UNIVERSITY OF IOANNINA
SCHOOL OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY CAMPUS, IOANNINA

October 16th 2015

Discovery of associations between technical and lexicographical attributes extracted from internet reviews

Dimitriadou Despoina

SUPERVISING PROFESSOR: P. Tsaparas

UNDERGRADUATE
THESIS

Foreword

The present thesis was conducted between October 2014 and October 2015 under the supervision of professor Panagiotis Tsaparas, to whom I am grateful for the valuable help he provided where needed.

October 16th 2015

Despoina Dimitriadou

Summary

The present thesis is a first attempt at discovering possible associations between technical characteristics taken from a product catalogue and lexicographical terms taken from the corresponding products' online reviews. This is achieved by using known statistical methods.

The goal is to build a system which will collect a dataset of digital cameras' information and will produce a list of sorted lexicographical terms for each technical characteristic depending on how relevant they are with its appearance in the list of the product's characteristics.

For this goal to be fulfilled, we compute the tf-idf scores of all lexicographical terms to find those most important per review collection and as a next step we apply Pearson's Chi-Square Test for Independence on the first 50 terms from each and on some selected technical attributes. We consider an association important when the p-value < 0.10.

The discovery of strong associations between any technical characteristics and lexicographical terms that describe the technical characteristics well or give insightful information, for example, about the users commenting, could later be utilized to improve the search results in product catalogs or for the purposes of recommendation systems.

Keywords: technical characteristics, user reviews, chi - square test for independence, tf-idf

Table of Contents

1 Introduction 1

1.1 Subject of thesis 3

1.2 How the volume is organized 5

2 Terminology and relevant studies 6

2.1 Formal definition 6

2.2 Relevant studies 7

2.2.1 Learning to Question: Leveraging Preferences for Shopping Advice 7

2.2.2 Structured Query Reformulations in Commerce Search 8

2.3 Goal 9

3 Collection of the data 10

3.1 Retrieving the data 10

3.1.1 Best Buy API 11

3.1.2 JSON Format 12

3.1.3 Products API 14

3.1.4 Reviews API 15

3.1.5 Deciding the query on the API 15

3.2 Data statistics 17

4 Preprocessing of the data 20

4.1 Reviews 20

4.1.1 Lexical and Linguistic Processing 21

4.1.2 Removing rarest words 22

4.1.3 Tf-idf score calculation 22

4.2 Technical Characteristics 23

5 Analysis of the data 26

5.1 Discovering associations 26

5.1.1 Chi - Squared Test for Independence 26

5.1.2 Frequencies table 28

6 Experiment Results..30

6.1 Data used in experiments..30

6.2 Results and Evaluation..31

6.2.1 Results..31

6.2.2 Evaluation of Results..38

6.3 Conclusions..40

7 Epilogue..41

7.1 Synopsis and conclusions..41

7.2 Future extensions..41

8 References..42

1

Introduction

With the spread of webstores and the development on information retrieval systems, an important benefit for the buyers is the fast search inside catalogues that contain hundreds or maybe thousands of products and the overview of the information that accompanies them, including photographs, reviews, technical characteristics, and several other sorts of information. From these information categories, technical characteristics are almost always included in catalogues, either briefly or with more detail. Reviewing the products has also become a common practice for the users, resulting in a minimum of hundreds of reviews per product in big webstores (e.g. Amazon[1], Best Buy[2]).

Because of the ignorance most users have concerning technical details for the products they look for, the queries that webstore search engines have to process may contain words that can not be directly matched with the products' technical information in the catalogue. Most searches are made by the users as free-form queries and usually without them taking into account the structure of the catalogue in which they perform the query. Often the search is based more on the needs of the user and less on the target

[1] https://www.amazon.com

[2] http://www.bestbuy.com

product and thus may be either descriptive or ambiguous using words that are probably not contained in the catalogue structure. For example, a query may be *“lightweight digital camera”*. The list of product’s specifications usually contains the weight of the product in kilograms or grams but it is not very common to describe the product as *“lightweight”* or *“heavy”*. Such a qualitative description is most likely to appear in user vocabulary contained in product reviews. We could say that the information included in user reviews forms a second unofficial description of the products, this time from a user’s perspective, which can be very useful when trying to answer free-form queries.

In short, a simple text search in a catalogue for the words users use in queries and therefore in their everyday life when describing a product as far as technical details are concerned, is not guaranteed to give results. On the other hand, the search in the reviews for this product could give some results, providing that there *are* reviews for this product, something that may not be true for new products on the catalogue or generally on the market.

For now, the problem of finding better results to satisfy the needs of the users has lead researchers to look for associations between words from queries made on search engines and technical characteristics extracted from their routes on webpages (browser trails) [1] and between words from reviews or user tags and the technical specifications of the products that these describe [2].

A similar approach is attempted here, so that words from user reviews are found to be associated not with the products themselves, but with the products’ technical characteristics. The goal is to find if there are words in the reviews that can describe well certain technical characteristics.

The highlighting of such associations would help us better understand how users perceive the technical specifications of the products they buy and use this knowledge to improve the results in online catalogue product search. Also, this approach could solve the problem of searching catalogues with zero reviews; using the associations we have found between technical and lexicographical characteristics we can, based on their technical specifications, relate keywords to products with the same specifications even if they do not have reviews. Further, the lexicographical description of the products can be used also in other applications, such as recommendation systems.

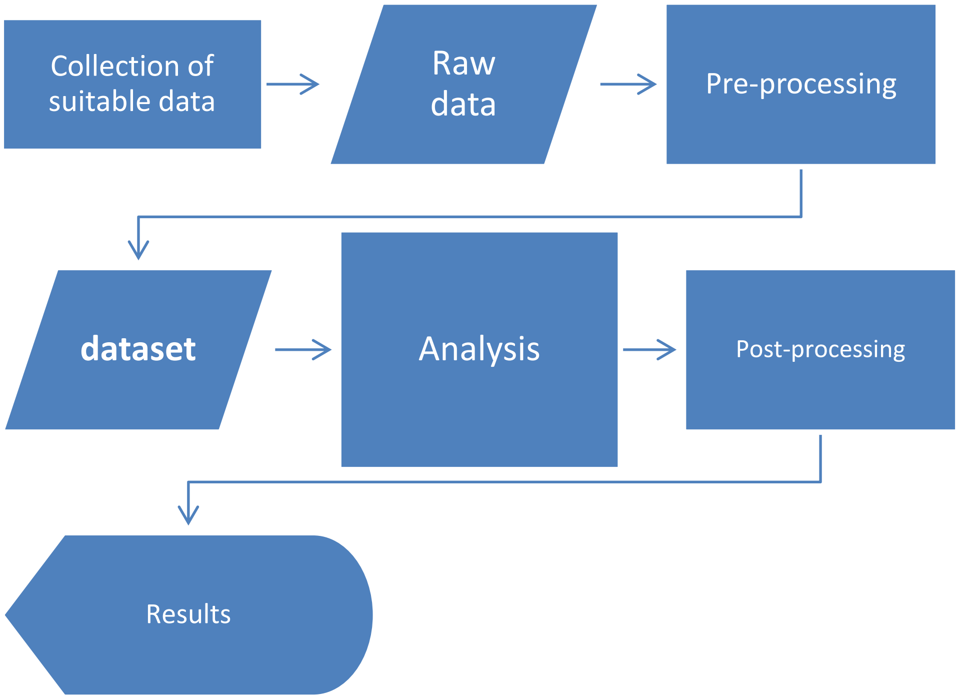
1.1 *Subject of thesis*

The main purpose of the thesis is the extraction of the most relevant review words for a certain technical attribute. The research is made in the context of a collection of photo cameras that we have retrieved from an online product catalogue.

The product reviews mostly contain a description of the product and text that describes the user experience with positive or negative comments. Since there is not a norm in writing reviews, usually the text is enriched by the users with “chatter”, in other words information not so much about the product itself but about the buyer’s experience. Also, from all the words used, some may appear frequently because of their extensive use in language regardless of context. The hard part is highlighting those words that indicate the importance of a feature even in natural language. Therefore, we need the *frequencies* of the words but also a way to distinguish those *important* in the text and the collection in general. Moreover, we must find a way to show that the importance of a word is *crucial* for the appearance of a technical characteristic in the list of specifications.

In total, a system is built to collect the data directly from the catalogue and after the application of text cleaning and term normalization techniques, it analyzes the data and extracts lists of the most important words for a set of technical characteristics. We will see how this system is constracted and the results it produces and we will discuss these results.

