

ISSN 2281-4299



DIPARTIMENTO DI INGEGNERIA INFORMATICA  
AUTOMATICA E GESTIONALE ANTONIO RUBERTI



**SAPIENZA**  
UNIVERSITÀ DI ROMA

**Ontology Extraction from Question/Answer  
Sections on Online Marketplaces**

Sara Di Bartolomeo  
Riccardo Rosati

Technical Report n. 3, 2018

# ONTOLOGY EXTRACTION FROM QUESTION/ANSWER SECTIONS ON ONLINE MARKETPLACES

Di Bartolomeo, Sara\*  
dibartolomeo.sara@gmail.com

Rosati, Riccardo\*  
riccardo.rosati@uniroma1.it

April 4, 2018

## Abstract

Online marketplaces are a significant part of the global market. They aggregate an enormous amount of data about the products that they are selling, in the form of product descriptions and reviews. Users form their opinions about an item according to other users' opinions, expressed in the form of reviews. A number of marketplaces are also starting to use a question/answer system along reviews, in which any user can post a question and another user can answer to that same question. There is a wealth of relevant data, though this information is expressed in natural language, meaning that it's understandable for a human being but difficult to translate into meaningful data for a computer program. We addressed the problems involved with the programmatic analysis of the language contained in question/answer sections with the purpose of extracting information to build and populate an expressive knowledge model. The approach we choose relies on the use of named entity recognition and part-of-speech tagging to identify concepts to populate the ontology, and learn relations between the elements of the ontology.

**keywords**— ontology learning, entity recognition, information extraction

---

\*Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti DIAG, Sapienza Universita' di Roma

# 1 Introduction

In recent years, the shopping habits of the general public have been changing considerably: more and more people are choosing to use online marketplaces for their purchases. This form of shopping platform has introduced several advantages: users have access to a much bigger selection of products, and barriers that would otherwise have prevented access to the market to smaller sellers have been broken down.

Nevertheless, these advantages are balanced by other downsides. As an example, judging the quality and features of an item is a more complex task, as an user will only have access to photos and textual descriptions of the item.

In this context, a trust-based system between multiple users has gained more and more relevance: reviews and reputation. The judgement of a buyer will be substantially affected by other users' reviews, as well as a level of trustworthyness assigned to sellers by other users.

Making an user able to gain knowledge about the quality, purpose and details of an item is a problem that concerns not only the users, but also the sellers. Indeed, a number of good reviews may be the deciding factor for a successful sale.

Along with reviews, a number of online marketplaces started introducing question/answer systems, in which users are able to ask questions to other users about a particular product. An excellent example of the success of this approach is Amazon: the most popular products on the platform have thousands of questions and answers, addressing each property or use case of a product that may be unclear to the consumers.

## 1.1 The problem with unstructured information

Questions and Answers are, though, unstructured information about a product. They are written in natural language, that is the language that is

commonly used by anyone when speaking or writing, and that is not easily interpreted by computer programs.

Indeed, if we consider a text describing a product like this one:

*This is a black phone, it has two sim slots, a long lasting battery and the screen looks very good.*

It is easy, for a human reader, to understand that the text is describing an item belonging to the category **phone**. In other words, a human reader classifies the informations contained in the text as described in the table below:

category	phone
color	black
sim slots	2
battery	>3000 mAh
screen	good-looking

That is a much more structured form, produced by the ability of the human brain to semantically understand the language contained in the text.

If we also consider the issue of querying the products, having product data stored in a structured way would allow consumers to perform much more precise queries. If we search for a *"black phone"* on Amazon right now, the first result is a *"black phone cover"*. This happens because Amazon only matches our query into the title strings of the products. If we were allowed to specify that we want an object of **category: phone** and **color: black**, the search would have returned the correct set of results.

Still, the amount of data we are dealing with requires us to create an automated system to extract the information and store it in a meaningful way. The aim of this project is to propose a method for extracting and structuring information from questions and answers on online marketplaces. To structure and store the information, we chose to use an ontology.

