J. Hlaváčová (Ed.): ITAT 2017 Proceedings, pp. 23–26
CEUR Workshop Proceedings Vol. 1885, ISSN 1613-0073, © 2017 M. Linková, P. Gurský

Attributes Extraction from Product Descriptions on e-Shops

Michaela Linková, Peter Gurský
Institute of Computer Science
Faculty of Science, P.J.Šafárik University in Košice
Jesenná 5, 040 01 Košice, Slovakia
michaela.linkova@student.upjs.sk, peter.gursky@upjs.sk

Abstract. Some e-shops present product attributes in structured form, but many others use the textual description only. Attributes of products are essential in automated product deduplication. We suggest methods for automated extraction of attributes and their values from product descriptions to a structural form. The structural data extracted from other e-shops are used as background knowledge.

1 Introduction

Nowadays there is an increasing interest in effective process of extracting information from big amount of data. The problem of searching and obtaining relevant information is handled by several areas of computer science. Project Kapsa [1] deals with extraction and unification of information from web pages, focusing on products on e-shops. The aim of the project is the creation and management of a collection of products which are offered by e-shops. Crucial part of processing the e-shops' data is a deduplication of products, i.e. the decision if any two products extracted from different e-shops are the same. To increase the precision of the deduplication, structured data about the products (product properties and their values) are essential.

Although some e-shops present attributes of products in table from, many other e-shops provide a textual description only. The descriptions usually contain values of many product properties and are written in natural language.

This work-in-progress paper presents our current methods of automatic extraction of product attributes with their values from product descriptions. Products have attributes of 3 main types: String, number with unit and Boolean. Each type is presented individually in natural language. Therefore we propose unique extraction method for each attribute type.

2 State of the Art

To extract product attribute/property with its value from a text description, we need to recognize that the attribute and/or its value are mentioned in the text. Named-entity recognition (NER) is a close research area to our problem. NER is the information extraction task of identifying and classifying mentions of people, organizations, locations and other named entities within text. Approaches to NER are surveyed in [3]. The dominant technique for addressing the NER problem is supervised learning. A usual NER method consists of tagging words of a test corpus when they are annotated as entities in the (rather big) training corpus. A semi-supervised techniques decrease amount of manual annotation needed to train a classifier. Typically, the sentences in Wikipedia articles are considered annotated, because they contain context links to other Wikipedia pages in sentences. The titles of such pages are then considered to be the names of the entities and their URLs become the identifiers. The common learning constellation for supervised and semi-supervised techniques is the processing of annotated texts. Majority of learning models process entity names as well as surrounding words. Many learning approaches have been used to handle NER: Hidden Markov Models [4], decision trees [5], Support Vector Machines [6], Conditional Random Fields [7].

Another approach, similar to NER, is terminology / entity / term extraction. The goal of terminology extraction is to automatically extract relevant terms from text, typically based on a vocabulary of domain-relevant (possibly multi-word) terms. Typical approach is to extract term candidates using linguistic processors and filter them using statistical and/or machine learning methods. The C-value/NC-value method [8] can be an example. To handle multi-word terms, the methods usually use n-grams, that is, the combination of n words appearing in the corpus.

3 Background Knowledge

Unlike general named entity recognition, as a part of natural language processing, we can profit from knowledge of product domain and drastically reduce the number of possible entities to search in product description. The product domain can be determined from the product web presentation, since it is usually presented on specific position on every product detail page of the e-shop.

The second advantage is the structured and annotated data of product domain in background knowledge. These data are extracted from the e-shops with structured attribute presentation in form of tables. Therefore, we can use the dictionary of the attribute names in different languages (English, Slovak …) and variations (synonyms, abbreviations) for each attribute of a given product domain. Similarly, we can use various forms of units' names (e.g. kg, kilograms, kilos, kilogramov, kíl …). Our background database contains also unit conversions between convertible units (e.g. grams vs. kg). Finally, the attribute types and the list of extracted values of each attribute and product domain is stored in the background knowledge.

The annotation of attributes in Kapsa [1] is a semiautomatic process driven by administrator in web GUI. Input for the annotation is a list of attribute names and values in String form for each product extracted from e-shop web pages, possibly with some additional tags. The annotation produces a set of rules that determines product domain, attribute identification (including attribute deduplication), attribute type, value and unit extraction, etc. If the product domain or attribute is already annotated for other e-shop, annotator usually just plays the role of the validator of an automatic annotation.

