References

[1] "Blocket," May 2018-05-15. [Online]. Available: https://www.blocket.se

[2] "About blocket," May 2018-05-15. [Online]. Available: http://www.blocket.se/omblocket.htm

[3] B. Dhillon, "Classifying different types of recommender systems," nov 2015. [Online]. Available: https://www.bluepiit.com/blog/classifying-recommender-systems/

[4] A. Idrees and S. Taie, "Online price recommendation system for shopping used cell phones," vol. 13, pp. 15–23, 07 2016.

[5] E. BV, "Elastic search," may 2018, https://www.elastic.co/.

[6] ——, "Document apis," may 2018, https://www.elastic.co/guide/en/elasticsearch/reference/current/do

[7] D. Patterns and C. Pattern, "Model-view-controller," *Microsoft Patterns & Practices, http://msdn. microsoft. com/practices/type/Patterns/Enterprise/DesMVC*, 2003.

Appendix A

```
public class Ad {
        private String titleName = "title";
        private String attributeName = "attributes";
        private String priceName = "price";
        private String regionName = "region";
private Map<String, Object> json = new HashMap<String, Object>();
}
```

Appendix B


```
{"_index":"bilar","_type":"ad","_id":"15000","_version":1,"found":true,"_source":
{"price":239000,"attributes":{"Modellar":"2015","Miltal":"5250",
"Tillverkningsar":"2014","Bransle":"Diesel","Vaxellada":"Automat"},
"title":"BMW 320 d xDrive Touring / Automat / Navi -15","region":"varmland"}}
```


Appendix C

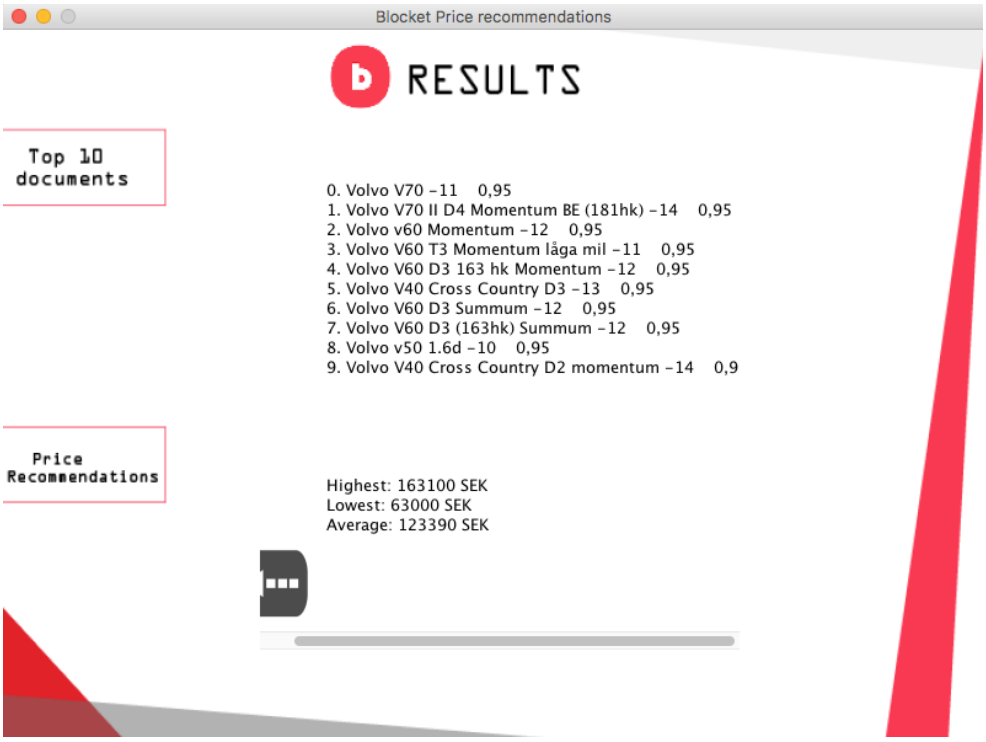

Figure 2: Result for Test Case 1.

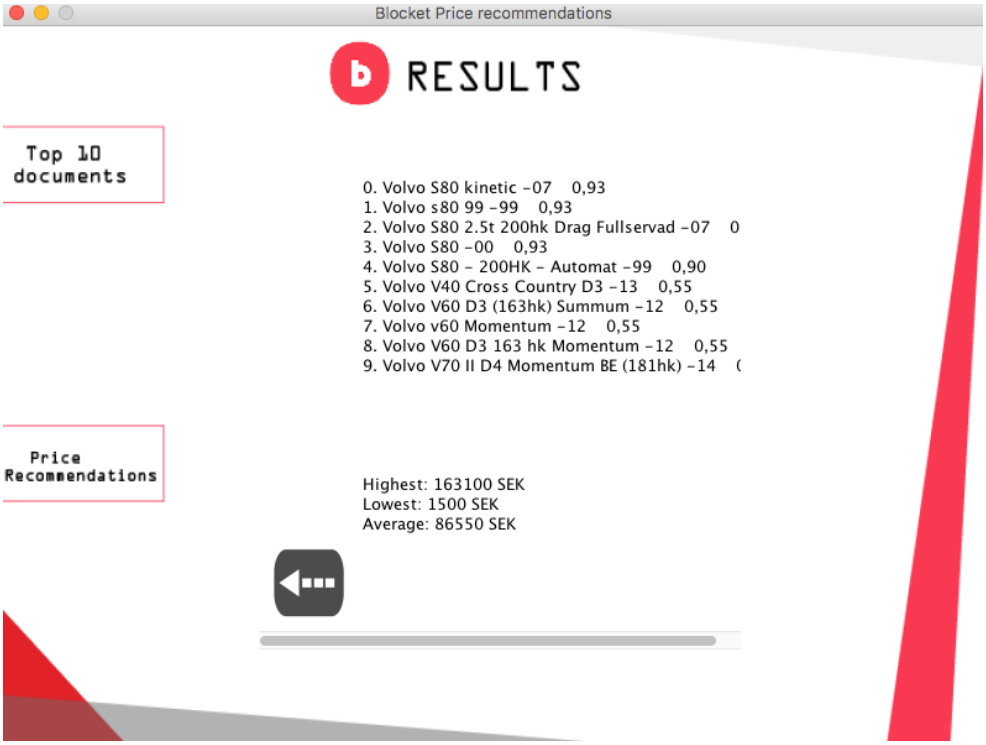

Figure 3: Result for Test Case 2.