We consider each product on the marketplace as an Entity in the ontology. Each product has a set of characteristics, that may also be shared among different products (e.g. multiple products may share the same brand). Each product is classified in categories, that are the classes in the ontology.

Since the task of manually populating an ontology was too time-consuming, the need for a system for automatic extraction of ontologies from text was poignant for this purpose. The field that studies information extraction with the purpose of building ontologies is called **ontology learning**, and a number of research efforts and different approaches have been proposed for the task in the last years.

The project is composed of three main steps: fetching the data, processing it, and populating an ontology with the results of the analysis. The output of each step is the input of the following one.

## 1.2 Approaches for ontology learning

The objective of automatic or semiautomatic population of ontologies has been addressed with several techniques. There are, though, some details to take into account: the domain and the structure of the input. The input can in fact come in multiple forms: unstructured natural language, semi-structured documents (e.g. XML), an existing knowledge base, a dictionary, or a database. In our case, the input was mostly made of unstructured text, with some added information given by the HTML structure of the product page on the marketplace that we incorporated into the results of the extraction.

Examples of ways to extract structured data from natural language text are:

- **Word co-occurrence:** (Roark and Charniak 2000, Yarowsky 1995) Simply measuring how many times two words appear in the same phrase can give us an idea of how much two words are related. In Roark et al., a number of 'seed' words, representative of several categories, is chosen at the beginning, then the words that co-occur with the seed the highest amount of times are added to the respective categories. Also related to **association rules**.
- **Pattern-based extraction:** (Thelen and Riloff 2002) Similar to regular expressions, we look for patterns in the text. Starting from 'seed' words - we look for predefined patterns such as "<some verb>

<some object>” (e.g. ”is made of <X>”)

- **Clustering-based techniques:** (Schickel-Zuber and Faltings 2007, Buitelaar and Pustejovsky 1998) a distance measure of terms has to be defined in order to find out which terms are most similar and cluster them in groups. One particular word - the hypernym of the cluster (Hearst 1992) - can be extracted from each cluster to be representative of that cluster. If we extend this to hierarchical clustering, we can obtain a hierarchy of words. Methods using Latent Semantic Indexing or Latent Dirichlet Allocation fall under this category, building clusters by trying to discover topics.
- **Part-of-speech tagging:** (Maedche and Staab 2001, Maedche and Staab 2000) Part-of-speech tagging (identifying subject, verb, object in a phrase) is used to discover entities and possible relations in the phrases. **Named entity recognition** may be used to identify the types of particular named entities.
- **Ontology pruning:** (Kietz, Maedche, and Volz 2000, Volz et al. 2003) A generic ontology (e.g. WordNet) is used. Input text is classified according to the generic ontology to acquire domain concepts. The generic ontology is pruned to remove non-domain specific entries. Decision about which entry is domain specific and which one is not is often based on the assumption that a domain-specific concept is mentioned more often in a domain-specific corpus than in generic text.

### 1.3 Methodology

We collect questions, answers and product pages from Amazon. Keeping track of the product page allows us to use some of the information contained in them. The products are classified in classes, from which we build a taxonomy to have an initial structure. Then, we analyze the questions and answers for each product. Each question is classified in open-ended or closed-ended. We then perform part of speech tagging to identify the subject, the

verb and the object of the question. The idea is that the *subject* of a phrase has a relationship of type *verb* with the *object*. Since we are dealing with questions, we also have to take into account the answers, that are classified into positive and negative answers with a simple Naive Bayes classifier. The same relationship translates to a role or an attribute in the ontology. Named entity recognition is used to identify possible entities in the questions and classify them in different categories. Wordnet is used to find synonyms and aggregate them into the same structures. For ontology management, we used the *owlready2* python library, we used *owl* for representing the ontology, and for natural language related tasks we used *pattern*, another python library.