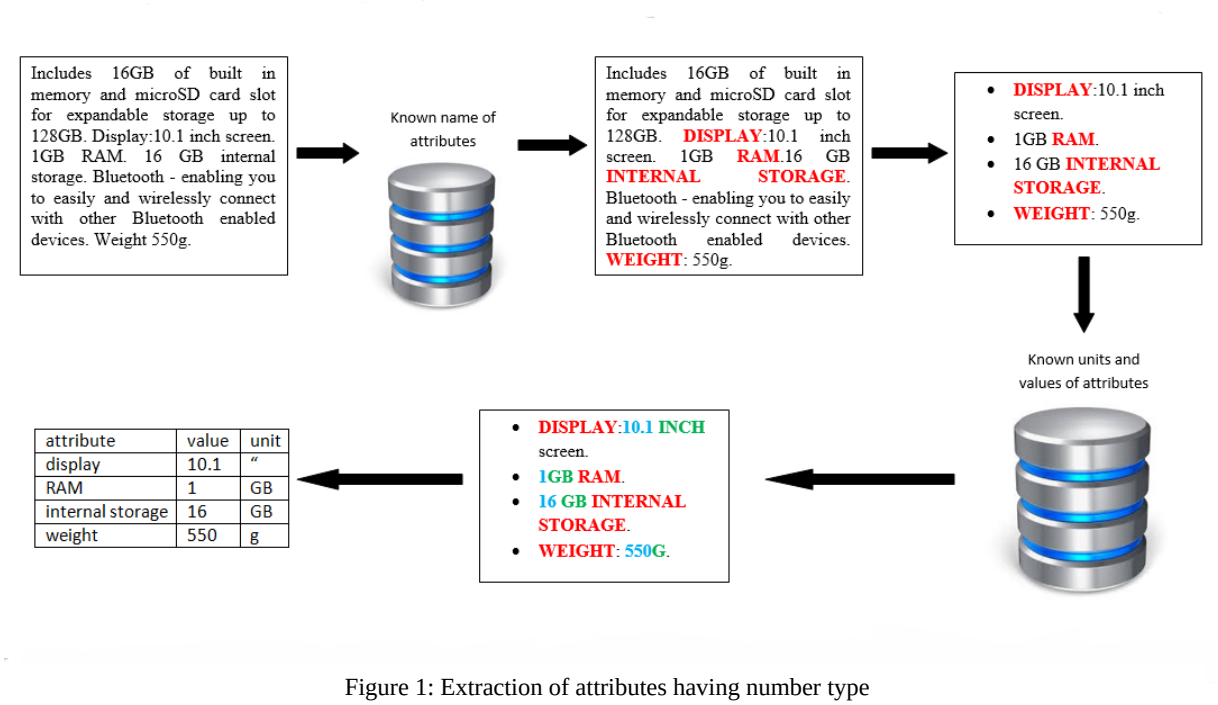

Figure 1: Extraction of attributes having number type

4 Extraction Methods

Our analysis of products' descriptions showed that attributes and their values are presented differently for Boolean, String and number types in natural language. Since we have the type information for each attribute we are searching for in background knowledge, we can utilize extraction method for each type. All of the presented methods are still works in progress and represent the baseline methods for the future work. Some of possible modifications, we believe that they can improve the quality of methods, are proposed at the ends of the following sections as well as in the experiments section.

4.1 Extraction of Boolean Attributes

Our method for extraction of Boolean attributes suggests, that the presence of the attribute name in product description induce that the value of the product's attribute is 'true'. The method searches every variation of the attribute name (languages and synonyms) that is present in background knowledge. If the attribute name is matched, the attribute with value 'true' is sent to output result.

Since quite a lot of the attributes were misspelled or inflected in our test data, we have replaced the exact search of the attribute names by fuzzyfied search using Levenshtein (editing) distance. The threshold for the positive result was set to 75% match of the attribute name. The method uses fuzzyfied search, only if the exact match is not found.

We believe that improvement of our method can be achieved by stemming or lemmatization of the attribute names and words of the product description to cover inflections as an equivalent to exact match. Another improvement can be to include common misspelled attribute names to background knowledge. However it requires an administrator intervention and converts our automatic method to the semiautomatic.

Our method does not recognize sentences as positive or negative. So the sentence *The mobile phone does not have thermosensor*, induces the result that the product has Boolean attribute thermosensor with value 'true'. The approaches known from sentiment analysis can be incorporated to cover this problem.

