Universitat Oberta de Catalunya

uoc.edu

A Product Data Matching System for an e-Commerce Aggregator

Teguayco Gutiérrez González
Master's Degree in Data Science
Data Mining & Machine Learning

Laia Subirats Maté (Consulting Professor)
Jordi Casas Roma (Responsible Professor)

June 2019

SHEET OF WORK

Title:	A Product Data Matching System for an e-Commerce Aggregator
Author:	Teguayco Gutiérrez González
Consulting Professor:	Laia Subirats Maté
Responsible Professor:	Jordi Casas Roma
Delivery Date (mm/yyyy):	06/2019
Studies:	Master's Degree in Data Science
Area:	Data Mining and Machine Learning
Language:	English
Keywords:	Product Matching, e-Commerce, Machine Learning
Abstract	
**TODO	

Table of contents

1. Introduction 1
1.1 Background and motivation 1
1.2 Objectives 1
1.3 Approach and methodology 2
1.4 Planning 2
1.5 Summary of obtained products 2
1.6 Description of the chapters in the memory 2
2. State of the art 3
2.1 Introduction 3
2.2 Approaches 3
2.2.1 Text similarity 3
2.2.2 Attribute extraction 4
2.2.2.1 Dictionaries and regular expressions 4
2.2.2.2 Sequence labelling 5
2.2.3 Image recognition 5
3. 7
3. Conclusiones 9
4. Glosario 10
5. Bibliography 11
6. Anexos 12

List of figures

No se encuentran elementos de tabla de ilustraciones.

Acknowledgements

TOD

1. Introduction

1.1 Background and motivation

Nowadays we can find on the Web a bunch of different e-Commerce aggregators, which are websites that collect data of products from different online shops such as Amazon, MediaMarkt or ToysRUs with the aim of showing their customers the best options to buy a certain product. Google Shopping or Idealo are two famous examples of e-Commerce aggregators.

However, the problem of identifying the same products collected across many sources, known as “Product Matching”, must be addressed in order to offer an optimal service. In most cases this is not a trivial task, as data is presented in different formats depending on the source they come from.

In this work, it is proposed the construction of a system using Machine Learning and Natural Language Processing (NLP) techniques to automatically detect matches between products whose data come from different sources. In particular, this product data matching system will be developed based on the data products collected by a real-world company that has developed an e-Commerce aggregator. This matching system would help this company to automate the product matching task (which is currently done manually), thereby freeing human resources to attend to other tasks.

1.2 Objectives

The main objective of the work is to develop one or more than one Machine Learning classifier that can be able to detect matches, with considerable confidence levels, between products coming from different web shops and products known for the company contained in its master-data storage. Thus, the matching will be mainly performed based on the product titles.

These products to be matched will be initially electronical products such as computers or mobile phones, which are the predominant types of products the company operates with.

Furthermore, the following can be taken as secondary objectives:

- Develop an app to input data for a certain product to output the best match for it. This app can serve as a proof of concept, so that it can be extended in the future according to the specific needs of the company.

- Extend the ML classifiers to work with more heterogeneous products, e.g. toys, CDs, televisions, etc.

1.3 Approach and methodology

Given the large variety of products the company handles and the amount of Machine Learning classifiers that can be applied, a progressive approach will be used throughout the development of this work. So, to begin with, ML techniques will be implemented for an homogeneous set of products (e.g. electronical devices). Thus, if time allows, more heterogeneous datasets will be generated and used to train different ML models each time.

Furthermore, as there are a lot of works addressing the “Product Matching” problem, the already-used approaches and techniques will be thoroughly analysed to then apply the subset of them which best fit for the specific product data the client company owns.

1.4 Planning

Descripción de los recursos necesarios para realizar el trabajo, las tareas a realizar y una planificación temporal de cada tarea utilizando un diagrama de Gantt o similar. Esta planificación tendría que marcar cuáles son los hitos parciales de cada una de las PEC.

1.5 Summary of obtained products

No hay que entrar en detalle: la descripción detallada se hará en el resto de capítulos.

1.6 Description of the chapters in the memory

Explicación de los contenidos de cada capítulo y su relación con el trabajo en global.

