

RESEARCH

Open Access



An aspect-driven method for enriching product catalogs with user opinions

Tiago de Melo^{1,2}, Altigran da Silva^{1*} and Edleno S. de Moura¹

Abstract

In this paper, we propose a method for enriching product catalogs, which traditionally include only objective data provided by manufacturers or retailers, with subjective information extracted from reviews written by customers. Our method was designed to associate opinions taken from reviews with the product attributes they refer to. This is done by matching aspect expression identified in opinions with attributes from the product, which we model here as *aspect classes*. To verify the effectiveness of our method, we executed an extensive experimental evaluation that revealed that customers frequently mention aspects related to product attributes in their reviews. The attributes often receive more mentions than the product itself. Our method consistently reached almost 0.7 of F_1 measure in the task of associating the opinion with the correct attribute (or with the product as a whole), across four product categories, in two different scenarios. These results significantly improved the results achieved by a representative baseline.

Keywords: Product catalog enrichment, Aspect-based summarization, Opinion mining, Sentiment analysis, e-Commerce, Online reviews

Introduction

In typical e-commerce Web sites, descriptions of products in the catalog usually consist of static, objective, and factual data provided by manufacturers and retailers informing customers of product's characteristics, which are represented as *attributes*. For instance, for laptops, the brand, the weight, and the processor model are commonly available to help potential customers make their purchase decisions. However, with the rise of the so-called Web 2.0, there is a large amount of dynamic, subjective, and opinative information available on products and their characteristics. This information can be aggregated to the product description to potentially enrich the customer's knowledge about the product, improving their decision-making process. In most cases, this subjective information is provided by opinions issued by other customers in evaluative texts, e.g., reviews posted in forums, blogs, or e-commerce Web sites.

Users of typical e-commerce Web sites, such as Amazon or Best Buy, can easily find out that the laptop *Lenovo Yoga 710* has an *Intel 7th Generation 1.6 GHz Core i5 processor*.

Although this information is important, some customers may find it difficult to properly evaluate whether this is a good reason to buy the product or not. Therefore, it would be interesting if the customer could also consider other customers' opinions of a particular attribute. Figure 1 shows a review written by a real customer of this laptop. From the first sentence of this review, we realize that this customer thinks that the laptop has excellent speed and performance. Intuitively, these opinions count positively towards the laptop's processor.

In fact, considering opinions issued by other people before purchasing a product is a common practice, especially since there are plenty of opinions available on the Web [1–3]. According to the Wall Street Journal, 92% of people put more trust in information individuals published in social media about products and services than information published in other, more traditional sources, such as advertisements [4]. More recently, a comprehensive survey of online shoppers from five different continents revealed that 45% of consumers consider reviews as the most influential aspect of social media for their online shopping behavior [5]. This survey also shows that checking reviews about products or retailers is the fourth most common activity of in-store shoppers with mobiles/smartphones.

*Correspondence: alti@comp.ufam.edu.br; tmelo@uea.edu.br

¹Institute of Computing, Federal University of Amazonas, Manaus, AM, Brazil
Full list of author information is available at the end of the article

The speed and performance is excellent and I really like the option to detach and use as a tablet and to have a sturdy keyboard available is really nice. I am in love with this laptop.

Fig. 1 Review of the laptop Lenovo Yoga 710

Unfortunately, it is not generally feasible for an ordinary potential buyer to examine a large set of reviews on a given product for useful information on certain attributes. The reasons are multifold. First, reviews are often large and detailed, covering many different characteristics of the product being reviewed. Readers need to carefully inspect each review to find the information they seek. Second, even in shorter reviews, customers may refer to specific attributes using several different expressions, and readers can overlook disparate expressions that refer to a same attribute. For instance, in Fig. 1, both expressions “speed” and “performance” refer to the laptop’s processor. Third, retailers sometimes try to ease the decision-making processes by providing numeric ratings (e.g., stars, likes) to summarize customer opinion. These summaries refer to the product as a whole and are hardly useful for customers who seek information on specific characteristics of a product. Finally, especially for popular products, there can be too many reviews to analyze. In principle, having many opinions on a product can potentially improve the decision-making process, even in spite of problems such as controversies and fake reviews [6]. However, it may be unattainable for customers to evaluate a large number of reviews on a product. In fact, the time and effort spent by customers on this task cannot scale at same pace as the volume of product reviews provided by e-commerce Web sites.

To overcome these limitations, we propose a method whose goal is to enrich product catalogs, which traditionally include only objective data provided by manufacturers or retailers, with subjective information extracted from reviews written by customers. Our approach is in-line with recent efforts that aim at enriching static structured data repositories with lively subjective information obtained from the Web [7].

A critical task of our method is to associate opinions taken from user reviews with product attributes they refer to. For instance, as mentioned above, in Fig. 1, users express positive opinions regarding the laptop’s processor. Figure 1 also illustrates an opinion which is not associated to a specific attribute of this laptop, but to the product as a whole, when the user says “I am in love with my laptop.” The user can still comment on a characteristic of the

product that is not represented in the product catalog. For instance, in Fig. 1, positive user opinions such as *the option to detach* and *to use the laptop as a table* does not refer any specific attribute of the product, nor the product as a whole. Considering these examples, we established in our work that the opinions written by users can be targeted for existing attributes in the product catalog, the product as a whole, or some characteristics that are not represented in the product catalog. These three cases are covered by our method.

The problem we tackle in this paper is related to the well-known problem of *aspect-based opinion summarization*, the grouping of opinions about an entity into clusters according to aspects they contain. Each aspect cluster summarizes the user’s opinions towards an aspect of this entity. This problem has been widely studied in last decade in the literature [8–12], and it has also been applied in real systems such as Google Product Search and Bing.

When grouping opinions by aspects, one must consider that users often refer to a same aspect using distinct terms or *aspect expressions* [13]. For instance, the phrase “performance is excellent,” found in Fig. 1, is likely used to express a positive sentiment regarding the laptop’s processor and, as such, can be associated to the aspect cluster “processor” of this particular laptop. Thus, there have been proposals to categorize aspect expressions according to the aspect clusters they refer to [10, 14–17].

To verify the effectiveness our method, we executed an extensive experimental evaluation using more than 450,000 real reviews, composed of more than three million sentences, and more than 22,000 products in four different categories of electronic products (cameras, DVDs, laptops, and routers). The results from this evaluation revealed that customers frequently mention aspects related to product attributes in their reviews. Sometimes, attributes receive more mentions than the product itself as a whole. The opinions are distributed among many distinct attributes. This validates our hypothesis that reviews are valuable sources for enriching product catalogs with subjective information.

When evaluating the quality of our method’s results for the task of mapping opinions to targets, our method consistently reached almost 0.7 in terms of F_1 across all

categories, in two different scenarios. This is a significant improvement compared to the results achieved by our baseline.

In summary, the contributions of this paper are as follows:

- We introduce a novel problem of product catalog enrichment with user opinion provided from social media. We describe the specific challenges that this problem poses, based on an analysis of real user opinions. Solutions to this problem are important for a wide range of applications such as recommender systems and product design.
- We describe the architecture and implementation of our method. Experimentally, the proposed method outperforms the baseline method that can be applied to the task. We demonstrate that our method achieved promising results with F_1 above 0.75, which is significantly higher than the baseline method. Furthermore, we show that *AspectLink* is an effective alternative for product catalog enrichment in a representative set of categories.
- We describe our experience of processing almost 0.5 million reviews and over 20,000 products. We examine several features of a real scale dataset regarding the use of references to product attributes in typical reviews written by users, which is our main motivation for this work.

The rest of this paper is organized as follows: The “[Related work](#)” section discusses related work. The “[Concepts and terminology](#)” section addresses some fundamental concepts and terminology related to our work. The “[AspectLink](#)” section describes the *AspectLink* Algorithm we propose, and the “[Matching aspect expressions](#)” section details some important parts of this algorithm. The “[Experimental results](#)” section presents our experimental evaluation. Finally, the “[Conclusions and future work](#)” section presents our conclusions and insights for future work.

Related work

A number of recent researchers have exploited the idea of enriching databases with information available in online sources [7, 18, 19].