4.2 Extraction of Numeric Attributes

Attributes of number type have their values composed of a number and a unit (12 g, 15 cm, 42 '', 2 pieces…). Our method is composed of 4 main steps. First, the method searches for attribute names like the method extracting the Boolean attribute names. Next, all the variants of the attribute unit and the variants of units convertible to this unit are searched in the sentence, where the attribute name was found. The only exception is the unit *pieces*, because it is not common in natural language sentences (e.g. *2 shelfs* instead of *2 pieces of shelves*). In our method the search of the units requires an exact match. If both the attribute name and the unit was found, then, in the third step, the numbers are searched in the sentence using regular expression "([0-9]+)(\\ )*(\\.)*(\\,)*([0-9]*)". If the sentence contains more numbers, the closest number to the attribute name is selected in the 4th step.

The extraction method can be extended to cover word variants of the numbers (i.e. one, two, twenty-three...), but it requires new dictionary for each language. Stemming and lemmatization can be also used for unit search (in Slovak language there are 3 variants for singular and plural forms of units e.g. kilogram, kilogramy, kilogramov).

4.3 Extraction of String Attributes

String attributes are the most sensitive type to the size of background knowledge. The specialty of this type is that

Table 1. Precision, recall and F-score for English descriptions

Domain	numeric			Boolean			String		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
tablet	100	97.47	98.7	100	100	100	100	50	66.67
refrigerator	100	100	100	100	100	100	100	100	100
average	100	98.8	99.4	100	100	100	100	85.71	92.31

Table 2. Precision, recall and F-score for Slovak descriptions

Domain	numeric			Boolean			String		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
tablet	87.5	20.59	33.34	100	78.95	88.24	100	70	82.35
refrigerator	100	50	66.67	80	88.88	84.21	80	70.59	75
average	96.15	35.71	52.08	92	82.14	89.79	86.36	70.37	77.55

the attribute values are often self-explanatory and the attribute name isn't necessary. For example, in the sentence *"This Candy GC41472D1S Washing Machine with stylish Silver finish looks great in any home."* three String attributes can be found: producer (Candy), product name (*Candy GC41472D1S*) and color (*Silver*). If the washing machine was already extracted from another e-shop in structural form, all the String values are present in the background data and can be used to identify the attributes.

Extraction method for String attributes firstly searches for attribute names as well as the previous methods do. If the attribute name is found, values of the same attribute extracted from all products of the same domain are searched in the same sentence. If the value is found the attribute-value pair is sent to the result. String attributes of the product domain, which were not found in the first step, are searched only by their known values. Since each value corresponds to some attribute in the background knowledge, it is easy to send attribute-value pair to the result. The implemented method does not use the fuzzyfied search of the attribute values in product descriptions.

Similarly to the attribute names search, the attribute value search could by extended with stemming and lemmatization to cover inflections as an equivalent to exact match.

5 Experiments

To verify the methods, we created test data containing the real e-shops product descriptions of 2 domains: fridges and tablets. We have selected 20 products from each domain. 10 descriptions were in English and 10 were in Slovak. We have manually selected attributes and their values that appeared in the descriptions and typed them into the test table. Each product description was an input for our extraction methods and the results were compared to the manually selected ones.

Tablet descriptions contained 4 Boolean attributes, 1 String attribute and 5 Number attributes. Fridge descriptions contained 4 Boolean attributes, 4 String attributes and 9 number attributes.

The background knowledge was created by extraction of structured data from 2 e-shops with table representation of attributes. Data contained 142 tablets and 41 fridges[1]. All attributes found in test data descriptions were present in background knowledge.

The results of our tests are summed up in tables 1 and 2 separately for English and Slovak descriptions.

5.1 Results for Numeric Attributes

Method for attributes of numeric type correctly found 98.8% of all attribute name and value pairs in English descriptions, but only 35.71% of pairs in Slovak description. Such a low recall in our test is caused by various reasons. We have analyzed the results and identified the following problems:

- the absence of synonymic names of given attribute in the background dictionary,
- the absence of the synonymic unit of the attribute value,
- presence of a shortcut, instead of full form of attribute name, or missing words of the full multi-words terms,
- missing attribute name (just the value and units were present in the description),
- different order of words in multi-word name of attribute, and
- other words inserted into multi-word name of attribute.

The first three problems are caused by a small dictionary. After adding more e-shops to the background knowledge, it should become a less important problem. Different e-shops can use different terminology and unit abbreviations, which expands the background knowledge dictionary.

Sentence "*V chladničke je možné uchovávať 225 l potravín v 4 sklenených poličkách"* (en. It is possible to store 225 l of groceries on 4 glass shelves) mentions the

[1] Dataset is available at: http://kapsa.sk/2017-itat-dataset.zip

volume of the refrigerator and the number of shelves in the refrigerator, but because the full names of the attributes are not present in the sentence, the method for numeric types did not find these product properties in the sentence. A definite solution for the missing attribute name problem would probably not be easy. One approach can be to use attribute values' units. If the unit found in the description, is used by only one known attribute of the product domain, the value and unit can be assigned to the attribute.