## 1.4 Structure of the article

In the following chapters, we'll describe in depth these three steps:

- The first chapter, **Fetching questions and answers**, describes the initial step of gathering the data used for the project, including the challenges and the methodologies. The questions and answers have been collected from Amazon via scraping, so the process required some foresight to be efficient and functional.
- The second chapter, **Natural Language Processing**, describes how we extracted relevant information from the wealth of text collected previously. Natural Language Processing is the field that studies the interpretation, from an algorithmic point of view, of phrases in commonly written/spoken languages. Thanks to advancements in this field, we are able to identify, extract and classify details about the products from the reviews.
- The last chapter, **Ontology Extraction and Population**, describes how the analyzed data has been used to programmatically populate an OWL ontology.

Each chapter refers to a slightly different field of study, and starts with a general description of the field and of the techniques, concepts and algorithms used, and then describes how the same concepts have been applied during the development of the project.

## 2 Fetching questions and answers

Currently, products on Amazon are the focus for the domain of the project. The same techniques could be applied to other websites containing similar content, but we choose to concentrate our efforts on Amazon. As of November 2017, Amazon has 573 million products currently on sale <sup>1</sup>, and is undoubtedly a giant in its field. In the last couple of years, they also attempted the sale of a wider number of products, like fresh food and pantry items, expanding the range and types of products that can be found on their platform.

More than the number of products that the marketplace offers, Amazon gives to the buyers the ability to review items, and post questions and answers to previously published questions. Precise data about the numbers of reviews and questions present on Amazon remains undisclosed, but a popular product can easily reach thousands of both. Having access to user comments is incredibly relevant to online buyers, because the trust they can put in other users is the best way they have to judge the quality and properties of a product.

Unluckily, Amazon API was not useful to collect questions and answers, as they are not offered among the possible data obtainable via their methods. Therefore, we needed to scrape the pages in order to obtain them.

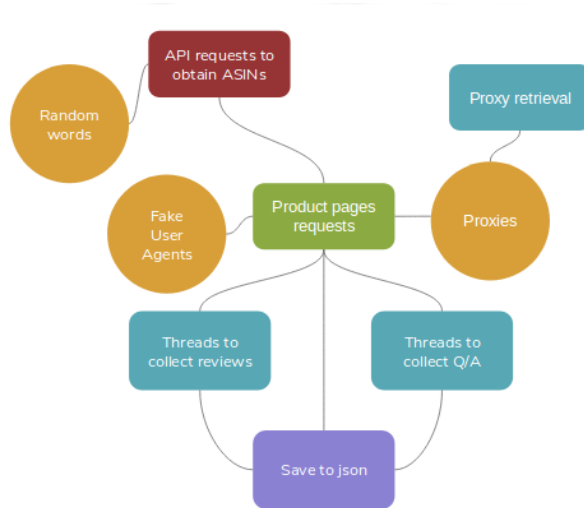
To build the scraper, we payed particular attention to two aspects:

- **It needs to avoid Robot Checks.** Robot Checks are particular pages that Amazon can use to respond to GET requests. They contain no information about the product, and only serve to block scrapers attempting to download the pages via scripts. In order to not let Amazon suspect of the program as a scraper, we made it so that each request to Amazon has a randomly crafted user agent and a random proxy selected from a pool of 400 proxies each time.
- **It needs to avoid bottlenecks.** When dealing with a huge number of HTTP requests, bottlenecks are a particularly important aspect

---

<sup>1</sup>source: <https://www.scrapehero.com>

because of the slowness and low reliability of transmission. Therefore, our scraper needed to be multithread and be very tolerant to network errors.



11

### 3 Natural Language Processing

An information extraction system is a system that looks through large bodies of text for specific types of entities and relations. The text is first segmented, tokenized, and then part-of-speech tagging is performed. The resulting data is used for named entity recognition: we look for specific types of entities. Based on the position of the words in the text, we try to determine if there exists a particular relationship between the entities. The process is shown in 1.

The main idea is that, after we learn which one is the subject of a phrase, we can say that <subject> has a relationship of type <verb> with the <object>, with adjectives if there are some.