Mansuri and Sarawagi [18] propose to enrich an existing database with unstructured records taken from web sources. By matching information extracted from the input records and the data already in the database, their system inserts new data into the database to represent new entities and new relationships. The tasks of extraction and matching are based on models that are previously trained.

Yakout et al. [19] present a system called *InfoGather* for augmenting entities in a database with information

gathered from web tables. As the input data is already structured, no extraction is needed. Their ultimate goal is to supply new values for existing attributes or to supply new attributes with values for existing entities. For this, the authors propose a strategy that first identifies web tables that match a given target table to be augmented and then selects values in this table that can be used to supply values and attributes to the entities in the target table based on several similarity models.

While these proposals focus on factual information, the *Surveyor* system [7] aims at associating subjective properties expressed in opinions mined from the Web to entities represented in a knowledge base. Given a collection of annotated Web documents, the system first applies NLP tools to identify mentions to entities in their target knowledge base. Then, it extracts subjective properties and their sentiment polarity from the text. Finally, the system selects the dominant properties and associates them to the target entity.

In comparison with these works, we also aim at enriching databases with information available from online sources. In addition, we also face the challenge of extracting and matching information with elements from the database. However, as our scenario involves products in a catalog and opinions expressed in user reviews, we need to deal with several distinct requirements. First, we handle textual information, as in some approaches [7, 18], while the approach proposed by Yakout et al. [19] works with structured data. Second, we deal with subjective properties, as in the work of Trummer et al. [7], differently from other work [18, 19], which focus on factual properties. Finally, our method works at the attribute granularity, as in the work of [19], while Mansuri and Sarawagi [18] and Trummer et al. [7] work at the entity granularity.

In the opinion mining realm, there has been a recent increase interest in researching the problem of *review summarization*, also called as *opinion summarization*, which is a broad and diversified research topic. The most common type of opinion summarization technique is *aspect-based opinion summarization*. This technique generates opinion summaries around a set of aspects. These aspects are extracted from reviews, and then the sentiment towards each aspect is identified and summarized. In the simplest case, the summary is a presentation of the positive and the negative sentiments towards each aspect [8, 9, 20].

Hu and Liu [8] can be considered the pioneer on aspect-based opinion summarization. In their paper, the authors use an unsupervised itemset mining technique to identify product features that have been commented on by users. Then, their method decides whether each opinion sentence is positive or negative. The results of the summarization process show the number of opinion sentences considered as positive and negative for each feature.

In the work by Hu and Liu [9], the same authors propose using a supervised rule mining technique to generate language patterns to identify product features. By doing so, they address several linguistic problems that were not well treated by Hu and Liu [8]. Li et al. [20] have the same goal as Hu and Liu [8], but the former uses a conditional random fields (CRFs) model.

Summarizing opinions based on aspects is often the case that distinct aspect expressions refer to the same aspect *category* or *class*. Thus, the important task of *aspect grouping* has been addressed in the literature [10, 14, 15, 21]. In other approaches [15, 21], the authors proposed a constrained semi-supervised learning method to group aspect expressions into the user-specified aspect groups, where each group represents a specific aspect. The method starts from a small number of seed aspect expressions supplied by a user. It then assigns the remaining aspect expressions to suitable groups using an Expectation-Maximization (EM) algorithm based on labeled seeds and unlabeled examples. The method proposed by Carenini et al. [14] groups aspect expressions into nodes of a taxonomy, where each node represents a feature of products in some category. This taxonomy is supplied by a user. Except for this, the method is fully unsupervised since it relies on similarity functions to verify if aspect expression match features in the taxonomy. In this case, the aspect expression is mapped to the matching feature. The method described by Yu et al. [10] groups opinions according to their aspects according to a taxonomy. Unlike the work of Carenini et al. [14], this taxonomy is not related to a product category, rather to a specific product. It is initially built from information available in the product's Web page, and it is then incrementally rebuilt and refined according to the specific aspects found in a set of reviews on the target product. The method relies on a semantic distance learning algorithm, to group opinions based on their semantic relations, requiring training data. Similar to these methods, our *AspectLink* method relies on a strategy that groups aspect expressions into aspect classes. In our case however, these aspect classes are provided by the structure of the product catalog, while in the work of Carenini et al. [14], the taxonomy must be handcrafted by a user. *AspectLink* was designed to be unsupervised, and for this, we adapted and improved the similarity functions proposed by Carenini et al. [14]. This method is used as one of the baselines in our experiments.

Our work is similar to Yu et al. [10] in the sense that we use a set of reviews as input on each specific product and group aspect expressions identified in the reviews around features of this product. However, in *AspectLink*, the target features come from a product catalog; they are fixed and pre-defined for all products in a given category. In turn, in the approach proposed by Yu et al. [10], distinct

taxonomies can be generated for two products in the same category. In fact, depending on the set of reviews, and even on the order in which reviews are processed, the same product can lead to different taxonomies. For this reason, although the approach proposed by Yu et al. [10] is effective for the task of building product-oriented aspect taxonomies, it can hardly be used for the task of enriching product catalogs.

Concepts and terminology

This section reviews concepts and terminology we use through the paper. The definitions discussed here will be used in the next section that details our proposed approach, especially in Algorithms 1 and 2. In most cases, we rely on the definitions presented by Liu [13], with a few adaptations to better fit the problem we tackle.

Product catalog

In our work, we consider a catalog as a set of products in a given category (e.g., DVD players and digital cameras), where each product is represented by its attributes and their corresponding values. More formally, a product catalog is a set of products $C = \{p_1, \dots, p_n\}$, and each product from a given category is represented by a pair $p_i = \langle t, \mathcal{A} \rangle$, where t is the title of the product and $\mathcal{A} = \{A_1, \dots, A_m\}$ is a set of *attributes* that are common to all products in this category. For all products in a category, the same attribute A_m has a unique *name* N_{A_m} , which is used to refer to the attribute. For a given product p in the category, each attribute A_m has a value $V(p_i, A_m)$, which is a set that can have one or many elements, or it can be empty.

Figure 2 illustrates an example of a product whose title is "Apple Macbook Air Notebook," which can be found in a typical product catalog of the Laptop category. In this example, products in this category are represented using seven attributes: Processor, Screen, Price, Storage, Dimension, Battery, and Software. Notice that in this case, as in many other real life cases, some attributes can be split into sub-attributes. For instance, Storage is further divided into Capacity and Type. In our work, however, we only consider top-level attributes, since addressing them is sufficient for our purposes. Thus, the value of each attribute is a set that includes the values of all its sub-attributes. For instance, the value of Processor for this product is given by {"Intel"; "1.6 giga-hertz"; "Intel 5th Generation Core i5"}.

Reviews, opinions, and aspects

A *review* is a text posted by a user on an e-commerce Web site, usually reporting their experience with a specific product, which is the target *entity* of the review. Each review is composed of a set of *sentences* which can be factual or subjective. Sentences that express factual information are called *objective* sentences, while sentences that

Apple MacBook Air Notebook		
Attributes	Sub-attributes	Values
Processor	Brand	Intel
	Speed	1.6 giga-hertz
	Model	Intel 5th Generation Core i5
Screen	Resolution	1440 x 900
	Touch Screen	No
	Type	LED
	Size	13.3 inches
Price		849.99 dollars
Storage	Capacity	128 gigabytes
	Type	Solid State Drive
Dimension	Height	0.68 inches
	Width	12.8 inches
	Depth	8.94 inches
	Weight	2.96 inches
Battery		12 hours
Software		Mac OS X

Fig. 2 Example of a product from a typical catalog of the Laptop category

express some personal feelings or beliefs are called *subjective* or *opinionated* sentences. We are interested in the latter because they contain aspect expressions, that is, we are interested in sentences that represent the reviewer's *opinions* of a product. Sometimes a single sentence may have multiple opinions. For example, the sentence “The design is incredible, but display is a junk” has two different opinions: a positive opinion regarding the design and a negative opinion about the display.

An *aspect* is any reference made in an opinion to a particular part or characteristic of the product, or even to the product as a whole. The same aspect can be expressed using different *aspect expressions*. For instance, the sentences “The design is incredible” and “The body style is killer” refer to the same aspect—the design—using two different aspect expressions: *design* and *body style*.