Figure 1: A schematic representation of the system



A schematic representation of the system is shown in Figure 1. The suggested system will consist of the following pieces:

- **Collecting the data**:
  At this stage we collect the data. For our purpose we will use the APIs provided by the online stores and we will collect suitable data. The data are given as input to the stage of pre-processing.
- **Pre-processing**:
  From the dataset's reviews we clean the text from punctuation marks, special characters, we single out the words, apply stemming on them and remove stopwords. We compute the basic statistics for our dataset. We remove the rarest words by setting a threshold on frequency of appearance. We compute the t*f-idf* scores for the words in reviews and also the raw frequencies of the words. Technical product attributes are separated based on their type (boolean, numerical, categorical, mixed) and we keep those we believe are likely to give us good results. Numerical and categorical features are converted to boolean. The dataset is ready for analysis.
- **Analysis**:
  We analyze the now normalized data and compute for them the frequencies of

the values of the technical attributes we have chosen and the frequencies of the words to create a Frequency Matrix. We apply the Chi-Squared Test for Independence to words and attribute values. The results are given to post-processing.

- **Post-processing and evaluation:**
  From the Analysis stage we keep the words that have stood out for each technical characteristic, dismissing those under a certain threshold that we determine. At this stage a review of the results will be done and we will evaluate if our method is good for the problem we try to solve.

1.2 *How the volume is organized*

On Chapter 2, we will view the problem more formally and describe some already applied techniques used in relevant research papers. The 3rd Chapter describes how the collection of data is performed and some statistics regarding the data. On Chapter 4, we describe the pre-processing and on Chapter 5 the analysis of the dataset. On the 6th Chapter we can see the results of our experiments and their evaluation and also some comments on them. Finally, in the epilogue, we discuss about further processing that could give better results and we mention some possible future extensions.

2

Terminology and relevant studies

Online user reviews have been a subject of interest for researchers because of the rich and multilevel information they can provide. For example, they have been used for opinion extraction in [3], summarization based on product features in [4] and [5], summarization generally, etc. The present work focuses on the relationship between the technical characteristics of products and the words from their reviews and draws its ideas from similar approaches that either find an indirect relationship between the two [2], with the technical attributes themselves not being visible to the user, only the results presented as a ranking of products, or a more direct approach [1], with the words used by the user on an e-commerce search engine being restated as technical characteristics.

Below follows a more formal definition for the problem and then the two most relevant researches are presented.

2.1 *Formal definition*

Our research is done in the context of a collection of photo cameras. Stated more formally, we have a collection of products $\boldsymbol{P}$ = $\boldsymbol{(p_1, p_2}$, $\boldsymbol{p_3}$, ...), with a number $|\boldsymbol{P}|$ = $\boldsymbol{N}$, for which we are provided also with their reviews $\boldsymbol{R_P}$= $\boldsymbol{(r_1, r_2, r_3,}$ ...). A set $\boldsymbol{A}$ = ($\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$, $\boldsymbol{\alpha_3}$, ...) contains all the technical attributes that could describe a product, like *"Optical Zoom", "Weight", "Aperture Range",* and a set $\boldsymbol{W_A}$= $\boldsymbol{(w_1, w_2, w_3,}$ ...) all the values that appear in the catalogue for a specific attribute, e.g. *"Weight"* can have values like *"3.2 ounces"*, *"1.5 pounds"*, etc. Finally, a set $\boldsymbol{V} = (\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, ...)$, with $\boldsymbol{v_i}$ depicting a word, is the users' vocabulary extracted from the reviews. As vocabulary we define the set of all different words being used by the users in this particular collection of reviews. Based on the above, every product $\boldsymbol{p_i} \in \boldsymbol{P}$ can be described as a set of attribute – value pairs, that is $\boldsymbol{p_i} = \{(\boldsymbol{\alpha_1}, \boldsymbol{w_1}), (\boldsymbol{\alpha_2}, \boldsymbol{w_2}), \dots, (\boldsymbol{\alpha_k}, \boldsymbol{w_k})\}$ and every review can be described with a set of lexicographical terms (words), as $\boldsymbol{r_1} = (\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, ..., \boldsymbol{v_n})$ like in the Bag-Of-Words model.

The goal is to fing a set of lexicographical terms $(\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{v_3}, \dots, \boldsymbol{v_j})$ for each technical characteristic $\boldsymbol{w_i}$ and to see if it is described "well" by the set of terms or else to understand what the users *say* about this characteristic in their reviews. From this we can see whether a certain characteristic can be represented with our method by lexicographical terms or if we can construct a profile regarding the user that is interested in the particular technical characteristic in their shopping, something that could be used in a recommendation system [2].

2.2 ***Relevant studies***

2.2.1 Learning to Question: Leveraging Preferences for Shopping Advice

In Mahashweta Das', Aristides Gionis', Gianmarco De Francisci Morales' and Ingmar Weber's work, an interactive recommendation system has been constructed [2] by implementing the *ShoppingAdvisor Tree*, a form of a Decision Tree that has questions – nodes about technical characteristics of products based on tags of the users on their own photos on *Flickr* or tags extracted from the reviews of car shoppers on *Yahoo! Cars*. This idea is drawn on the observation that shoppers "feel better supported when presented with qualitative product information rather than technical details" [2]. This qualitative product information is composed by the tags we mentioned.

In "*Learning to Question: Leveraging User Preferences For Shopping Advice*" the mapping of words with technical characteristics is made indirectly, as a Decision Tree with nodes the words extracted from reviews or tags decides a ranking of products at each node with the help of a *learning-to-rank* function that learns weights for product technical characteristics. The decision on which word will end up on which node is made with the help of the following function:

$$payoff(q, a) = combine(payoff(Uq(a)), payoff(Uq(a)), \\ |Uq(a)|, |Uq(a)|, |Uq|),$$

where a is an attribute provided by the user, one of the words that he uses, q the node and U_q the set of users that match it. The split of the set of users happens on each node so that payoff is maximized. The learning of the function learning-to-rank *rank(p)*: P -> R, takes place on each node with the SVM-Rank algorithm and the result is a product ranking that is evaluated with a known method so that the formula *payoff(U) = eval(rank)* is valid. P is the set of prodicts as a representation based on their characteristics and R the set of real numbers. We observe that the two components of the system are co-dependent and this fact determines the association between the technical characteristics of each product and the words that have been chosen as nodes on the tree.

2.2.2 Structured Query Reformulations in Commerce Search

To face the problem of mapping *free tokens* of a query on an e-commerce search engine to technical characteristics (attribute-value pairs) Sreenivas Gollapudi, Samuel Ieong και Anitha Kannan use a semantic parser [6] that splits the tokens of a query in *typed* and *free*. Instead of handling the free tokens as keywords like search engines did at the time to solve the problem, something that would possibly give inconsistent results because of searching exclusively from one source of information (e.g. a certain online catalogue), it draws information about them based on the user behaviour that make the query, using their browse trails (namely all the webpages they visit on each search session).

For each domain name that belongs to the dataset and each token, frequency counts are computed (i.e. how often a query that contains the particular token *t* ends up on a click on a page of domain *d*) and based on them distributions are computed for each domain (*d*) and free token (*f*), $P(f|d)$. These distributions are combined with the number of domains for which the free token does not have zero weight, resulting in a function imp(*f*) that shows how important the free token is considered as far as the set of results is concerned:

$$imp(f) = \sum_{d \in D} \mathbb{P}(f|d) \log\left(\frac{|D|}{1.0 + df(f)}\right)$$

where $|D|$ is the number of domains and $df(f)$the number of domains for which f has a non-zero weight. We observe that the function is a lot similar with the computation of the tf-idf score for a word in the standard problem of Text Mining. (In our work this exact tf-idf version is used at the beginning to distinguish the words that appear to have some importance for each camera.) The first 10 tokens with the largest imp(f), are considered candidate modifiers and the estimation of the association probability will be made for them with the attribute – value pairs.

The association probability is estimated with the formula:

$$\mathbb{P}((a,v)|m) = \frac{\mathbb{P}((a,v),m}{\sum_{m' \in M} \mathbb{P}((a,v) \in AV, m')}$$

where we find the probability a pair (a, v) being associated with the m. The combined probability P((a,v), m) is calculated by adding the products of the probabilities P(d)* P(a|d)* P(v|a, d)* P(m, d) for each d and is the intersection of the probabilities of selecting a certain domain P(d), and from this an attribute-value pair and a modifier to be chosen, facts independent from one another as the domain is chosen first.