2. State of the art

2.1 Introduction

As explained in the introductory section, the problem of matching products coming from different sources brings with it the challenge of identifying identical products whose data may be missing (for instance, universal identifiers such as EAN or UPC codes could not be found) or presented in multiple formats. These difficulties make this matching task to be a non-trivial one and, as a result, a lot of research studies and tools have already arisen, most of them implementing Natural Language Processing (NLP) and Machine Learning techniques.

In this section, a comprehensive review of these previous works will be done based on the approaches they pose to analyse how they implement the solutions for this issue. It should also be noted that some of these works use more than one approaches.

2.2 Approaches

2.2.1 Text similarity

The first approach could consist in comparing two product titles or descriptions to determine their degree of similarity, as the same product offers coming from different web shops should have similar titles and descriptions. In NLP, a common solution is to use TF-IDF vectorization and calculate some measure, like the cosine similarity, to quantify the degree of similarity between two titles. TF-IDF vectorization could constitute a first valid approach, as it is a method which is able to identify the most important words in a given document.

However, in [1] they pose an adaptative string similarity measure that improves the matching results using TF-IDF and therefore demonstrate that the latter method has limitations when addressing the product matching problem. This proposal is based on giving higher weights to those terms in product titles that are more relevant to identify the product, as depending on the product, some terms will identify them better than others. For example, to identify a camcorder like the following:

Canon Vixia HF S10

The manufacturer code (HF S10) should have a high weight, as it is the most relevant term to identify the product. However, for the following cleaning toolkit for the former camcorder:

Canon Vixia HF S10 Cleaning Toolkit

The type of product (cleaning toolkit) should have the highest weight in order to avoid matching it with the former camcorder.

Word embeddings are another alternate approach to handle words as vector representations. Word2Vec [8] and GloVe [9] are possibly the best known developed techniques to get such representations of words.

2.2.2 Attribute extraction

Most of the times, product titles include technical information which is crucial to identify them. However, in order to apply Machine Learning techniques, a more structured data about products should be managed. Therefore, the problem of obtaining these structured data about products from their textual information is arisen.

As an example, from the following product title:

Apple iPhone XS (64GB) – Gold

It would be desired to filter it to obtain the following structured data:

```
{
   “Brand”: “Apple”,
   “ProductName”: “iPhone”,
   “Model”: “XS”,
   “MemoryCapacity”: “64”,
   “MemoryCapacityUnit”: “GB”,
   “Color”: “Gold”
}
```

This problem can be seen as an instance of an NLP task known as **Named Entity Recognition** (NER). The goal in NER is to locate and classify the entities mentioned inside some given unstructured text.

2.2.2.1 Dictionaries and regular expressions

One straightforward solution may consist in using **regular expressions** and **dictionaries** of known values for the attributes to extract. However, this approach can have several disadvantages.

On one hand, the use of regular expressions could lead to an inaccurate identification of attributes. For example, given the product name “Apple iPhone 32GB”, a regular expression can be written to extract the phrase “32GB”. However, this method would not be able to correctly resolve if this value correspond to RAM memory or to hard-disk capacity, as both attributes are commonly given in GB. Furthermore, as the information may be presented in many different formats, a regular expression should be written to handle every instance. For example, the inches of the screen for a certain TV could be written like “50-inches” or ‘50”’, so two different regular expressions are needed in this case.

In order to bypass the problem of writing a lot of regular expression to handle every possible case, studies address the issue of “learning” regular expression [4, 5, 6]. Genetic programming is the favourite used approach when exploring this issue.

On the other hand, the use of dictionaries may be a good approach for attributes with a fixed list of possible values like the colour. However, it could fail with attributes with an open list of possible values such as the brand, as new brands come eventually out on the market. In this latter case, the dictionary used to perform brand extraction should be regularly updated.

2.2.2.2 Sequence labelling

Some studies [2, 3, 10] address this issue by implementing more sophisticated methods. They first train some NER model to extract products features from their titles to then train Machine Learning models based on the structured data obtained to make the matching.

Thus, ***Conditional Random Fields*** (CRF) can be seen as the favourite alternative, as it is used by the three of the above-cited works. It is a linear model for sequential labels. In fact, CRF can be considered as the sequential version of logistic regression [7].