An opinion expresses a *sentiment* of the reviewer towards an aspect of a product. A sentiment has a *polarity* which can be *positive*, *negative*, or *neutral*. The set of words used to express the sentiment are called *sentiment words* or *opinion words*. For example, “love,” “incredible,” and “best” indicate positive sentiments, while “junk” indicates a negative sentiment.

In the work of Liu [13], an opinion is a quintuple $o = \langle e, a, s, h, t \rangle$, where e is the target entity, a is the target aspect of entity e on which the opinion has been given, s is the sentiment polarity of the opinion towards aspect a of entity e , h is the opinion holder, and t is the opinion

posting time. In our work we adapt this definition to the problem setting we address. We are given a product p and a set of reviews R_p written about p . Typically, R_p is the set of reviews extracted from a landing page about p in an e-commerce site. Thus, the target entity e is assumed to always be p and we decide to omit this element from the representation of opinions. Furthermore, due to its informative nature, we opt to include the sentiment words in the opinion representation. Hence, from this point on, opinions are represented by a quintuple $o = \langle a, w, s, h, t \rangle$, where a , s , h , and t are the same as above, and w corresponds to the sentiment words of the opinion.

In Fig. 3, we present examples of reviews written by real users about the product in Fig. 2. Sentiment words are underlined, and words composing aspect expressions are bold faced since they are very important in our work. The symbols near the expressions will be explained latter.

Enriching product catalogs with opinions

In our work, we tackle the problem of enriching product catalogs with user opinions extracted from product reviews. Our main goal is to automatically map opinions to specific attributes from the product catalog that the opinions refer to. However, it is often the case that reviews also include opinions that do not refer to a specific attribute of a product, but to the product as a whole. Furthermore, opinions may also target attributes that are not represented in the product catalog. Thus, we consider these three cases in our work. Our general strategy consists of grouping opinions according to their aspect expressions. Specifically, each opinion is mapped to one or more *aspect classes*, according to the following three cases.

Case 1 The user posts an opinion referring to one of the attributes of the product catalog. For instance, in the sentence “The battery life is great,” the user is expressing a positive opinion (“great”) towards the battery. Therefore, we should map this opinion to the attribute “Battery” of the product. To handle cases like this, we create a distinct aspect class to represent each attribute of the product and then we map each opinion to the corresponding aspect class. In this simple example, we have a clear match between the aspect expression and the attribute name. However, there are many situations in which the match is not so obvious. In Table 1, we show seven aspect expressions extracted from Fig. 3 identified by the symbol ♠, along with the attributes that should be matched. Notice that matching “fast” or “Intel i5” to the *Processor* attribute is not obvious. Thus, as detailed in the “*AspectLink*” section, we leverage information taken from the product catalog to improve the

- a** By Doris B on May 15, 2016. After tons of research, including reading reviews like this, I finally committed to a MacBook Air. I'm in love with my new **MacBook Air**[◇]. The **design**[♣] is beyond incredible. The sleek **body style**[♣] is killer like any product applied product. It **runs**[♣] great and very **fast**[♣]. Has never crashed on me. The **battery life**[♣] is great. I like the fact that the base of **keyboard is metal**[♣]. I use it for everyday things like Facebook, YouTube, Netflix. Everything is perfect EXCEPT the **display**[♣]. I believe that this the best laptop[◇] out there today in every way.
- b** By B-Aron June 14, 2016. The **computer**[◇] is good, the newest version up to now, but just upon arrival, its price from the same seller dropped down about \$100!!! which I can hardly accept. its the first laptop with a great keyboard[♣]. The **size**[♣] and **weight**[♣] are great. **Intel i5**[♣] combined to **Mac OS**[♣] does the job.

Fig. 3 Examples of reviews of the product in Fig. 2

matching performance. In Fig. 4, the first seven lines show opinions extracted from the reviews of Fig. 3 that were mapped to aspect classes corresponding to the attributes of the product.

Case 2 The user posts an opinion on the product as a whole. In the sentence “I’m in love with my new MacBook Air” from Fig. 3, the user posts an opinion expressing a positive sentiment (“love”) for the product as a whole, and not for one of its attributes. Therefore, we should map this opinion to the target product. In our work, we map these kinds of opinions to an aspect class called *General*. In this example, this mapping is done because the aspect expression and the product title match. As in case 1, there are situations in which the mapping must be done according to other forms of matching. Examples of aspect expressions that should be associated to product as a whole are identified in Fig. 3 with the symbol ◇. Notice that the aspect expressions *laptop* and *computer* do not directly match the product title. Details on how the match is evaluated in cases like this will be discussed in the “*AspectLink*” section. This case is illustrated in Fig. 4, where we show opinions extracted from the

reviews of Fig. 3 that were mapped to the *General* aspect class.

Case 3 The user posts an opinion on a characteristic of a product that is not explicitly represented as an attribute in the product catalog. Consider the following example: “*The design is beyond incredible*”. In this sentence the user posts a positive opinion (“incredible”) for a characteristic of the product that is not represented as an attribute in the product catalog, that is, its design. To handle these cases we create an aspect class called *Other*. We assume case 3 whenever case 1 and case 2 do not hold. This decision is based on the assumption that all input reviews are related to the product and so are the opinions in them. Thus, if the opinion does not refer to an attribute of the catalog, or to the product itself, it must refer to some other characteristic of the product. This and other aspect expressions that are associated to *Other* are identified in Fig. 3 with the symbol ♣. This case is illustrated in Fig. 4 where we show opinions extracted from the reviews of Fig. 3 that were mapped to the *Other* aspect class.

More formally, given a product catalog C , our ultimate goal is to generate an *Enriched Catalog* C^+ as follows. For each product $p_i = \langle t, \mathcal{A} \rangle$ in C , there is an enriched representation $p_i^+ = \langle t, \mathcal{A}, \mathcal{S} \rangle$, where $\mathcal{S} = \mathcal{A} \cup \{General, Other\}$. Each element in \mathcal{S} is called an *aspect class*. For each aspect class $S \in \mathcal{S}$, we have a set of opinions $O_{p_i, S}$ as follows. When $S = A \in \mathcal{A}$, $O_{p_i, S}$ is the set of opinions referring to attribute A in p_i , that is, opinions that fall in to case 1 above. When $S = General$, $O_{p_i, S}$ is a set of opinions that refer to the product as a whole, that is, opinions that fall in case 2 above. Finally, when $S = Other$, $O_{p_i, S}$ corresponds to the set of opinions on a product characteristic that is not explicitly represented as an attribute in the product catalog, that is, opinions that fall in to case 3 above.

To accomplish the mapping task that leads to the generation of the enriched catalog, we rely on a strategy that tries to match the aspect expressions of the opinions to aspect classes. This strategy is implemented by a method

Table 1 Aspect expressions from reviews of Fig. 3 and the common attribute of the product catalog to which they are associated

Aspect expression	Attribute
“battery life”	Battery
“size”	Dimension
“weight”	Dimension
“display”	Screen
“runs”	Processor
“fast”	Processor
“Intel i5”	Processor
“Mac OS”	Software

Apple MacBook Air Notebook	
Aspect Class	Opinions
Processor	<"runs", "great", "positive", "May 15, 2016", "Doris B"> <"fast", "", "positive", "May 15, 2016", "Doris B"> <"Intel i5", "does the job", "positive", "June 14, 2016", "B-Aron">
Screen	<"display", "perfect except", "negative", "May 15, 2016", "Doris B">
Price	-
Storage	-
Dimension	<"size", "great", "positive", "June 14, 2016", "B-Aron"> <"weight", "great", "positive", "June 14, 2016", "B-Aron">
Battery	<"battery life", "great", "positive", "May 15, 2016", "Doris B">
Software	<"Mac OS", "does the job", "positive", "June 14, 2016", "B-Aron">
General	<"MacBook Air", "love", "positive", "May 15, 2016", "Doris B"> <"laptop", "best", "positive", "May 15, 2016", "Doris B"> <"computer", "good", "positive", "June 14, 2016", "B-Aron">
Other	<"design", "incredible", "positive", "May 15, 2016", "Doris B"> <"body style", "killer", "positive", "May 15, 2016", "Doris B"> <"keyboard is metal", "like", "positive", "May 15, 2016", "Doris B"> <"keyboard", "great", "positive", "June 14, 2016", "B-Aron">

Fig. 4 Example of product catalog enriched with opinions showed in Fig. 3

we call *AspectLink*. *AspectLink* is described in the next section.