2.3 ***Goal***

The goal is to take advantage of the frequencies of certain words and technical characteristics appearances' on the same photo cameras and to see if there is an association between them. The discovery of some strong associations will be an indication that we can represent the technical characteristics as a set of words or terms taken from the users' own comments. We believe that this can appear useful in searching a catalogue even for products that don't already have reviews, by assuming that as users describe a quality of a product in their reviews they will describe it in their queries while searching.

3

Collection of the data

The process of collecting the data is the basis for the research taking place here. Finding suitable data that will best fit our problem is a quite big part of the research.

Regarding our case, most commercial stores have adopted online evaluating of their products, while the technical characteristics are usually included in the description of the product. Also, some major scale webstores provide open access to these data through their API.

The analysis of the data for the discovery of associations between words and technical characteristics is done in the context of a collection of data that we have taken from *www.bestbuy.com*, an online store of various sorts of products. This collection is about the photo cameras found in the electronic catalogue and consists of 376 cameras, for each of which there is a list of technical characteristics and a list of reviews.

3.1 *Retrieving the data*

For the thesis' purposes all the photo cameras returned by Best Buy's API were collected. In total, 376 photo cameras' data were collected, without including camera accessories and product bundles.

Generally a Web API is an interface that provides us with a set of functions for the exchange of information between an application and the provider company, often through the use of HTTP request messages that return response messages in a certain format like e.g. Extensible Markup Language (XML) or JavaScript Object Notation (JSON).

Best Buy’s Web API offers this capability and provides a set of different APIs, for example:

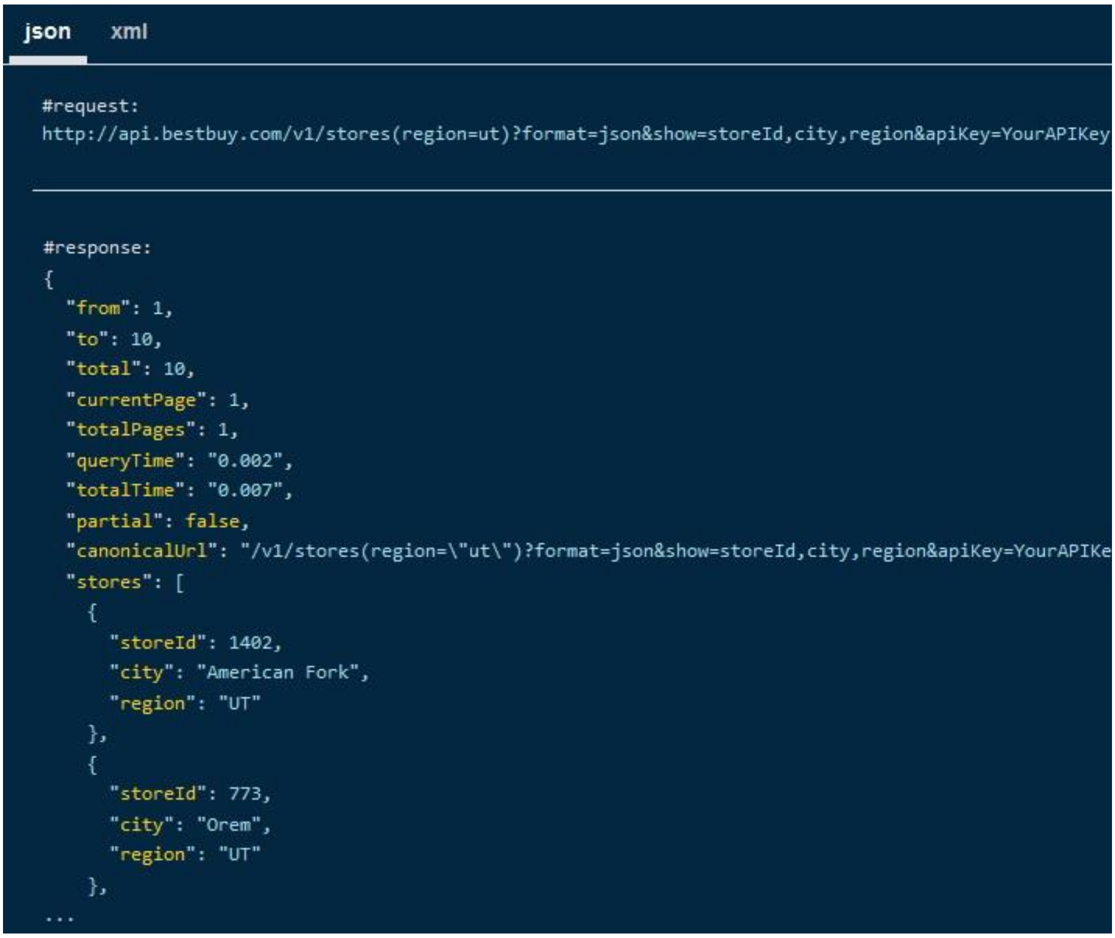
- Buying Options API,
- Categories API,
- Products API,
- Recommendations API,
- Reviews API and
- Smart Lists API

from which we can make queries through a Web Browser and receive formatted answers (responses) in an XML or JSON format.

For the needs of this project only Products API and Reviews API were used.

3.1.1 Best Buy API

Collecting data from Best Buy’s catalogue is made possible by using a standardized HTTP GET request on the respective Best Buy API. The user’s query included in the request consists of operators, values and attributes that are combined with conjunctions **&** or disjunctions **|** and are all defined in the API’s documentation. The operators include =, !=, >, <, >=, <=, in (for lists). A simple example in Best Buy’s documentation site is the query to find stores in the region of Utah. The query is made with the help of the “region” attribute: `http://api.bestbuy.com/v1/stores(region=ut)?format=json&show=storeId, city,region&apiKey=YourAPIKey.` In the following image a request and its matching API response is showed.



```
#request:
http://api.bestbuy.com/v1/stores(region=ut)?format=json&show=storeId,city,region&apiKey=YourAPIKey

#response:
{
  "from": 1,
  "to": 10,
  "total": 10,
  "currentPage": 1,
  "totalPages": 1,
  "queryTime": "0.002",
  "totalTime": "0.007",
  "partial": false,
  "canonicalUrl": "/v1/stores(region=\"ut\")?format=json&show=storeId,city,region&apiKey=YourAPIKe
  "stores": [
    {
      "storeId": 1402,
      "city": "American Fork",
      "region": "UT"
    },
    {
      "storeId": 773,
      "city": "Orem",
      "region": "UT"
    },
...
```

Image 1: API request - response example

As we can see, the request defines the attribute, the value, the formatting of the response message, the features we want to appear in it and the user's key required to make the request. The returned data are shown in the JSON format while there is also the option of formatting in XML.

The results are returned by tens and there is the option of increasing the returned number of results per page to 100 by using the parameter *pageSize*. The results can be thousand and for this reason the API provides us with Pagination with meta-data about the page and the parameter *page* for changing pages.

3.1.2 JSON Format

JSON Format is an open standard data format that is used for the transmission of data consisting of attribute-value pairs, in human-readable text form. It is independent of the programming language and there are already implementations for reading and creating

JSON data in many languages. An example of a possible representation of a person's information in JSON form is shown below[3]:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

In fact, Python has a json module integrated in its standard installation, for coding and decoding JSON objects to its corresponding objects. The mapping of types is as follows:

- JSON object: Python dict
- JSON array: Python list
- JSON string: Python Unicode
- JSON number (int): Python int, long
- JSON number (real): Python float
- JSON true: Python True
- JSON false: Python False
- JSON null: Python None

The data returned from an API request consist of a JSON table that contains JSON objects, each divided by comma, and every object is an unsorted collection of name/value where the names (or "keys") are alphanumeric values. Like *dictionaries,* they are structures that provide a mapping between name and value, so usually the

[3] https://en.wikipedia.org/wiki/JSON#JSON_sample

names contained are unique to the structure. Also, each name/value pair is separated from the next with a comma.