Therefore, the steps to obtain the final results are:

- **Lexical Analysis:** Tokenization in words, chunks, sentences the text.
- **Part-of-Speech tagging:** Mark chunks with part-of-speech tags

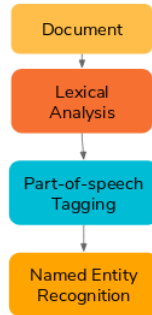


Figure 1: Steps in the natural language analysis

(nouns, verbs, adjectives...) based on context and definition.

- **Named Entity Recognition:** Locate and classify entities in text into predefined categories (locations, organizations)

### 3.1 Closed-ended and open-ended questions

Questions about a product can be open-ended or closed-ended. The affiliation of each question to one of these categories entails major differences in how information is expressed.

A **closed-ended** question is a question that accepts 'yes' or 'no' as answer. An example of a closed-ended question is:

**Question:** Does this phone have a camera?

**Answer:** Yes.

An **open-ended** question is, instead, a question that expects a more complex answer, such as a list, an explanation, a description. An example of an open-ended question is:

**Question:** What features does this phone have?

**Answer:** A camera, two sim slots and a headphone jack.

As you can see, information is expressed in a very straightforward way in closed-ended questions. If we, through natural language processing, manage to understand the object of the question, the answer is just a boolean value representing if the analyzed product corresponds to the requested quality or not.

The approach used in this section is based on the research of (McAuley and Yang 2016), who in turn based their approach on a paper from two researchers at Google, (He and Dai 2011).

According to He and Dai, recognizing a closed ended question is similar to matching a regular expression. If we, in fact, consider the following categories:

**Be verbs :** {am, is, are, been, being, was, were, }

**Modal verbs:** {can, could, shall, should, will, would, }

**Auxiliary verbs:** {do, did, does, have, has, }

A closed-ended question is formed in this way:

$$[ S_{be} | S_{modal} | S_{aux} ] . * +? \quad (1)$$

Still, this approach would catch as closed-ended questions constructs like "Is he married or not?" or "Does anybody know the price of this product?", that are clearly not closed-ended. Therefore, we exclude from the set of closed-ended questions the two following formulae:

$$[ S_{be} ] [ a - z ] * [ or ] [ a - z ] ? \quad (2)$$

and

$$[ a - z ] * [ anyone | anybody ] [ a - z ] * [ tell | know ] [ a - z ] ? \quad (3)$$

He and Dai claim that this approach detects 91% of the questions correctly.

## 3.2 Named Entity Recognition

With the term "named entities", we refer to parts of the phrase that indicate specific types of individuals. The names of organizations, names of people and dates are good examples of details that could be identified as named entities. The table below reports types and examples of the most common classifications of named entities: while organizations, persons, location, date, time, money and percent may be self-explanatory, "GPE" indicates "Geopolitical entities" (e.g. country names), while "facility" refers to monuments and specific places.

ORGANIZATION	Google
PERSON	President Obama
LOCATION	Mount Everest
DATE	June, 2008-06-29
TIME	two fifty a m, 1:30 p.m.
MONEY	175 dollars
PERCENT	18.75%
FACILITY	Stonehenge
GPE	South East Asia

Named Entity Recognition (NER) is the practice of discovering and identifying all the named entities in a given text. First, the boundaries of the named entity need to be recognized (e.g. how many and which words compose a given NE). Then, the type of NE is identified, meaning which one of the aforementioned types does the NE belong to.

Named Entity Recognition can also be useful to find the answer to a question in a document, by recognizing and isolating the chunk that contains the answer. Suppose we have the question "Who was the first President of the US?", and we know that the answer is contained in the passage:

The Washington Monument is the most prominent structure in Washington, D.C. and one of the city's early attractions. It was built in honor of George Washington, who led the country to independence and then became its first President.

We know that we should expect, as a response, a named entity classified as a PERSON. Although the phrases contains the word "Washington" twice,

only one of them will actually be identified as a person.

Since we wanted to focus the object of the analysis on phones, we added three new types to the standard set of types used for Named Entity Recognition:

TECHNOLOGY	IR sensor, bluetooth, NFC
OS	Android, Marshmallow
CARRIER	Vodafone, Verizon

## 4 Ontology extraction and population

To reach our goal, we needed to define bridges from the data we extracted and the representation language we choose.