The last two reasons deal with multi-word names. The solution to the problem can be to search each word of the term separately. If each word of multi-word term was found in the same sentence, then we can declare the match. It is possible that automatic morphological analysis of the sentence can improve this approach, because it can reveal the connections between words and reduce false matches of such method.

The precision of the method is decreased by fuzzy matches, when the editing distance of 75% was too generous and matched the words with different meaning. We can improve the precision using stemming or lemmatization instead of fuzzy matching with editing distance. Another improvement can be achieved by accepting fuzzy matched words only if they are not present in classic dictionary of the language, i.e. they are probably misspelled.

5.2 Results for Boolean Attributes

The method for Boolean type of attribute was the most successful in finding attributes. Using this method, all the required attributes were found in the English descriptions and 82.14% of the attributes in the Slovak descriptions. The reason for not finding attributes in our tests within Slovak descriptions was similar to the synonymic variations mentioned in the previous method. Concretely, the term in our dictionary had fewer words, because some words were split into two words. Since we do fuzzy comparisons word-by-word, it made the match less than 75%.

For example, the sentence *Už žiadna námraza, Technológia No Frost zabraňuje vzniku námrazy a udržiava konštantnú teplotu v celej chladničke*, (en. No more frost cover, the technology No Frost prevent frost creation and keeps constant temperature throughout the fridge) didn't match with our two-word term *Technológia NoFrost*. The solution would be to add *Technológia No Frost* to the directory.

Since we used Levenshtein distance to search for a name, the method found two attributes in two descriptions that were not there. These were the Auto Defrost and NoFrost attributes.

5.3 Results for String Attributes

The method for attributes of String type is special, because it does not need the attribute name. It causes the ambiguity of the attribute assignment.

For example, in the sentence *Farba kombinovanej chladničky Goddness je biela*.(en. The color of the Goddness fridge is white.), the value *biela* (en. *white*) is appropriate for attributes color and color of the front of the refrigerator.

The second problem is again the small dictionary, this time, the dictionary of known attribute values. For example, in sentence *Pri hrúbke len 6,1 mm je vôbec najtenší iPad zároveň aj najschopnejší* (en. Having the depth only 6.1 mm, it is the thinnest iPad as well as the most capable.), the method did not found attribute "product name", since *iPad* value is not in the value dictionary. Again, to remove the problem of the absence of an attribute value, it is sufficient to increase the set of attribute values in the dictionary.

The precision was decreased by false fuzzy match of the attribute value with different word. Again, we can improve the precision using stemming or lemmatization instead of fuzzy matching with editing distance.

6 Conclusions

This work-in-progress paper presents our base-line algorithms for automatic attribute-value pairs extraction from product descriptions on e-shops. We divided attributes to 3 main types: Boolean, String and numeric. Boolean attributes are matched, if the name is found in the description. String attributes are search by match with pair attribute name and its value or by value only. Numeric attributes require three things to find: attribute name, number and unit.

We have probed our methods against real world data, analyzed the results and proposed the improvements that would be incorporated in our methods in the future.

This work was supported by the Agency of the Slovak Ministry of Education for the Structural Funds of the EU, under project CeZIS, ITMS: 26220220158

References

[1] Project Kapsa, web page: http://kapsa.sk/
[2] J. NothMan at al.: Learning multilingual named entity recognition from Wikipedia. Artificial Intelligence 194 (2013) 151–175
[3] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (2007) 3–26
[4] D. M. Bikel et al.: Nymble: a High-Performance Learning Name-finder. In ANLP-97, Washington, D.C., pp. 194 – 201, 1997.
[5] J. Cowie: Description of the CRL/NMSU System Used for MUC-6. In Proceedings of the Sixth Message Understanding Conference, Morgan Kaufmann, 1995
[6] J. M. Castillo et al.: Named Entity Recognition Using Support Vector Machine for Filipino Text Documents. International Journal of Future Computer and Communication, Vol. 2, No. 5, October 2013
[7] J. Lafferty, A. McCallum, F. Pereira: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In proceedings of ICML, pages 282–289., 2001
[8] K. Frantzi, S. Ananiadou, J. Tsujii: The C-value/NC-value Method of Automatic Recognition of Multi-word Terms. In proceedings of ECDL, pp. 585-604. ISBN 3-540-65101-2, 1998