AspectLink

Our method takes as input a product p and a set of reviews R_p written about p . Typically, R_p is the set of reviews extracted from a landing page about p in an e-commerce site. The main strategy we rely on in our method is based on the aspect expressions from opinions in the reviews.

To address case 1, our method tries to match each aspect expression with an attribute, more specifically with a *descriptor* of the attribute. Informally, a descriptor is a set of words that describe the attribute. This concept is precisely stated in Definition 1.

Definition 1 Let A be an attribute of the products in a product catalog C . We define $\Delta_{p,A} = \{N_A\} \cup V_{p,A}$ as a descriptor for A in a product p from C , where, as defined in the “Product catalog” section, N_A is a unique name used to refer to the attribute, and $V(p,A)$ is a the set of values of A in product p .

As it will be more clear later, the idea of including the attribute N_A and the values of the attribute $V_{p,A}$ together in the descriptor is to allow multiple ways of matching aspect expressions and attributes. For instance, the descriptor for *Software* attribute from product catalog illustrated in Fig. 2 is formed by the name of attribute (“Software”) and its value (“Mac OS X”). Thus, according to Definition 1, the descriptor for *Software* attribute is {“Software”, “Mac OS X”}.

We notice that in practice, we can apply some common pre-processing steps for handling sets of words when building descriptors. For instance, in our experiments we considered using a stemming function as alternative building descriptors because stemming is widely used in information retrieval systems with the aim of increasing recall [22].

A match between an aspect expression and an attribute is defined as follows.

Definition 2 Let A be an aspect class from an enriched catalog C^+ , created to represent an attribute A from a product catalog C , which has a descriptor $\Delta_{p,A}$. Let α be an aspect expression from an opinion o . We map o to A , if α matches $\Delta_{p,A}$. We say that α matches $\Delta_{p,A}$, if at least one of its words, say w , matches at least one word, say u from $\Delta_{p,A}$ according to one of the following similarity functions: *str_match*, *syn_score* or *sim_score*.

As an example, the opinion <“Mac OS,” “does the job,” “positive,” “June 14, 2016,” “B-Aron”> presented in Fig. 2 would be mapped to attribute *Software* because the aspect “Mac OS” from opinion matches at least one word with the descriptor for *Software* attribute. Recall that the descriptor for *Software* attribute is {“Software”, “Mac OS X”}.

The three similarity functions referred in Definition 2 will be detailed in the “Matching aspect expressions” section.

We handle case 2 in a similar fashion to case 1. However, in this case we use a different kind of descriptor

to represent products and we try to match each aspect expression with this descriptor. The descriptor concept is precisely stated in Definition 3.

Definition 3 Let p be a product in a catalog C . We define $\Delta_p = \{t\}$ as a **descriptor** for p , where, as defined in “Product catalog” section, t is the title used for this product.

In this case, the descriptor for the product (laptop) illustrated in Fig. 2 is {“Apple MacBook Air Notebook”}.

Definition 4 Consider the aspect class *General* from an enriched catalog C^+ , that represents a product from a catalog C , whose descriptor is Δ_p . Let α be an aspect expression from an opinion o . We map o to *General*, if α matches Δ_p . We say that α matches Δ_p , if at least one of its words, say w , matches at least one word, say u from Δ_p according to one of the following similarity functions: str_match , syn_score or sim_score .

For instance, the sentence “My laptop is amazing” on the product illustrated in Fig. 2 has an opinion that is mapped to *General* aspect class because there is a match between the descriptor Δ_p , whose value is “Apple MacBook Air Notebook”, and the aspect expression α , whose value is “laptop”. More specifically, there is a match between the words “Notebook” and “laptop”.

Finally, our method assumes case 3 whenever case 1 and case 2 do not hold.

Definition 5 Consider the aspect class *Other* from an enriched catalog C^+ , associated with a product p from a catalog C , for representing characteristics of the product that are not represented as an attribute in the product catalog. Let o be an opinion. We map o to *Other*, if o was not mapped to another aspect class, according to Definitions 2 and 4.

The AspectLink algorithm

Algorithm 1 presents a complete description of our method. The algorithm receives as parameters a product catalog C and a set of reviews R for the products from C , and returns an enriched catalog C^+ , where each of its p^+ product are formed by p added with opinions. Our algorithm iterates through the set of products in C (Loop 1–33), and for each product p_i , two sequential phases are performed. In the first phase (Lines 3–10), the algorithm generates, from a set of reviews R_i on p_i , a set of opinions O whose target is p_i . In the second phase (Lines 13–32), the algorithm maps each opinion $o \in O$ to an aspect class from the enriched version of p_i , p_i^+ . We describe the algorithm in detail in the following paragraphs.

Algorithm 1: AspectLink

Input: A product catalog C
Input: A set of reviews R
Output: An enriched product catalog C^+

```

1 foreach product  $p_i$  from  $C$  do
2   let  $p_i = \langle t, \mathcal{A} \rangle$ ;
3   let  $R_i$  be a set of reviews on  $p_i$ ;
4    $O \leftarrow \emptyset$ ;
5   foreach review  $r \in R_i$  do
6      $SS \leftarrow extractSubjSent(r)$ ;
7      $DS \leftarrow removeCompSent(SS)$ ;
8      $O_i \leftarrow$  opinions extracted from  $DS$ ;
9      $O \leftarrow O \cup O_i$ ;
10  end
11  let  $p_i^+ = \langle t, \mathcal{A}, \mathcal{S} \rangle$ , where  $\mathcal{S} = \mathcal{A} \cup \{General, Other\}$ ;
12  foreach  $S \in \mathcal{S}$  do  $O_{p_i, S} \leftarrow \emptyset$ ;
13  foreach opinion  $o \in O$  do
14    Matched  $\leftarrow$  FALSE;
15     $\alpha_o \leftarrow$  the aspect expression from  $o$ ;
16    foreach Attribute  $A \in \mathcal{A}$  do
17      let  $\Delta_{p_i, A}$  be a descriptor for  $A$ ;
18      if Match( $\alpha_o, \Delta_{p_i, A}$ ) then
19        add  $o$  to  $O_{p_i, A}$  ▷ Case 1
20        Matched  $\leftarrow$  TRUE;
21      end
22    end
23    let  $\Delta_{p_i}$  be a descriptor for product  $p_i$ ;
24    if Match( $\alpha_o, \Delta_{p_i}$ ) then
25      add  $o$  to  $O_{p_i, general}$  ▷ Case 2
26      Matched  $\leftarrow$  TRUE;
27    end
28    if Not Matched then
29      add  $o$  to  $O_{p_i, other}$  ▷ Case 3
30    end
31  end
32   $p_i^+ \leftarrow p_i$  with opinions  $O_{p_i, S}$ ;
33 end
34 return  $C^+$ 

```

In the first phase, our method starts by breaking down each review $r \in R_i$ into sentences. In Line 6, the function *extractSubjSent()* is used to extract the subjective sentences from each review r , since, as discussed in the “Reviews, opinions, and aspects” section, only these kinds of sentences contain opinions. To accomplish this, the function *extractSubjSent()* was implemented based on the method proposed by Qadir [23].

Next, in Line 7, we eliminate comparative sentences through the function *removeCompSent()*, that accepts the subjective sentences which were extracted in the previous step. This function was implemented based on the

method proposed by Liu [24]. This is done because sometimes users compare one product with another product, or one characteristic of one product with another. As our goal is to enrich each product of the catalog with the opinions of users regarding the specific product, we decided to eliminate comparative sentences, even if they are subjective. In our experiments, we notice that there are very few of these types of sentences in product reviews (0.01% of all subjective sentences). This is due to the fact that e-commerce site users focus on writing only about the product of interest, unlike what occurs, for instance, in forums where users usually write comments comparing products.

In Line 8, the algorithm extracts the opinions from the remaining subjective sentences. Recall from the “Reviews, opinions, and aspects” section that there can be more than one opinion per sentence. The extraction of opinions from sentences is carried out using standard methods from the literature. For instance, we can use techniques such as presented by Kim and Frasinca [25] to identify the polarities of the user opinions on a specific aspect class. In the case of aspect expressions, which in our method guide the mapping of opinions to aspect classes, we implemented the well-known unsupervised aspect extraction method described by Poria et al. [26]. Finally, the opinions extracted in the previous step are added to O (Line 9).