- Each item on Amazon is an OWL **class**, which describes a set of instances - that are the physical objects with the same name and same characteristics.
- Each category of items on Amazon is a **class**.
- Each property that an item can have is either a **Data Property** or an **Object Property**. More on this to be discussed in later sections.
- Properties extracted from questions and answers may either be considered standard data types (strings, integers, floats, dates) or having their own class if there is more data regarding them specifically. Examples are the classes **Brand** and **Author**.

### 4.1 Taxonomy

First of all, we extracted a taxonomy based on the item categories already present on Amazon. Each item, indeed, belongs to a category that describes a broader set of items. Each category may have child categories, and a parent category, designed mainly to ease navigating through products,

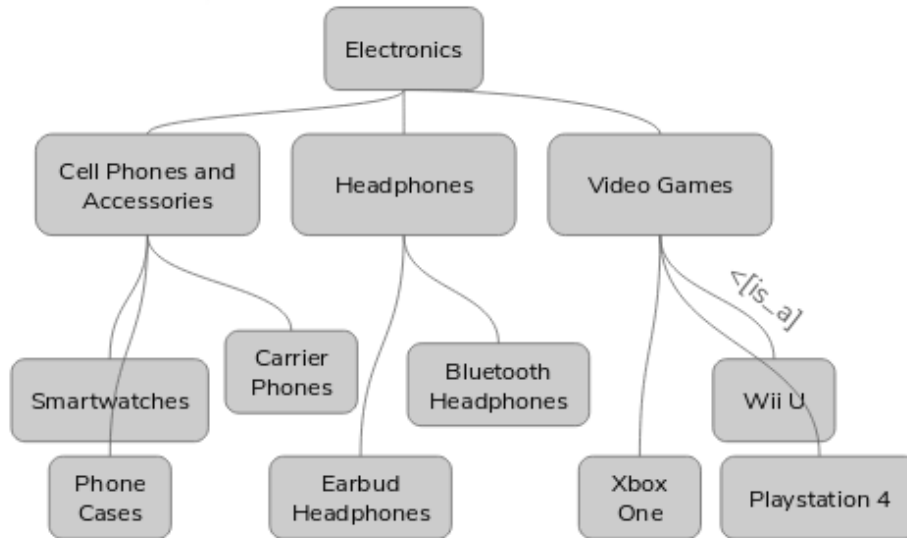


Figure 2: Taxonomy extracted from the structure of product pages

but not always consistent - some categories may overlap in the items they contain.

These classes form a tree-like hierarchical structure.

For example, the item `Trivial Pursuit Game: Classic Edition` is classified as belonging to the class `Board Games`. The class `Board Games` is a subclass of the class `Games`, which is in turn subclass of `Toys & Games`.

`Super Jumbo Playing Cards`, instead, are categorized as `Standard Playing Card Decks`, subclass, again, of `Games`

The same structure of categories has been extracted from the page and represented in OWL as classes and `subClassOf` relationships.

Amazon does not offer a clear visualization or explanation of the categories. The tree of categories has been rebuilt by scraping the content of the page

and extracting the data about how the product is classified. In figure 2, a part of the discussed taxonomy extracted from the documents is shown, about the product category "Electronics".

## 4.2 From text to property

Once we have questions and answers about a product, and once we analyzed their contents, we map the extracted properties into the ontology.

**Question:** Does this phone support 4g?

**Answer:** Yes, it does.

The part-of-speech tagging of this questions tells us that **this phone** is the subject (SBJ) of this phrase, **support** is the verb, and **4g** is the object. We also know, thanks to named entity recognition, that 4g is a named entity belonging to the category technology. Another thing we can learn is that the answer is positive.

We can now state that the ontology class that we are analyzing, a phone, has a relationship of type "supports" with another item, technology 4g, and therefore add a property to the class into the ontology linking the phone with 4g.