3.1.3 Products API

Products API contains information like costing, availability, technical specifications, images and other information for thousands of products. The lists of attributes it provides are distinguished by the type of information they belong to. So, we have attributes that concern the general description of the the products, the costing and sales, images, categorizations, offers and more. For the needs of the project the following attributes were used:

- *sku*: The product’s unique identifier in the catalogue
- *name*: The name of the product as is visible on the webpage
- *customerReviewCount*: The number of reviews for the particular product
- *details.name*: The product’s technical characteristics
- *details.value*: The values of the product’s technical characteristics
- *categoryPath.name*: The number of categories the product belongs to, from a set of hierarchically structured categories
- *weight*: Product’s weight
- *width:* Product’s width
- *height*: Product’s height
- *depth:* Product’s depth

We are interested mostly in the features of the products. The rest will be used for checks during the processing of the data. In the returned request’s result, every product is a JSON object and every pair of attribute/value is contained in the corresponding JSON pair of name/value. For example:

```
{
       "sku": 8334575,
       "name": "Sony - DSCWX220 18.2-Megapixel Digital Camera - 
Black",
       "customerReviewCount": 188,
       "details": [
         {
           "name": "Memory Card Included",
           "value": "No"
         },
         {
           "name": "Integrated Flash",
           "value": "Yes"
         },
         {
           "name": "Low Light/High Sensitivity",
           "value": "Yes"
         },
       …

 }
```

3.1.4 Reviews API

The reviews users submit on Best Buy are available through the Reviews API and are accompanied by information such as date of registration, the user's identity, the evaluated product's unique identifier, the average rating of the users and the comment of the user itself. In the returned answer of the request for a product, a list of reviews is contained with each review being a JSON object and each pair of attribute/value matching the corresponding JSON name/value pair, as shown below e.g. for the product with sku 8334557:

```
[{
      "sku": 8334557,
      "rating": 5.0,
      "comment": "Easy to use and takes good pictures. I really like
the wifi option so I dont have to fiddle with the memory card."
    },
    {
      "sku": 8334557,
      "rating": 5.0,
      "comment": "Easy to use and good quality pics. Nice design
easily fits in my purse."
    },
….

}]
```

3.1.5 Deciding the query on the API

Singling out the photo cameras from all the products in Best Buy was made with the help of attributes ***categoryPath.name*** and ***categoryPath.id*** from Products API. To confirm that this ensures the desirable results we have tried several queries on the API:

QUERY	**RESULTS NUMBER**
categoryPath.name **=** Digital Cameras	496
categoryPath.name **!=** Digital Cameras	736734

categoryPath.name* != (empty field)	133
categoryPath.name* !=* AND *categoryPath.id* !=	133
search = camera **(αναζήτηση σε όλα τα γνωρίσματα) AND *categoryPath.name* !=***	4
All Products	737230

Table 1: API queries, no 1

The query for the products that have the word “camera” in any of their attributes but their *categoryPath.name* field is empty gave 4 results out of which not one is a photo camera. So, there is not a photo camera in the API with the field *categoryPath.name* empty. The check for an empty field is safe because there are as many products with empty *categoryPath.name* field as with field *categoryPath.id***.**

Next step was the exclusion of all products relevant to photo cameras, like accessories and offer bundles. Except the field *categoryPath.name* there are more fields in the Products API relevant to categorization or grouping of products, like ***class***, ***department***, ***subclass*** in Categorizations category and also ***type,*** in Listing Products category, that contains the type of the product. Below some tests that were made to find the right query are shown:

QUERY	**RESULTS NUMBER**
class = DIGITAL CAMERA ACCY	3001
class = DIGITAL CAMERA ACCY **AND categoryPath.name =** Digital Cameras	69
class **=** DIGITAL CAMERA ACCY **AND *categoryPath.name*** **=** Cameras&Camcorders	2809
class **!=** DIGITAL CAMERA ACCY **AND *categoryPath.name*** **=** Digital Cameras	427
type **=** bundle **AND *categoryPath.name*** **=** Digital Cameras	51

***type* !=** bundle **AND categoryPath.name =** Digital Cameras	445
categoryPath.name = Digital Camera Accessories	2300
class != DIGITAL CAMERA ACCY **AND** **categoryPath.name** = Digital Camera Accessories	410
class = DIGITAL CAMERA ACCY **AND** **categoryPath.name** = Digital Cameras **AND** **type !=** bundle	69

Table 2: API queries, no 2

At last, with the query ***type*** != bundle and ***class*** != DIGITAL CAMERA ACCY, together with ***categoryPath.name*** = Digital Cameras, all accessories and bundles are excluded and only photo cameras are left.

The results deviate a little from the actual numbers we see on the webpage, as far as the number of reviews or products is concerned. This is caused by inconsistencies on the API's data and does not have an impact on the results of the experiments.

The retrieval of the data from the Best Buy's Web API was done with the help of the *python-bestbuy-client*[4] by transferring the suitable queries into the program's code.

3.2 ***Data statistics***

In brief, the basic statistics for our collection's data are presented below:

DATA	**NUMBER**
Photo Cameras	376
Technical characteristics	178
Number of reviews	22330
Number of reviews after merging	376
Largest number of reviews per camera	1519

[4] https://factory84.com/blog/bestbuycom-python-client-library/

Average number of reviews per camera	60
Total number of words	875665
Vocabulary size (normalized)	13886
Average number of words per review	40
Average number of words per camera	2328
Total number of words after removing those with DF(w) < 10	96625
Vocabulary size after removing those with DF(w) < 10	3154
Number of stopwords removed	242

Table 3: Collection's statistics

On the next page, in Image 2, we can see the distribution of reviews with respect to the number of cameras on a log - log plot histogram [5] while in Image 3 the distribution of words with respect to the reviews' number is shown[6].

[5] https://plot.ly/~des.dim/29/log-log-plot1/

[6] https://plot.ly/~des.dim/31/log-log-plot2/

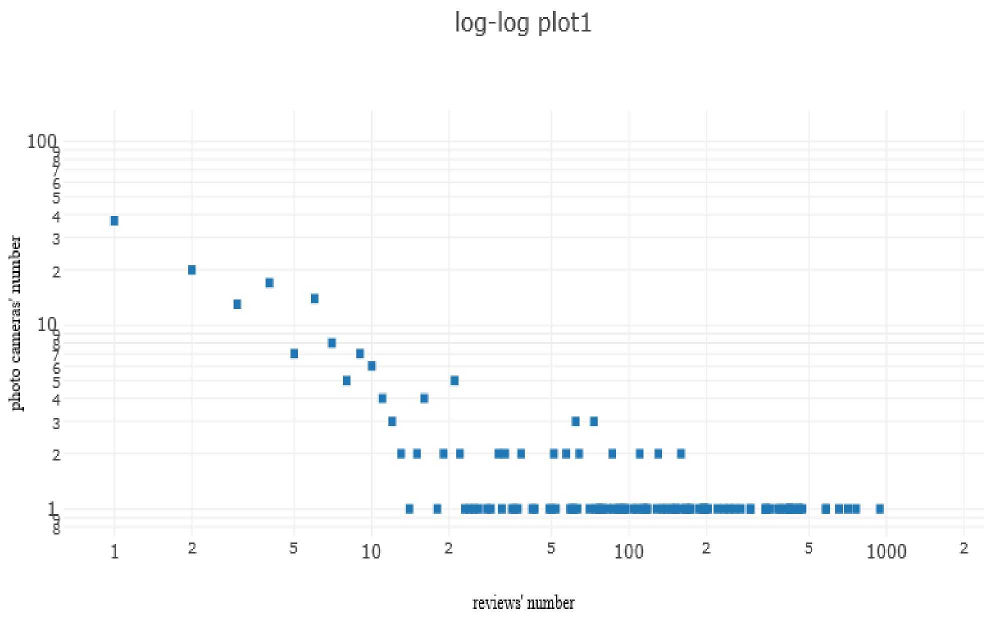


Image 2: Reviews vs Photo Cameras

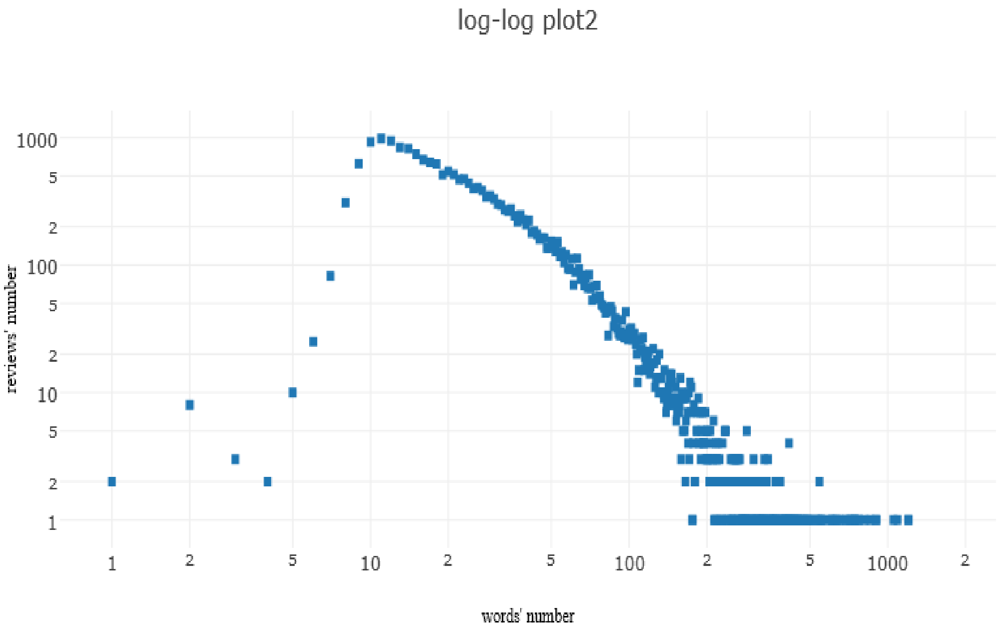


Image 3: Words vs Reviews

4

Preprocessing of the data

The preprocessing of the data refers to the lexical analysis of the reviews and the application of techniques for cleaning the data from words we do not consider impotant as well as the selection and analysis of the technical characteristics.