In the second phase, our method groups opinions $o \in O$ according to the aspect classes S of p_i . For this, each opinion o will be “stored” in a set of opinions $O_{p_i,S}$, where p_i is the product being processed and S is an aspect class in S . In Line 12, we create an empty set of opinions $O_{p_i,S}$ for each aspect class. For each opinion $o \in O$, we have three distinct strategies according to the cases defined in the “Enriching product catalogs with opinions” section. Our strategy for case 1 is implemented in Lines 16 to 22, where we attempt to map each opinion o to some aspect class corresponding to an attribute. We use the function *Match()* to verify if the aspect expression α_o of opinion o matches the descriptor $\Delta_{p_i,A}$ of attribute A (Line 18). If it matches, the opinion o is added to the set of opinions $O_{p_i,A}$ (Line 19). Notice that the algorithm does not interrupt the loop even if the function *Match()* returns TRUE. This happens because our method allows the same opinion to be mapped to more than one aspect class. Thus, we let the current iteration continue, so we can try to match the same aspect expression with descriptors of other attributes. We postpone the detailed description of the *Match()* function to “Matching aspect expressions” section.

Our strategy for case 2 is implemented in Lines 23 to 27, where we attempt to map each opinion o to the aspect class *General* that represents the product p as a whole. In Line 24, we use the function *Match()* to verify if the aspect

expression α_o of opinion o matches the descriptor Δ_{p_i} of the product. If it matches, opinion o is added to the set of opinions $O_{p_i,General}$ (Line 25).

The strategy for case 3 is quite simple. We consider that if there were no match in previous cases, then the opinion o will be added to the set of opinions $O_{p_i,Other}$ (Line 29).

In Line 32, the algorithm enriches current product p_i with opinions in $O_{p_i,S}$ and set to p^+ . This operation is performed for each p_i from C and finally, the algorithm produces an enriched catalog C^+ in Line 34.

Expanding attribute descriptors

To map opinions to aspect classes, our method relies heavily on matching aspect expressions and descriptors. According to “AspectLink” section, we use the descriptor $\Delta_{p,A}$ when we want to map an opinion to an aspect class corresponding to an attribute A , and we use the descriptor Δ_p when we want to map an opinion to the aspect class *General*. Both descriptors consist of words that come from the attributes of the product catalog C or from the title of p , respectively.

However, in results obtained from preliminary experiments, we noticed that the set of words used to represent the attributes or titles of products in the catalog may be sometimes incomplete. For instance, in the Laptop category many manufacturers only provide the name of the operating system in the attribute *Software*, while other manufacturers provide a complete list of the software that come installed on the laptop, such as applications, anti-virus, browsers, etc. In the sentence “McAfee is always able to protect my personal data”, there is a clear opinion regarding the *Software* attribute, but we need to know that “McAfee” is a kind of software. Therefore, the information that a product p_i has on its values for *Software* can be used for another product p_j .

Another common problem with data in product catalogs is that manufacturers and stores often represent products that should have the same name with slightly distinct titles. For instance, Apple laptops are presented in many different ways, such as “MacBook,” “Mac,” and “Mac Book.” Thus, words appearing in the title of a product p_i from a given category may be useful to describe another product p_j from the same category.

To cope with problems such as these, we also consider an expanded form of attribute descriptors in our work, as defined below.

Definition 6 Let A be an attribute of the products in a product catalog C , and let N_A be the name of the attribute, and let $V(p,A)$ be the set of values of A in product p . We define $\Delta_{*,A} = \{N_A\} \cup V_{p_1,A} \cup \dots \cup V_{p_n,A}$ as an **expanded descriptor** for A , where $\{p_1, \dots, p_n\}$ is the set of all products in C .

Definition 7 Let p be a product in a catalog C , and let t be a unique product title used to product p . We define $\Delta_{p^*} = t_1 \cup \dots \cup t_n$ as a **expanded descriptor** for p , where $\{t_1, \dots, t_n\}$ is the set of titles for all products in C .

To use these expanded descriptors, the only modifications required in Algorithm 1 are to replace $\Delta_{p,A}$ by $\Delta_{*,A}$ in Line 17 for case 1, and to replace Δ_p by Δ_{p^*} in Line 23 for case 2. However, we assume that before running the algorithm, all the expanded descriptors have been generated in a preprocessing step.

We carried out experiments with the two kinds of descriptors and noticed that the use of expanded descriptors led to higher recall values in all categories, with a comparatively small loss in precision. The details on these experiments are presented in “Common vs. expanded descriptors” section.

Matching aspect expressions

According to Definitions 2 and 4, three string similarity functions are combined to match aspect expressions with descriptors. In Algorithm 1, a function called *Match* encapsulates the three functions combined. This function is detailed in Algorithm 2.

Algorithm 2: Match Function

Input: An aspect expression α

Input: A descriptor Δ

```

1 let  $\Theta_1, \Theta_2$  and  $\Theta_3$  be global parameters;
2 foreach word  $w \in \alpha$  do
3   foreach word  $\delta \in \Delta$  do
4      $s_1 \leftarrow \text{max\_str\_match}(w, \delta)$ ;
5     if  $s_1 \geq \Theta_1$  then return TRUE;
6      $s_2 \leftarrow \text{max\_syn\_score}(w, \delta)$ ;
7     if  $s_2 \geq \Theta_2$  then return TRUE;
8      $s_3 \leftarrow \text{max\_sim\_score}(w, \delta)$ ;
9     if  $s_3 \geq \Theta_3$  then return TRUE;
10  end
11 end
12 return FALSE

```

Given an aspect expression α and a descriptor Δ , the algorithm iterates over all words w of α , and for each word δ of Δ , it computes a similarity score value between w and δ , using the three similarity functions. The algorithm terminates and returns TRUE, if one of the pairs w, δ gives a similarity score value higher or equal to some predefined global threshold value, for any of the three similarity functions. Otherwise, it terminates and returns FALSE. The threshold values Θ_1, Θ_2 and Θ_3 are predefined and they are global for all calls of the function.

The three similarity functions we use are adapted from the ones originally proposed by Carenini et al. [14]. In that paper, the authors consider that each product category has a taxonomy that represents the main features of products. They then use these functions to evaluate matching between aspect expressions and terms that identify the product features in the taxonomy. We argue that the product features in the taxonomy play the same semantic role as attributes of a product catalog. However, in our work we generalize the original strategy proposed by Carenini et al. [14] by matching aspect expressions not only to attribute titles, but also to attribute values of a given product. This is accomplished by the concept of attribute descriptors introduced in “AspectLink” section. In addition, we check for matches between aspect expressions and the target product as a whole. In this case, we rely on the concept of product descriptors, also introduced in “AspectLink” section. We further generalize the original strategy by using the concept of expanded descriptors, which also include information from all products in the catalog. As we will discuss in the “Experimental results” section, both generalized strategies led to improved results compared to the original strategy proposed by Carenini et al. [14].

In what follows, we describe the adapted versions of three similarity functions proposed by Carenini et al. [14].

Metric 1 This function consists of a simple comparison of a word of the aspect expression (w) with a word (δ) of the descriptor, as defined below:

$$\text{max_str_match}(w, \delta) = \begin{cases} 1, & \text{if } w = \delta \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Metric 2 This metric employs WordNet and the classification of words into lexical categories or part of speech (POS). In WordNet words are grouped into sets of cognitive synonyms called *synsets*. Polysemous words belong to more than one synset. This metric verifies whether two words appear in the same WordNet synset, given their POS. If any intersection occurs between the synsets of each word, the metric returns 1, otherwise the metric returns 0. This metric uses a function $\text{syns}(w)$, that returns all synsets to which the word w belongs, considering all senses for w .

$$\text{max_syn_score}(w, \delta) = \begin{cases} 1, & \text{if } \text{syns}(w) \cap \text{syns}(\delta) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Metric 3 This metric evaluates the degree of similarity between two words using information derived from a semantic network. We implemented the method proposed by Li et al. [27], which defines the similarity between two words as a combination of two functions

$\ell(\alpha, \delta)$ and $h(\alpha, \delta)$, where ℓ gives the length of the shortest path between two words in WordNet, and h gives the height of the lowest common ancestor of the words in WordNet.

$$\text{max_sim_score}(\alpha, \delta) = \ell(\alpha, \delta) \cdot h(\alpha, \delta) \quad (3)$$

Notice that function ℓ alone could be used as a similarity function. However, according to Li et al. [27], this function may be less accurate when applied to larger and more general semantic nets, such as WordNet. The reason for this is that words at upper layers of hierarchical semantic nets have more general concepts and less semantic similarity between words than words at lower layers. To address this drawback, the authors suggest that the result of ℓ must be adjusted by the function h , which uses hierarchical information. More details about this method can be found in the work of Li et al. [27].