Due to the huge amount of questions for each product (and the human, error-prone nature of the answerers), there are obviously going to be contradictions in the answers. When more than one question is present about a specific aspects, the property is added according to what the majority of the answers are saying, even if there are contradicting answers.

## 5 Results

The aim of this project was to propose a method for extracting and structuring information extracted from product reviews and more metadata on an online marketplace, a type of market in which reviews are particularly relevant.

At the end of the data collection step, we had access to 62651 reviews, 37990 questions and answers, and 7927 product pages.

The ontology produced in the end contains 5243 products and 97870 axioms, extracted from the natural language of the products.

Each product has properties extracted from the structure of the page and from the content of questions and answers.

We believe that the results of this project may be relevant not only for research purposes, but also for an eventual development for a useful tool for both clients of online marketplaces and sellers.

Suppose, for example, that you are a seller, and want to learn more about the public's opinion of your product based on reviews. One possible approach would be using sentiment analysis on the text, or considering the star rating that accompanies each review. Still, that wouldn't inform you on what details of the product are producing bad or good reviews. For example, that your company produces phones, but one of your phones is having bad reviews. What details of the phone are causing the bad reviews? Is it the duration of the battery? Is the screen not bright enough? Or is the design of the phone considered ugly?

This kind of information is made easily accessible, allowing a seller to better identify the public's perception of the products, and without needing to read through the wealth of reviews that every product receives.

Now imagine that you are a customer, and you want to look for a specific product. You want a phone that has a durable battery, two sim slots, and you want it to have NFC. Amazon's search bar looks only through the items' names, and may therefore report not precise results. By having access to this ontology, instead, the search could be now a DL-query written as:

```
Phone and has_sim_slots value 2 and has_battery value "3300
      mAh" and supports_technology value "NFC"
```

## References

- Buitelaar, Paul and James Pustejovsky (1998). *CoreLex: systematic polysemy and underspecification*. Brandeis University.
- He, Jing and Decheng Dai (2011). “Summarization of yes/no questions using a feature function model”. In: *Asian Conference on Machine Learning*, pp. 351–366.
- Hearst, Marti A. (1992). “Automatic acquisition of hyponyms from large text corpora”. In: *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pp. 539–545.
- Kietz, Joerg-Uwe, Alexander Maedche, and Raphael Volz (2000). “A method for semi-automatic ontology acquisition from a corporate intranet”. In: *EKAW-2000 Workshop Ontologies and Text, Juan-Les-Pins, France, October 2000*.
- Maedche, Alexander and Steffen Staab (2000). “The text-to-onto ontology learning environment”. In: *Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures*. Vol. 38. sn.
- (2001). “Learning ontologies for the semantic web”. In: *Proceedings of the Second International Conference on Semantic Web-Volume 40*. CEUR-WS. org, pp. 51–60.
- McAuley, Julian and Alex Yang (2016). “Addressing Complex and Subjective Product-Related Queries with Customer Reviews”. en. In: ACM Press, pp. 625–635. ISBN: 978-1-4503-4143-1. DOI: 10.1145/2872427.2883044. URL: <http://dl.acm.org/citation.cfm?doid=2872427.2883044> (visited on 12/03/2017).
- Roark, Brian and Eugene Charniak (2000). “Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction”. In: *arXiv:cs/0008026*. arXiv: cs/0008026. URL: <http://arxiv.org/abs/cs/0008026> (visited on 01/05/2018).
- Schickel-Zuber, Vincent and Boi Faltings (2007). “Using hierarchical clustering for learning the ontologies used in recommendation systems”. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 599–608.
- Thelen, Michael and Ellen Riloff (2002). “A bootstrapping method for learning semantic lexicons using extraction pattern contexts”. In: *Pro-*

*ceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 214–221.

Volz, Raphael et al. (2003). “Pruning-based identification of domain ontologies”. In: *J. UCS* 9.6, pp. 520–529.

Yarowsky, David (1995). “Unsupervised word sense disambiguation rivaling supervised methods”. In: *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 189–196.