Our data is in JSON format. As we saw previously, the details for the products the API returns contain a list of technical characteristics but also some other details, such as weight, which is a common feature for all kinds of products. So, at a first level, we group all the characteristics we assume should be together. In particular, fields *weight, width, length, height* are inserted in the list of technical characteristics so that we can analyze them all together and the initial separate fields are deleted. The second change on the structure is the insertion of a new field *reviews* in the products to fill with the total of reviews of each product. As mentioned in chapter 3, reviews are retrieved with a separate call on the API, so a call for each product is needed to retrieve the reviews for each one. The final change on the dataset's collection is the merging of all reviews into a single text so that it is later considered as one document to compute the *tf-idf scores*.

4.1 *Reviews*

To be able to count raw frequencies and compare the words with each other, they have to be comparable. This means that our system should be able to recognize the same words, something that is hard for data that have not been preprocessed. This problem arises from the difference in writing manners of the users, for example an expression could be altered by giving emphasis or by writing the words as they sound. This means that spelling mistakes or deliberate orthographic alterations may appear. Moreover, the text must be cleaned from punctuation marks and be split into separate words. These can be corrected during the lexical and syntactic analysis of the text, processes that belong to the field of Natural Language Processing.

On second level, while the reviews are now sets of lexicographical terms, we have to take into account one more parameter: Even if we can get the common appearances of lexicographical terms and their frequencies, we do not forget that, as far as reviews are concerned, we are dealing with text where words that appear too often do not hold much of importance, like articles and linguistic links, as well as that in a collection of documents some words stand out more than others, something that depends greatly on the specific subject of the collection in total, so some method is needed for them to be extracted. For this purpose, we can use the *tf-idf score* but also the removal of some really rare words based on a threshold we ourselves set.

4.1.1 Lexical and Linguistic Processing

For the process of cleaning the reviews' data we use *cleandata.py* which implements the following:

- Removes all ***punctuation marks*** except hyphen because it is often used as part of a word-phrase.
- The text breaks into words with the help of *textblob*, a Python tool utilizing use with the basic methods for natural language processing (NLP), only simplified.
- Removes ***stopwords***. Stopwords are called the words that have a high possibility of appearance in the language but do not alter the text's general subject. There are ready made files that contain stopwords for every language and are available on the internet. We use the stopwords available for the english language that is part of the NLTK corpus, a module that contains various functions to do NLP. In total the stopwords we remove from the text are 242.
- The words are replaced by their ***roots*** where possible, based on the part of speech the word belongs to. This is done with the method word.lemmatize() found in textblob which at the same time normalizes the words that appear in singular and plural in different points of the text.

4.1.2 Removing rarest words

For the frequencies of appearance for the words we define a threshold equal to 10, meaning words that appear in less than 10 reviews do not participate in the analysis. This is about 3% of the reviews.

4.1.3 Tf-idf score calculation

In our collection, after an initial processing, we have a document $\boldsymbol{R_i}$ for each photo camera that consists in reality of the union of all reviews for this camera.

This modification is made because the method we're using to extract words from the text is to find the *tf-idf* scores of the words. This method hightlights the most important as well as frequent words *per review* based on the frequency of their appearance in text and the number of documents they appear into. So, because we want to highlight the most important words per camera, the merging of reviews is necessary.

The *tf-idf* score of term *t* regarding document *d* of collection *D* is computed by $tfidf(t,d,D) = tf(t,d) * idf(t,D)$, where:

- $tf(t,d) = \frac{f(t,d)}{|d|}$, where f$(t,d)$ is the sum of all appearances of a word in a document and |*d*| the number of words of a document, that gives us the relative frequency in document *d* of term *t*, and
- $idf(t,D)$ = $\log(\frac{N}{1+df(t)})$, where *N* is the total number of documents and $df(t)$ the total number of documents containing term *t* at least once.

This way, we "weigh" the words so that those most important (larger *tf-idf* score) are those that are not trivial (*idf*), and are important for a document as they appear in it often (*tf*).

This analysis satisfies well the problem of highlighting words per review in our collection; since all users comment on one kind of products, words that are trivial for the sort, like *camera, photo* etc, will not have a high score, while words that are used often when users refer to a photo camera will increase its *tf-idf* value.

For the present project, we will compute the *tf-idf* for each photo camera and review. Next, we will sort in descending order the words based on the *tf-idf* score and we will keep for analyzing only the first 50. So, experiments in Chapter 5 will be done on those words that have been highlighted in every document as most important.

4.2 ***Technical Characteristics***

A complex matter is also the processing of the technical characteristics. One technical characteristic has a set of possible values that it can take from a set of values we extract from the catalogue. The most basic issues that we have to address when dealing with a list of technical characteristics are:

1. Different value types; our collection contains numeric data, categorical data, boolean data (in *Yes/No* format) and data of mixed type with combinations from the above. In reality, all values are alphanumeric but we can apply a simple preprocessing to make them belong to one of these types accordingly.
2. Multiple types and how to split them; for example, in our collection, some characteristics' values contain as separators **;**, **|**, and **,**. To be able to compare the values with each other we have to turn them into a list with single values.
3. Usage of different types and form of expression; for example, *"40 ounces"* which is the same with *"2.8 pounds"* but also *"ISO 80-1600"* which is almost the same with *"ISO 80/100/200/400/1600"*, or *"230K"* instead of 230,000 and *"Wide-angle"*, *"Wide-angle lens"*, both of them values of *"Lens Features"* .
4. Symbols and special characters included in the string, that do not provide any useful information, and measurement units.
5. Values that are too close to each other, like *"Image Resolution": "4608 x 3072"*, *"4608 x 3440"*, *"4608 x 3456"* and need to be grouped in size classes that are meaningful for the specific technical characteristic, and
6. Values that resemble descriptions as well as categorical values that are many in number and can not be grouped so easily.

In this work, we will not go into so much detail so as to clean the data for all the above cases but we will deal with all that seem to hold some interest or are likely to give good results.

At first, *we split* the characteristics based on their type (boolean, arithmetic, categorical, mixed) after we have first split the multiple values and created a list with the singles. We use the boolean characteristics that are easier to handle and create two size classes for some of the arithmetic. The automated splitting of the values is beyond the purposes of this work. Specifically, the arithmetic technical features we select and the corresponding classes we create are:

- *"Total Megapixels": "Medium", "Large"*

  For *Total Megapixels* we put in a class-list its values from 5.0 MP to 17.2 MP and assign them the class *"Medium"*. The rest of the values begin from 18.3 MP and end at 53.0 MP and are placed in class "*Large"*. The borderline between *"Medium"* Megapixels and "*Large"* Megapixels is small but in this work we are not so interested in the most semantically precise meaning as in that, with the splitting, we will infer a possible association with words.

- *"Effective Megapixels": "Medium", "Large"*

  Into the same classes with *Total Megapixels*, we split also *Effective Megapixels* for *"Medium"* beginning from 5.0 MP and ending at 16.4 MP and for "*Large"* beginning from 18.0 MP and ending at 50.6 MP.

- *"Weight": "Light", "Heavy"*

  Weight is split into *"Light"* and *"Heavy"* and because of this the approach is more realistic, considering as lightweight all cameras under 11 ounces, meaning approximately 300 grams, and the rest as heavy.

- *"Optical Zoom": "Small", "Large"*

  Optical Zoom is considered the “significant” zoom in digital cameras and we split it into *Small* for values ranging from 1x to 8x and *Large* from 10x to 65x.

In Tables 4 and 5 are shown some statistics for the technical characteristics that we are going to use in our analysis and whose potential significance we will discuss in the results.