The function $\text{max_sim_score}(\alpha, \delta)$ returns a normalized value between [0,1], according the suggestion presented by Li et al. [27].

Experimental results

In this section, we present an empirical evaluation of *AspectLink* for the task of enriching product catalogs with opinions extracted from reviews. In the following, we first report the results of an experiment we carried out to validate our method. This experiment used a sample dataset that we fully annotated to generate a golden standard. This allows us to evaluate the effectiveness of our method and to compare it to a representative baseline. Next, we present results we obtained by running our method over a large-scale real-world dataset. In addition, we examine several features of this dataset regarding the use of references to product attributes in typical reviews written by users, which is our main motivation for this work.

Experimental validation

In this section, we describe a set of experiments we carried out to validate *AspectLink* and to compare it to a representative baseline method. This validation uses a dataset, whose size allowed us to manually label all aspect expressions found in the reviews.

Setup

To evaluate our method, we used a dataset that is composed of a set of reviews and a product catalog taken from the BestBuy Web site¹. In this dataset, we call *BestBuy*, we took a number of products from four different categories, along with their attributes, to form the product catalog. Then, for each product, we randomly selected a set of reviews from those available for the product in the BestBuy Web site. Next, we manually identified each aspect expression found in these reviews. Finally, according to our terminology from the

“Enriching product catalogs with opinions” section, we labeled each aspect expression α with the following labels: *A*, if α corresponds to an attribute *A* from the product catalog; *General*, if α refers to the product as a whole; or *Other*, if α refers to the some other product characteristic that is not represented as an attribute in the product catalog.

A summary of the *BestBuy* dataset is presented in Table 2. The four categories considered are: cameras (CAM), dvd players (DVD), laptops (LAP) and routers (ROT).

Regarding the product catalog, recall from the “Product catalog” section that the value of each attribute was built as a set that includes the values of all its sub-attributes, including multivalued attributes.

Baseline method

In this experiment, we use the method proposed by Carenini et al. [14] to serve as a baseline for comparison. Recall that this method uses the original versions of the word similarity metrics, which we use in our work. This method requires the input of a taxonomy of product features for a particular category. Their purpose is to map each discovered aspect expression to a node in the taxonomy based on similarities. We implemented this method according to the paper, assuming that the product features in the taxonomy play the same semantic role as the attributes of a product catalog. As this method works by matching aspect expressions to attribute titles, we used the most significant titles in the catalog to ensure a fair comparison.

Evaluation metrics

We used the well-known *precision*, *recall*, and F_1 as evaluation metrics. Let A be the set of correct mappings of opinions to aspect classes, according to the golden standard, and let B be the set of mappings of opinions to aspect classes generated by the method being evaluated. We define precision (P), recall (R) and F_1 as:

$$P = \frac{|A \cap B|}{|B|} \quad R = \frac{|A \cap B|}{|A|} \quad F_1 = \frac{2 \times (P \times R)}{(P + R)}$$

With respect to the baseline, this method does not generate mappings to the aspect classes *General* and *Other*.

Table 2 Summary of the *BestBuy* dataset

Category	No. of products	No. of reviews	No. of sentences	No. of aspects
CAM	12	291	790	377
DVD	8	160	388	287
LAP	20	383	1135	713
ROT	10	210	614	317
Total	50	1044	2927	1694

Thus, to ensure a fair comparison in evaluating the baseline method, we consider in A only the mappings for aspect classes that correspond to attributes.

General results

Table 3 presents the results achieved by *AspectLink* and the baseline for the task of mapping opinions to aspect classes. With both methods, we experiment building descriptors with and without applying stemming functions. These functions were performed through the traditional Porter algorithm [28]. We use the symbol S to indicate when the method uses stemming.

Our approach achieved higher F_1 values in all categories compared to the baseline. As expected, this is mainly due to the high increase in recall values. On average, the recall values obtained by our method are almost three times higher than those obtained by the baseline. Interestingly, in the majority of the cases, our precision values are also higher. In a single case, the baseline achieved a higher precision, but with a very poor recall. These results indicate that using attribute values available in the product catalog decisively contributed to the improvement in recall. For instance, the sentence “The Intel i7 works flawlessly with all my application programs including PhotoShop” has an opinion whose aspect expression “Intel i7” refers to the processor of the laptop. Thus, we should map this opinion to an aspect class corresponding to the attribute `Processor`. We argue that our method could map it correctly because *AspectLink* uses the brand value of the `Processor` attribute available in the catalog. Using only the attribute names would not yield the correct mapping.

Another issue that we analyzed was the influence of using stemming functions in the method effectiveness. As demonstrated in Table 3, in general, using stemming functions helped to improve precision and recall for both methods, and just in a few cases we had a slight reduction in precision. Flores and Moreira [29] state that the goal of stemming is to increase recall and, in practice, it tends to reduce precision as a side effect. This undesirable effect did not occur in our experiments because when we use stemming in similarity functions the method returns a smaller amount of possible matches between an aspect and the descriptors when stemming is not used.

Table 3 Precision, recall, and F_1 for *AspectLink* and the baseline with and without using stemming

Method	CAM			DVD			LAP			ROT		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Baseline	0.63	0.17	0.26	0.75	0.46	0.57	0.64	0.34	0.44	0.41	0.33	0.37
<i>AspectLink</i>	0.87	0.82	0.84	0.89	0.82	0.86	0.75	0.57	0.65	0.56	0.49	0.52
Baseline ^S	0.59	0.13	0.21	0.83	0.28	0.41	0.87	0.28	0.43	1.00	0.27	0.42
<i>AspectLink</i> ^S	0.88	0.82	0.84	0.88	0.71	0.78	0.88	0.52	0.65	0.77	0.59	0.67

For example, the “range” aspect in the ROT category is mapped only to *coverage area* class when the method is using stemming, but this same aspect is mapped to *dimension* and *coverage area* classes when the method is not using stemming. Note that in this example the precision would be not decreased. In the case of *AspectLink*, in a single case, in the DVD category, the results obtained using stemming functions had a noticeable impact on recall, and as a consequence, on F_1 . In this particular case, this is due to the fact that users commonly use acronyms as aspect expressions, and the stemming functions are generally unable to handle acronyms properly.

Common vs. expanded descriptors

In the “[Concepts and terminology](#)” section, we discussed how product descriptors are built. Two options were considered: common descriptors, which use values of the attributes and the title of the target product only, and *expanded* descriptors, which use values of the attributes and titles of all the products in the catalog that are from the same category as the target product. Table 4 compares the results obtained by *AspectLink* using common and expanded descriptors.

Using expanded descriptors led to higher recall values in all categories, with a comparatively small loss in precision. As a consequence, F_1 values with expanded descriptor are higher or equal to those obtained with common descriptors. This experiment corroborates our motivation for considering expanded descriptors. As discussed in the “[Concepts and terminology](#)” section, by using this kind of descriptor, we enrich the representation of attributes or the product as a whole, approximating it from the attribute domain. This explains the increase in recall observed in Table 4. From this point on, expanded descriptors are used in the remaining experiments we describe.