Another implemented solution by [2] along with CRF is ***Structured Perceptron***, which is a supervised learning algorithm. By contrast, [3] goes beyond implementing CRF also with **text embeddings**.

The benefit of these sequence labelling algorithms is that they leverage the information given by the context of a given word in a certain product title. For example, in a product title like “Apple iPhone 4 (4GB RAM) - Black”, “4GB” could be recognized as RAM memory considering the next word is “RAM”. Using some of the above-mentioned simplest solutions, this information would not be exploited.

Nevertheless, sequence labelling algorithms need a significant sizable training set (usually labelled using BIO encoding) whose manual labelling would require a huge usage time. Some implemented alternatives to avoid this manual task consist in the use of “distant supervision”, in which a training dataset is automatically built based on heuristics and rules.

2.2.3 Image recognition

Matching products identifying these by performing image recognition on them could constitute the hardest approach of all. It is not only the complexity of models like ***Convolutional Neural Networks*** (CNN), which are a common solution for image recognition problems, but also some other challenges that have to be faced regarding products recognition from images: the same product can be found photographed from different perspectives, with different colours or different levels of brightness. Furthermore, CNNs could need a huge number of

images to be trained, something that translates into managing and storing a lot of bytes of data.

Indeed, [3] implements also a CNN on the premise that most of the web shops use the same image for identical products. Treating the product matching problem as a two-class classification problem (given a pair of products, decide if they match or not), the implemented CNN allows to obtain image embeddings for the candidate products pair. Then, cosine similarity between both vectors is calculated to determine if they are similar enough to match. However, this work consider similarity image embeddings as a weak indicator for product matching, as it performs better for the category prediction task.

3. Design and implementation

3.1 Introduction

As pointed out in the introductory section, when building an e-Commerce aggregator, it is vital to address the **Product Matching** problem, as the capability of identifying the same product from different incoming product offers brings a high quality for a service such as this one. Thus, getting all the offers for a given product is possible, so that a client can compare among the existing offers in the market and then choose the one that best fits their needs. Furthermore, even a certain seller or supplier can use this service to study their competitors' offers.

The company which provides the data for the implementation of this work is currently tackling the Product Matching issue by manually assigning the new incoming product data offers to their corresponding base products, which are already existing in the company's database. The figure below illustrates how three offers for the same product coming from three different sellers point to the same basic product in the company's storage:

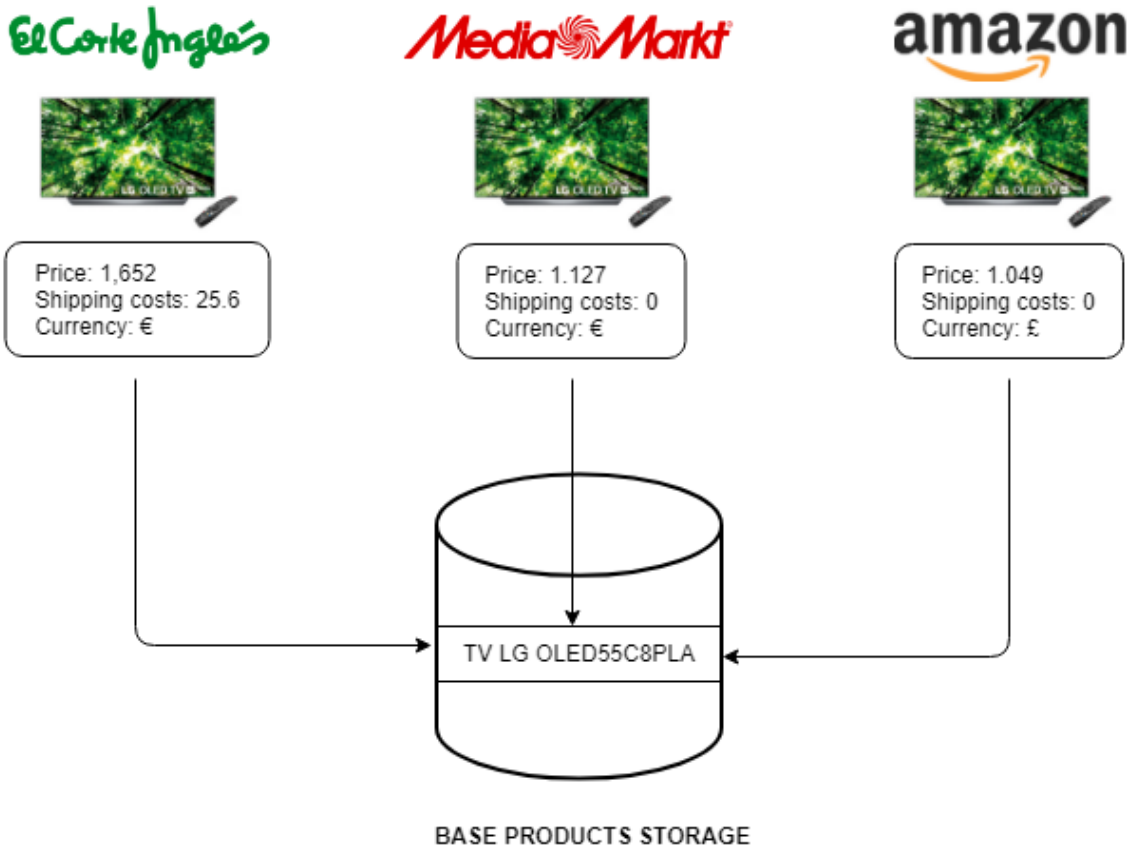

Figure 1: Product Matching for TV offer

The operators who perform this task usually navigate to the source URL of the crawled product, study their features, compare them to the most similar base products stored in the database and, if a good candidate is found, the matching or assignment is done. If none of the existing base products fits well enough with the incoming product offer, a new base product must be created in the database.

3.2 Problem statement

3. Conclusiones

Este capítulo tiene que incluir:

- Una descripción de las conclusiones del trabajo: Qué lecciones se han aprendido del trabajo?.
- Una reflexión crítica sobre el logro de los objetivos planteados inicialmente: Hemos logrado todos los objetivos? Si la respuesta es negativa, por qué motivo?
- Un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto: Se ha seguido la planificación? La metodología prevista ha sido la adecuada? Ha habido que introducir cambios para garantizar el éxito del trabajo? Por qué?
- Las líneas de trabajo futuro que no se han podido explorar en este trabajo y han quedado pendientes.

Puntos a mencionar:

- Por qué se escoge aproximación CRF + ML
- Cómo se encara el problema: problema de clasificación con ProdanetID = variable de clase
- Fundamentos teóricos
    - CRF
    - ML classifiers
        - KNN
        - Logistic regression
        - SVM
        - Neural Networks?

4. Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

5. Bibliography

1. Andreas Thor. (2010). Toward an adaptive string similarity measure for matching product offers. In GI Jahrestagung (1), pages 702–710.

2. Ajinkya More. (2016). Attribute Extraction from Product Titles in eCommerce. WalmartLabs, Sunnyvale CA.

3. Petar Ristoskia, Petar Petrovskia, Peter Mikab, and Heiko Paulheima. (2017). A machine learning approach for product matching and categorization. Semantic web.

4. Petrovski, P., Bryl, V., Bizer, C. (2014). Learning regular expressions for the extraction of product attributes from e-commerce microdata.

5. Svingen, B. (1998) Learning regular languages using genetic programming. In Genetic Programming 1998: Proceedings of the Third Annual Conference.

6. Morgan Kaufmann, M. 1998. Alberto Bartoli, Giorgio Davanzo, Andrea De Lorenzo, Marco Mauri, Eric Medvet, and Enrico Sorio. Automatic generation of regular expressions from examples with genetic programming. In Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO ’12, pages 1477–1478, 2012.

7. Charles Sutton, Andrew McCallum. (2012). An introduction to Conditional Random Fields. Fundations and Trends in Machine Learning. Volume 4 Issue 4, April 2012. Pages 267-373.

8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. pp. 3111–3119.

9. Pennington, J., Socher, R., Manning, C.D. (2014). Glove: Global vectors for word representation. EMNLP. vol. 14, pp. 1532–1543.

10. Mikhail Sidorov. (2018). Attribute extraction from eCommerce product descriptions. CS229.

6. Anexos

Listado de apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido (por ejemplo, manuales de usuario, manuales de instalación, etc.)

Dependiente del tipo de trabajo, es posible que no haya que añadir ningún anexo.