Technical Characteristic	"Yes"	"No"	Photo Cameras
Water Resistant	35	323	358
Shock Resistant	37	184	221
Cold Resistant	43	180	223
Integrated GPS	65	258	347
Camera Full Frame Sensor	32	118	150
Varying Angle Screen	43	26	69
Burst Mode	267	31	298
Touch Screen	90	191	281
Long Zoom	51	43	94
Instant Print	38	320	358

Table 4: Statistics for technical characteristics and their values, for boolean features

Technical Characteristic	"Small"	"Large"	Photo Cameras
Total Megapixels	149	152	303
Effective Megapixels	168	160	328
Optical Zoom	124	79	215
Weight	169	204	373

Table 5: Statistics for technical characteristics and their values, for categorical features

5

Analysis of the data

5.1 *Discovering associations*

After we bring the lexicographical terms and the technical characteristics in the form we want, we are ready to explore the data to discover associations. At this stage, we need a method that shows that certain lexicographical terms are determinant for a photo camera having some technical characteristic.

Finding which and how many words and features' values appear at the same time is easy. The real difficulty lies in understanding when, together with a word used by the users, there is a noticeable change on the characteristics' value. For example, to see that the word *"underwater"* appears more often when the feature *"Water Resistant"* has a *"Yes"* value but the opposite also, that the word does not appear often when *"Water Resistant"* has a *"No"* value. We also need a method that will ensure that the associations we discover are statistically significant. A method that does exactly that the Chi - Squared Test for Independence that checks if differences in observations of a crowd's categorical characteristics are random or not.

5.1.1 Chi - Squared Test for Independence[7]

Chi - Squared Test is applied when we have two categorical variables from a population to ascertain if there is a significant association between them. For its application certain criteria must be fulfilled:

[7] Από: http://stattrek.com/chi-square-test/independence.aspx?Tutorial=AP

- The sampling method is simple random sampling.
- The variables that describe the population are each categorical.
- The contents of a *contingency table* used should have value at least 5 in each cell.

The Chi - Squared Test is used as described below:

- We state an initial hypothesis for the relationship which we want our data to satisfy.
- We set the *Significance Level* to some suitable value. Usually one of 0.10, 0.05 and 0.01 is used.
- We analyze the contingency table's data by computing: *Degrees of Freedom*, *Expected Frequencies*, *Test Statistic* and *p-value*
- We interpret the results by comparing with the *Significance Level's* value.

If variable A has r different values and variable B has c different values, then:

- **Degrees of freedom:**

$$DF = (r - 1) * (c - 1)$$

- **Expected frequencies**:

$$E_{r,c} = (n_r * n_c) / n$$

- **Test statistic:**

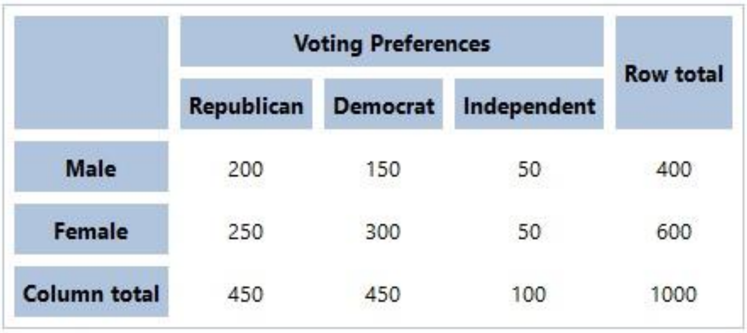
$$X^2 = \sum [(O_{r,c} - E_{r,c})^2 / E_{r,c}]$$

- **P-value (Cumulative Probability):** $P(X^2 < CV)$

where:

$E_{r,c}$: The expected frequency of appearance for cell *r*, *c*. $O_{r,c}$: Raw frequency of appearance for cell *r*, *c* and *CV*: Critical Value, meaning the Test statistic.

A simple example is to find if there is a significant relationship between gender and voting preference in a population with men and women. Image 4 shows the *contingency table* that results from this example.



	Voting Preferences			Row total
	Republican	Democrat	Independent	
Male	200	150	50	400
Female	250	300	50	600
Column total	450	450	100	1000

Image 4: Example of a contingency table by http://stattrek.com

The initial hypothesis that gender and voting preference are independent with each other is proven wrong with the application of the Chi - Squared Test for Independence because the value of the Test Statistic is 16.2 but the p-value = 0.0003 is smaller than the selected for this experiment *significance level*, which is 0.05.

In our case, instead of gender we have chosen words from reviews, meaning words from the vocabulary of our collection and instead of voted party the appearance of a technical feature’s value.

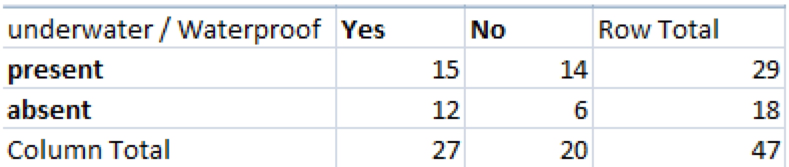
5.1.2 *Frequencies table*

The Chi - Squared Test is a way to find if there is an association between the appearance of a word in a camera’s reviews and each technical characteristic’s value. Our variables are respectively: Lexicographical Term, taken from a review, and Technical Characteristic of a photo camera. In contrast with example in Image 4, we do not choose two different terms but one, that either appears in the reviews or not, and the various values for the selected technical characteristic. Because our problem is to find the words that are *defining* the appearance of some technical characteristic we approach the problem by taking into account not only how often a term and a feature’s value appear together but also how often this term is absent from the reviews of a photo camera with this particular feature value. Of course, this way is not absolute because every user has their own way of expression and different knowledge background as far as their buy is concerned, so they choose from a different personal vocabulary, but it is a good measure for us to check which words are promising in our analysis.

Therefore, there is need for a table which, like the contingency table, gathers the raw frequencies of the words and technical characteristics. We create a frequencies table that contains:

- the number O_{wf} of photo cameras in whose reviews a lexicographical term appears while at the same time a specific feature value appears-does no appear and
- the number of photo cameras O_{wf} in whose reviews the term does *not* appear while at the same time a specific feature value appears/does not appear.

Such a frequencies table is shown in Image 5.



underwater / Waterproof	**Yes**	**No**	Row Total
present	15	14	29
absent	12	6	18
Column Total	27	20	47

Image 5: Frequencies Table

We choose the frequencies table to apply the Chi - Squared Test for Independence on.

In this way, we define an association so that a technical characteristic is examined so much for its relationship with a word as for its relationship with the word’s absence because the frequent use of a word may not be caused only by this feature’s appearance, as there is a whole list of other features co-appearing. Because the dataset we have is relatively small for our problem, we set the *significance level* = 0.10 – which is a relatively high boundary – and in the final results we keep those words with p-value smaller than this. In other words, each characteristic is finally associated with a set of words for which the Chi - Squared Test gives a p-value smaller than 0.1.

6

Experiment Results

6.1 *Data used in experiments*

From the set of all technical characteristics, we keep the following boolean and arithmetic ones that have already been grouped: *Shock Resistant, Cold Resistant, Water Resistant, Optical Zoom, Weight, Total Megapixels, Effective Megapixels, Burst Mode, Camera Full Frame Sensor, Instant Print* and *Touch Screen*.

The lexicographical terms for which we apply the Chi - Square Test are the terms that have been sorted out after the pre-processing of the reviews. This means that the lexicographical terms with absolute frequency of appearance 10 in mutually different reviews, are not taken into account for the experiment with the Chi – Square Test. Also, after the application of the *tf-idf* method on the lexicographical terms for each document-review and the sorting of terms with largest *tf-idf scores* as most important, we keep for the experiment the first 50 by each review.

After the application of the Chi - Square Test on these terms and the technical characteristics we have selected, and the computation of the Test Statistic and the p-value, for each technical characteristic we keep those lexicographical terms that have p-value < 0.10 and we sort the results based on the p-value in an ascending order. From these and for each feature we keep the first 24 and examine them with respect to their correspond feature, as follows.

6.2 *Results and Evaluation*

6.2.1 Results

The terms we consider good results are highlihted in bold writing.