Similarity functions

To better understand the results achieved with *AspectLink*, it is interesting to take a deeper look at each similarity function used in our method. Remember from the “[Matching aspect expressions](#)” section that our method applies the functions *str_match*, *syn_score* and *sim_score* in sequence. Initially, it uses function *str_match* to map the aspect expressions. Then, it uses function *syn_score* to map the aspect expressions which were not

Table 4 Analysis of the use of different descriptors by *AspectLink*

Method	CAM			DVD			LAP			ROT		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
<i>AspectLink</i> ^S _{common}	0.88	0.82	0.84	0.88	0.71	0.78	0.88	0.52	0.65	0.77	0.59	0.67
<i>AspectLink</i> ^S _{expanded}	0.83	0.85	0.84	0.88	0.72	0.79	0.84	0.63	0.72	0.74	0.62	0.68

mapped in the previous step. Finally, it uses function *sim_score* to map the aspect expressions that were not mapped by the two previous functions. We run a specific experiment to verify the cumulative effect of applying the function according to this sequence. The results are shown in Table 5. The precision decreases slightly, or remains the same, after each function is used but there is a significant gain in the recall in almost all categories. This demonstrates that combining various similarity functions has a positive impact on the overall performance.

In addition to this experiment, we also analyzed the performance of each similarity function individually in comparison to using all of them sequentially as in *AspectLink*. The results are presented in Fig. 5. Although *str_match* and *syn_score* achieved better values for precision individually, *AspectLink* achieved higher values for recall and F_1 in all categories. On average, our gains in F_1 over *syn_match*, *syn_score* and *sim_score* considered alone were about 0.06, 0.42, and 0.17, respectively. This demonstrates that no single similarity function has better results than *AspectLink* and our method is able to perform well in different categories.

Estimating parameter Θ_3

In the “Matching aspect expressions,” section the three similarity functions use global threshold parameters that determine when a given aspect expression α and descriptor Δ match according to the function. In the case of functions *str_match* and *syn_score*, their respective threshold values Θ_1 and Θ_2 must be equal to 1, since they only allow exact matches. In the case of function *sim_score*, we must have $0 < \Theta_3 \leq 1$. The experiments described so far all use $\Theta_3 = 0.5$. This value is the same as suggested by Carenini et al. [14]. To corroborate this choice, we perform experiments with different values of Θ_3 . The results are presented in Fig. 6, where we plot F_1 values obtained when varying Θ_3 from 0.1 to 0.9. As shown, $\Theta_3 = 0.5$ produces the best average result among four categories of products.

Discussion

The results obtained in this experiment indicate that *AspectLink* is effective for the task of mapping opinions to

aspect classes based on the aspect expressions that form the opinion. In particular, the results from Tables 3 and 4, and from Fig. 6, suggest that the configuration that uses *AspectLink* with stemming functions, expanded descriptors and parameter $\Theta_3 = 0.5$ can be used in practice to perform this task.

Large-scale experiment

In this section, we present the results of experiments carried out with a second dataset, *Amazon*, which is considerably larger than the *BestBuy* dataset used in experiments reported so far. Before describing the results we reached, we present a number of interesting features of this dataset, which we believe accurately represents the occurrence of aspect expressions in product reviews.

Setup

To form the second dataset, we call *Amazon*, we started with a large collection of about 142 million reviews previously crawled from the Amazon.com web site [30]², and selected all reviews from each of the four categories used in the previous dataset (CAM, DVD, LAP and ROT). As each review in this collection identifies the product it refers to, we were able to crawl the required data from each of these products to form the product catalog, that is, the attributes available for each product. A summary of the *Amazon* dataset is presented in Table 6.

Notice that this dataset is much larger than the *BestBuy* dataset, both in number of products and in number of reviews.

As discussed in the “Reviews, opinions, and aspects” section, from all sentences composing the reviews, our method requires only subjective and non-comparative sentences. The number of target sentences—those we considered for this experiment—is presented in Table 6 for each category. Overall, 45.68% of the sentences were considered as targets.

Figure 7 presents details on the classification we made for the sentences as subjective, comparative, and factual in each category. The fact that a large fraction of the sentences is subjective indicates that e-commerce Web sites, such as Amazon.com, are indeed useful as a source for enriching product catalogs with information taken from user opinions.

Even after filtering factual and comparative sentences, the number of sentences to be processed is still higher than 1.5 million. Since it would be unfeasible to manually annotate all aspects found this huge volume of sentences, our golden standard for this dataset is composed of the 100 most frequent aspect expressions found in the target sentences of each category. We argue that in a practical setting, handling a few top frequent aspect expressions is more valuable than showing every single aspect expression from a potentially huge list. This choice

Table 5 Results of similarity functions applied cumulatively

	CAM			DVD			LAP			ROT		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
<i>str_match</i>	0.86	0.83	0.84	1.00	0.54	0.70	0.92	0.55	0.69	0.88	0.40	0.55
<i>str_match</i> + <i>syn_score</i>	0.86	0.83	0.84	1.00	0.60	0.74	0.88	0.56	0.69	0.88	0.41	0.56
<i>str_match</i> + <i>syn_score</i> + <i>sim_score</i>	0.83	0.85	0.84	0.88	0.72	0.79	0.84	0.63	0.72	0.74	0.62	0.68

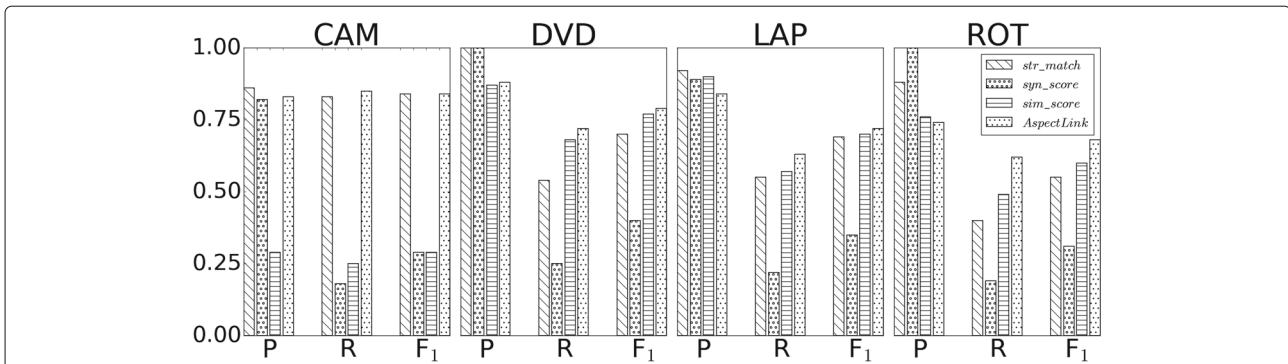


Fig. 5 Precision, recall and F₁ results comparing AspectLink to each similarity function applied individually

is further justified by experimental results we report later.

To select the 100 most frequent aspect expressions, we first run the aspect extraction method proposed by Hu and Liu [8]. This was implemented and all possible aspect expressions identified by this method were extracted. We then ranked these expressions according to their frequency. To assure that we only use true aspect expressions, we manually inspected and annotated the extracted expressions using the ranking order, and removed those that we did not consider aspect expressions. In the end, only the 100 most frequent true aspect expressions were kept for each category. The experiments reported here are based on a golden standard that only uses these 100 aspect expressions, instead of all aspect expressions found, as in the case of the *BestBuy* dataset.

To give an idea of the lists of aspect expressions obtained, Table 7 illustrates the ten most frequent aspect expressions extracted in each category, along with the aspect class to which they should lead in a correct mapping of opinions. From the results, it is quite apparent that the ten most frequent aspect expressions extracted are quite representative of each product category and more importantly, the results show which are the most commented aspects of each aspect class.