Water Resistant

Lexicographical term	**p-value**	**Lexicographical term**	**p-value**
underwater	2.11268795179e-35	spend	0.00229020513159
water	3.39479284061e-33	**tough**	0.0079906147858
wet	5.26218098664e-14	fairly	0.0079906147858
proof	6.04353641776e-10	appear	0.0184891399319
worry	1.34606933746e-08	**survive**	0.0184891399319
drop	2.22360024583e-08	**park**	0.0184891399319
hawaii	5.79640012069e-08	**sturdy**	0.0184891399319
vacation	8.3672057707e-07	**activity**	0.0184891399319
trip	7.06216176392e-05	mexico	0.0184891399319
didn	0.000295976049191	companion	0.0184891399319
cruise	0.000356384145382	ok	0.0184891399319
seal	0.00229020513159	**repair**	0.0184891399319

Table 6: Words sorted based on the p-value for feature Water Resistant

Varying Angle Screen

Lexicographical term	p-value
nice	0.00844773228618
battery	0.00844773228618
buy	0.0182524690242
picture	0.0248102476602
use	0.0276566646016
video	0.0463715801493
get	0.0527895376507
screen	0.0726591524917
lens	0.0726591524917
really	0.0983854412796

Table 7: : Words sorted based on the p-value for feature Varying Angle Screen

Total Megapixels

Lexicographical term	p-value	Lexicographical term	p-value
great	7.03364238164e-10	**purchase**	8.26024404017e-05
upgrade	3.62425669711e-07	picture	9.4419322801e-05
first	3.62425669711e-07	make	9.89141055659e-05
use	1.72783428027e-06	best	0.000109971970071
lens	1.88630790705e-06	good	0.000149103433302
love	2.99775429784e-06	dslr	0.000189293062467
easy	8.56554456142e-06	can	0.000240871089845
take	8.67876741731e-06	buy	0.000323062776806
quality	9.15789617852e-06	light	0.000382151398784
frame	1.78795017361e-05	new	0.000468336585692
photography	2.11044701461e-05	feature	0.000610249263109
learn	2.39721025173e-05	video	0.000628336649241

Table 8: : Words sorted based on the p-value for feature Total Megapixels

Optical Zoom

Lexicographical term	p-value	Lexicographical term	p-value
zoom	6.03419751501e-16	**point**	0.00995744822039
first	0.000205237259279	purse	0.011140646752
lens	0.000377342412412	will	0.0122538162373
clear	0.00137513847419	close	0.0163848361352
optical	0.00217370770876	nice	0.024084319684
long	0.00319581209904	professional	0.0248958273935
photography	0.00357896662695	wi-fi	0.0285255605074
t	0.00537257078539	aa	0.0285255605074
battery	0.00897818770571	need	0.0491537152879
dslr	0.00945693889702	image	0.0542798013874

Table 9: : Words sorted based on the p-value for feature Optical Zoom

Long Zoom

Lexicographical term	p-value	Lexicographical term	p-value
zoom	1.83080888009e-10	picture	0.0154041731357
point	0.00132915445259	battery	0.0162512488909
price	0.00209693551561	want	0.020344488727
easy	0.00288449068576	need	0.020344488727
like	0.00300032565627	still	0.0214263238203
clear	0.00426797601363	look	0.0256387054178
camera	0.00793501056716	work	0.0256387054178
feature	0.00901427480127	take	0.0308689790529
shot	0.00928602404629	one	0.0326530828103
just	0.0111444886883	photo	0.0343528551431

Table 10: : Words sorted based on the p-value for feature Long Zoom

Burst Mode

Lexicographical term	p-value	Lexicographical term	p-value
connect	1.07178545576e-05	easy	0.00689782489086
device	1.07178545576e-05	facebook	0.011275032978
automatically	0.000590259977388	email	0.011275032978
lens	0.00145354750052	need	0.0114880083893
buy	0.00178918655344	good	0.0126212173301
great	0.00343904226033	love	0.016890350041
android	0.0034461293249	want	0.0204399686638
tablet	0.0034461293249	alone	0.0239523557993
can	0.00357442055442	social	0.0239523557993
use	0.00448030190218	printing	0.0239523557993

Table 11: : Words sorted based on the p-value for feature Burst Mode

Camera Full Frame Sensor

Lexicographical term	p-value	Lexicographical term	p-value
full	3.12648906505e-09	iii	0.000812647306051
frame	4.63238167971e-09	zeiss	0.00105637742444
d800	3.63608982612e-07	d610	0.00105637742444
image	1.13593646127e-06	a7	0.00105637742444
iso	2.625345973e-06	photo	0.00122018470898
noise	2.12091341052e-05	**light**	0.00188255453997
sensor	8.49603339326e-05	**high**	0.00198596710655
full-frame	0.000121166564643	5d	0.00198596710655
body	0.000429683091169	best	0.0021520505793
6d	0.000484744465022	adapter	0.00266012295852

Table 12: : Words sorted based on the p-value for feature Camera Full Frame Sensor

Shock Resistant

Lexicographical term	**p-value**	**Lexicographical term**	**p-value**
vacation	7.08042931349e-06	button	0.0143054408288
worry	5.98718047096e-05	yet	0.0276399127763
trip	0.000464519896478	**work**	0.0332042224476
drop	0.000912055469041	get	0.0371367089921
last	0.00125349100268	raw	0.0371482832523
time	0.0041343613312	dslr	0.0374277327871
coolpix	0.00636193714021	**tough**	0.043925664108
seal	0.0133726202095	fairly	0.043925664108
kid	0.0133726202095	lens	0.0439620041232
spend	0.0133726202095	use	0.0472634030581

Table 13: **:** Words sorted based on the p-value for feature Shock Resistant

Touch Screen

Lexicographical term	**p-value**	**Lexicographical term**	**p-value**
touch	3.71114450972e-07	time	0.00257340037758
rebel	5.09002452916e-06	t5i	0.00302058628127
need	0.000245678286904	can	0.00326678962015
work	0.000626320347009	good	0.00346846031379
zoom	0.00109943077694	price	0.00364852045687
battery	0.0014799988643	one	0.00560392675238
nice	0.00175620826008	carry	0.00659965227936
screen	0.00229740956398	clear	0.00715771868596
pocket	0.0024315472116	use	0.00721018782022
t3i	0.00246835093927	look	0.00725682116376

Table 14: **:** Words sorted based on the p-value for feature Touch Screen

Cold Resistant

Lexicographical term	p-value	Lexicographical term	p-value
vacation	0.000303993729522	button	0.0363382116612
worry	0.000712153784054	didn	0.0392541609445
trip	0.00116652222739	dslr	0.0485875444204
coolpix	0.00276886946694	clear	0.0521061217794
time	0.00339622581443	**work**	0.0600931942483
drop	0.00364473951747	lens	0.0622270223366
last	0.0270479884026	yet	0.0622270223366
seal	0.0270479884026	wife	0.0631353283932
kid	0.0270479884026	photography	0.072438823342
spend	0.0349240498056	size	0.0740174095708

Table 15: : Words sorted based on the p-value for feature Cold Resistant

Integrated GPS

Lexicographical term	p-value	Lexicographical term	p-value
gps	2.4697517945e-08	proof	0.00321506707633
water	1.87950207244e-06	**durable**	0.00321506707633
beach	7.69102501171e-05	raw	0.00321506707633
snorkel	0.000107510265195	see	0.00321506707633
waterproof	0.000190434943802	micro	0.00505500664831
system	0.000563355794165	image	0.00515631220889
underwater	0.000813594271657	**button**	0.00609661390014
mark	0.00103862102605	wedding	0.00985811173109
iso	0.00159595327946	price	0.0119332019873
pool	0.00272116867894	photographer	0.0161374630902

Table 16: : Words sorted based on the p-value for feature Integrated GPS

Weight

Lexicographical term	p-value	Lexicographical term	p-value
lens	2.34610448469e-10	sensor	0.000240982696914
learn	2.31346826563e-08	first	0.000347573297062
dslr	2.31346826563e-08	vacation	0.000368409037279
photography	1.4573371027e-06	iso	0.000385341177471
upgrade	2.85523505498e-06	**snorkel**	0.000386340843205
great	3.30030445544e-05	image	0.000660916620613
focus	6.07071595199e-05	**professional**	0.000719222696648
beginner	0.000146348195232	photographer	0.000719222696648
pocket	0.000187182310276	much	0.000765108953888
amazing	0.000218219955986	make	0.000765108953888

Table 17: : Words sorted based on the p-value for feature Weight

Effective Megapixels

Lexicographical term	p-value	Lexicographical term	p-value
great	4.23581641704e-09	light	3.89691829948e-05
upgrade	1.28084605348e-07	learn	4.21238411979e-05
first	4.20348955221e-07	frame	5.96734466304e-05
lens	5.17451608202e-07	good	7.72815341358e-05
love	1.91854479686e-06	picture	7.74555007582e-05
quality	3.57970812393e-06	video	9.4174531175e-05
take	6.3049119614e-06	purchase	0.000110829721941
easy	9.28508679952e-06	dslr	0.000292741312523
photography	1.37565714252e-05	feature	0.000310831253847
use	1.45875688617e-05	size	0.000325799407009

Table 18: : Words sorted based on the p-value for feature Effective Megapixels

Instant Print

Lexicographical term	**p-value**	**Lexicographical term**	**p-value**
fun	1.30739384936e-12	**party**	6.28703166718e-07
cute	2.02318634942e-11	**expensive**	6.32124270591e-07
rebel	8.41648656325e-11	**daughter**	1.09652629979e-06
film	5.34538468775e-10	7d	2.64914478807e-06
instantly	1.22323308365e-09	**kid**	2.64914478807e-06
retro	3.74589418909e-09	t5i	3.80466601974e-06
print	1.48698352526e-08	**develop**	1.41540133799e-05
60d	6.7029066615e-08	**gift**	8.28611570331e-05
credit	2.42285762697e-07	upgrade	0.000117423959203
bring	2.42285762697e-07	**remind**	0.000129746212376
birthday	6.28703166718e-07	**memory**	0.000151700661446

Table 19: : Words sorted based on the p-value for feature Instant Print

6.2.2 Evaluation of Results

The results shown in Table 4 concern the feature "***Water Resistant***". From the 24 words we recognize as relevant the 17. The terms *underwater, water, wet* that are listed first indicate exactly this feature's distinctive quality: It is used near water. More specifically, as *proof, drop, tough, seal, survive, sturdy, worry, repair* further indicate, the product is characterized by durability or is very likely to be exposed to dangers. Also, we see that the users talk in their reviews about *hawaii, vacation, trip, cruise, park, activity,* meaning that they use the product in travels and outside activities. Generally the results are what we would want to see as a good output in our system.

As for feature "***Varying Angle Screen"***, the results in Table 7 are a little disappointing but this may be explained by the fact that there is a small number of reviews for the correspondent cameras. (see Technical Characteristics' Statistics in Chapt. 4) The term we can say is the best in our 10 results returned for the feature Varying Angle Screen is *screen* while the other words are not so interesting.

In Table 8 which is about "***Total Megapixels"***, most words are somewhat trivial (*use, lens, take,frame, photography, picture, make, dslr, feature, video*) or just depict the user's opinion (*great, first, love, easy, best, good*). The rest *upgrade, quality, purchase, buy, new* are expected to be frequent in reviews regarding shopping. It would be unsafe to come to conclusions about the feature "***Total Megapixels"*** as it is a certain to appear feature in photo cameras and also the separating of its values was difficult.

Table 9 with the first terms for feature Optical Zoom, has words *zoom, optical* first on the list which is meaningful in the sense that these words are the name of the feature itself. The words *lens* and *point* could have some relationship with zoom while *clear, long, close* (e.g. long distance, long zoom, close distance) could be describing the feature. The rest in the list, *photography, battery dslr* are trivial, while the words *purse, t, wi-fi, will,* do not seem relevant or some other feature is the reason for their appearance maybe because of a strong correlation with the feature Optical Zoom.

In Table 10, there are the terms regarding "***Long Zoom"*** with not so good results. The list contains mostly trivial lexicographical terms like what we saw in feature Optical Zoom with only 3 good results in a good rank, and also similar, the terms τα *zoom, point, clear.*

In Table 11 the word *automatically* can refer to the automatic setting of "***Burst Mode*"**. The rest of words are not exactly typical for it. These are: *connect, device, lens, buy, great android, tablet, easy, facebook, email, need, good, love, want, alone.*

For the feature "***Camera Full Frame Sensor"*** in Table 12 we have a somewhat good output as the system finds terms that are the components of the feature's name, namely the terms *full, frame, full-frame, sensor.* Except these, *light* and *high* indicate properties of the sensor which are relevant.

In Table 13, which is about "***Shock Resistant"***, we can see some common terms with feature Water Resistant. To this surely the component *"Resistant"* of the feature's name plays an important role. Terms *work, kid, drop* could refer to Shock. High also is the term *vacation,* which is expected as someone would want to have durable electronic equipment during their vacation.

In Table 14 the terms *touch, screen, pocket* are good estimates for feature ***"Touch Screen"*** but the rest are very common to be significant. Maybe only term *work* which is very high listed could be vaguely relevant.

In Table 15 for ***"Cold Resistant"*** the first words are the same with those of Shock Resistant (*vacation, worry, trip*). Besides them though, which may indicate a relationship with cold, the rest (*drop, kid*) probably appear because of a strong relationship between Cold Resistant and some other characteristic, maybe even the feature Shock Resistant itself. The rest of the words do not seem interesting.

For feature ***"Integrated GPS"*** in Table 16, term *gps* is the first and indicates the feature itself, and the same does the term *system* while terms *mark, button* are also quite good. The other marked terms (*water, beach, snorkel, waterproof, underwater, pool*) are relevant in the sense that a GPS system is needed usually when someone is in an unknown place, which with its turn is directly relevant with vacation, travel etc. while it is not again unlikely that we have a strong correlation between Integrated GPS and some other feature.

In Table 17 that holds the results for ***"Weight"***, term *dslr* probably is relevant with weight because DSLR cameras are usually heavy. The same applies for the *professional* ones too, as for term *lens* as well, because it is an accessorie for mostly bulky cameras. Terms *snorkel* and *pocket* on the other hand indicate lightness.

Like Total Megapixels, feature "**Effective Megapixels"** in Table 18 has some terms for which we cannot make conclusions, like *purchase, upgrade, quality, dslr* and generally trivial words.

Feature "***Instant Print"*** in Table 19 gives some of the best results. Photo cameras with an Instant Print function, like Polaroids, are few in the market but for this exact reason this technical characteristic is emphasized greatly in the user reviews. Some of the words characterize directly the Instant Print feature while others like gi*ft, party, birthday, daughter* give the impression that this characteristic is fun to use and an ideal gift or for photos in small private events.

6.3 ***Conclusions***

In total, we can see our results are not all good but we can not ignore some very good estimates. Given that the preprocessing could be done more thoroughly and that our dataset is relatively small, we believe there is space for further exploring for associations. An additional observation is that it happens many technical characteristics with terms we have marked as good results, to have a small frequency of appearance for their value "*Yes"* comparing to the frequency of appearance for their value *"No"*. This is obvious in the table with the technical characteristics values' statistics in Chapter 4 for the features *Water Resistant, Shock Resistant, Cold Resistant, Integrated GPS, Camera Full Frame Sensor, Touch Screen* and *Instant Print.* All these are either very good lexicographical descriptions or at least relevant.

7

Epilogue

7.1 *Synopsis and conclusions*

Summarizing, in this undergraduate thesis we tried the application of Chi – Square Test for Independence on a population of photo cameras. The technical characteristics and the cameras' reviews became the variables that we gave as input to the problem to find if they are correlated. Finally, we interpreted and evaluated results of the Chi - Square Test intuitively and based on some statistics on our data.

7.2 *Future extensions*

An extension of the present work could be the implementaion of a Decision Tree which could use on its nodes as questions the existence or not of a lexicographical term in a camera's review, with the purpose of finding the best representation for a technical characteristic. Also, it would be very interesting to apply the methods we used in phrases that could be extracted from the reviews instead of words. Finally, a more thorough preprocessing and the gathering of more data could give improved results, something worth researching.

8

References

[1] Sreenivas Gollapudi, Samuel Ieong, and Anitha Kannan. 2012. Structured query reformulations in commerce search. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (CIKM '12). ACM, New York, NY, USA, 1890-1894

[2] Mahashweta Das, Gianmarco De Francisci Morales, Aristides Gionis, and Ingmar Weber. 2013. Learning to question: leveraging user preferences for shopping advice. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '13), Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy (Eds.). ACM, New York, NY, USA, 203-211

[3] Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (HLT '05). Association for Computational Linguistics, Stroudsburg, PA, USA, 339-346

[4] Lizhen Liu; Wentao Wang; HangShi Wang, "Summarizing customer reviews based on product features," in *Image and Signal Processing (CISP), 2012 5th International Congress on* , vol., no., pp.1615-1619, 16-18 Oct. 2012

[5] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '04). ACM, New York, NY, USA, 168-177.

[6] Nikos Sarkas, Stelios Paparizos, and Panayiotis Tsaparas. 2010. Structured annotations of web queries. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (SIGMOD '10). ACM, New York, NY, USA, 771-782.