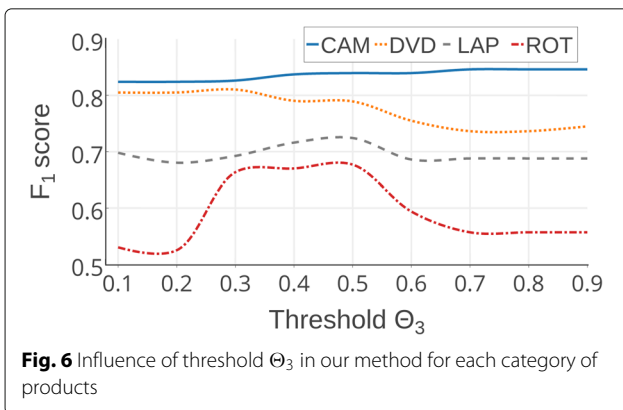


Fig. 6 Influence of threshold Θ_3 in our method for each category of products

Distribution of aspect expressions

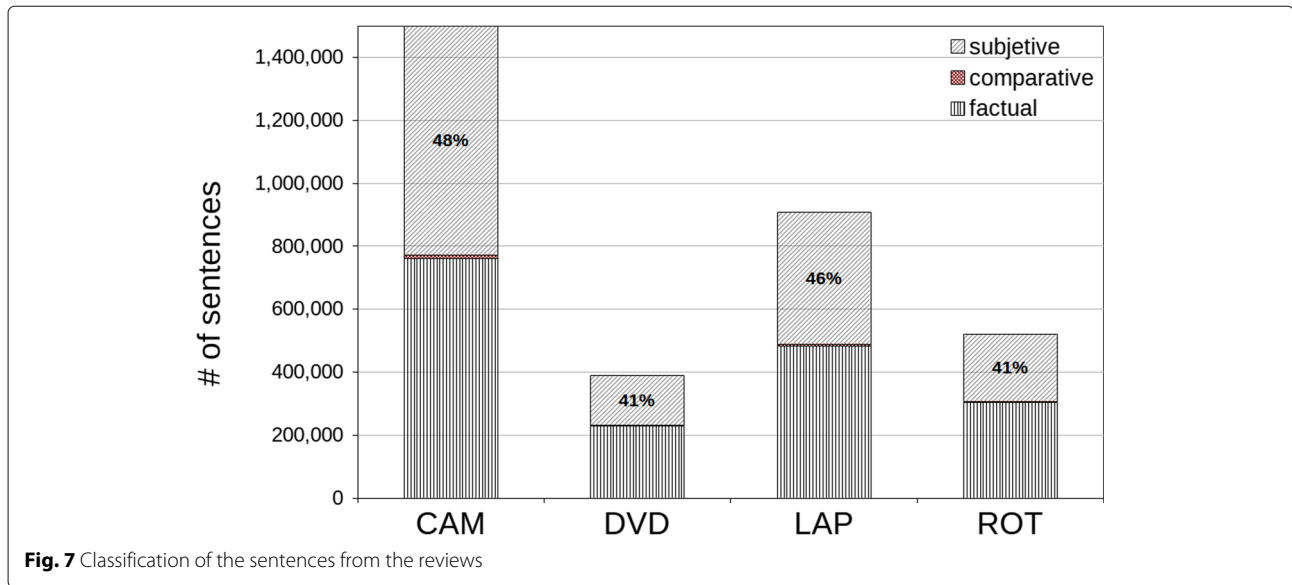
Figure 8 shows the distribution of the 100 most frequent aspect expressions from the *Amazon* dataset among the three kinds of aspect classes. Notice that for CAM, LAP, and ROT, the fraction of aspect expressions that represent the aspect class *Attribute* is higher than 50%, and for DVD, it is close to that percentage. This corroborates our assumption that reviews are viable as a source of lively knowledge to enrich the objective information available in product catalogs. In addition, it can be observed that a larger share of the aspect expressions refer to the aspect class *Other* than those in the catalog. For instance, in CAM and ROT, one quarter of the aspect expressions are in opinions which were mapped to the aspect class *Other*, and in DVD they account for 38%. An intriguing problem we left for future work is to further analyze cases such as these to look for specific latent aspect classes that, although not represented in the catalog by some attribute, are of interest for users. For instance, *keyboard* is the fifth most frequent aspect expression in LAP, but typically, there is no attribute referring to it in the product catalogs. In sum, Fig. 8 suggests that users comment more frequently on the specific characteristics of the products than on the product as a whole. This shows the relevance of properly addressing references to attributes in user reviews.

Distribution of sentences over kinds of aspect classes

Figure 9 summarizes the distribution of sentences among the three kinds of aspect classes (*Attributes*, *General* and

Table 6 Summary of the *Amazon* dataset

Category	Products	Reviews	Sentences	
			Total	Target
CAM	8839	204,127	1,499,405	726,388
DVD	2503	61,997	390,812	159,317
LAP	9491	115,521	907,031	417,278
ROT	1592	84,270	520,752	212,853
Total	22,375	464,871	3,318,000	1,515,836



Other). The number of sentences containing at least one of the 100 most frequent aspect expressions that form the opinions are mapped to a kind of aspect class. As explained in the “Reviews, opinions, and aspects” section, a single sentence may contain more than one opinion, and each opinion can be mapped to a different kind of aspect class. Thus, the sum of the percentage of all kinds of aspect classes may be greater than 100%.

Again we observe that most of the target sentences include aspect expressions that refer to attributes in the catalog. For instance, in CAM and LAP they account for more than half of the sentences. It is also of note that a

large share of sentences contain opinions referring to the product as a whole.

Distribution of sentences over aspect class referring to attributes

Figure 10 shows the distribution of sentences over aspect classes that correspond to attributes from the catalog for each category. In these graphs, each vertex in the polygon represents an attribute from the product catalog. The graph shows the percentage of sentences that contain an aspect expression that corresponds to a given attribute. For instance, 32% of the sentences that include at least one of the 100 most frequent aspect expression in the CAM category were mapped to the aspect class *Imaging*. In each graph, the attributes are placed in order clockwise from the most to the least frequently referred.

There are some attributes that are much more frequently referred to in reviews than others from the same category. For instance, in the ROT category, users comment four times more on *Frequency band* than on the *Price* of routers. This allows us to conclude that users are especially concerned with certain attributes of products in a category. Interestingly, in the four categories in this experiment, the price is not the most referred to attribute.

Diversity of aspect expressions over attributes

Figure 11 shows the distribution of aspect expressions extracted from the Amazon dataset over attributes from the product catalog in each category. In these graphs, we show the quantity of unique aspect expressions that refer to the same attribute. For instance, in the LAP category, we found fourteen different aspect expressions that refer to the aspect class *Software*. Analyzing the sentences,

Table 7 The 10 most frequent aspect expressions in reviews of each category from the Amazon dataset

CAM	DVD	LAP	ROT
camera (General)	dvd (General)	laptop (General)	router (General)
picture (Imaging)	player (General)	computer (General)	product (General)
feature (Other)	product (General)	windows (Software)	linksys (General)
canon (General)	picture (Video)	screen (Screen)	setup (Other)
photo (Imaging)	movie (Video)	keyboard (Other)	connection (Frequency Band)
video (Display)	price (Price)	price (Price)	range (Coverage Area)
lens (Zoom)	sony (General)	machine (Other)	wireless (Frequency Band)
battery (Power)	feature (Other)	hp (General)	feature (Other)
nikon (General)	disc (Format)	asus (General)	speed (Speed)
price (Price)	sound (Audio)	ram (Memory)	network (Frequency Band)

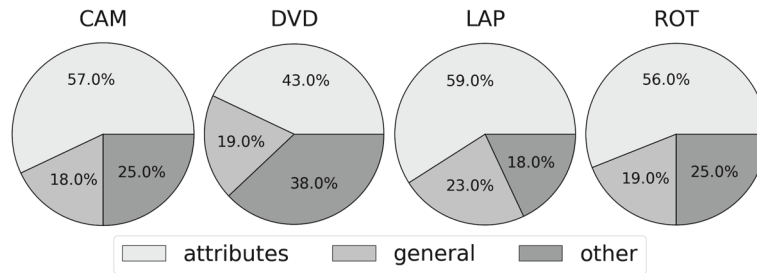


Fig. 8 Distribution of 100 most frequent aspect expressions among the three kinds of targets

we found that users do indeed employ several different terms such as “apps,” “system,” “application,” “vista,” and “program” to refer the aspect class *Software* in the LAP category.

Results of the experiments over the Amazon dataset

Table 8 presents the results achieved with *AspectLink* and the baseline for the task of mapping opinions to aspect classes, considering only the 100 most frequent aspect expressions extracted from *Amazon* dataset. For both methods we used the same setup described in the “Discussion” section. As was the case with the *BestBuy* dataset, our approach achieved higher Precision, Recall and F_1 values in all categories when we compared it to baseline.

We also notice that in all categories, precision values were above 0.8, with values very similar to those obtained for the *BestBuy* dataset. In the case of recall, the values obtained for the *Amazon* dataset are also similar to these achieved with the *BestBuy* dataset, except for the case of the ROT category, which is much lower in the *Amazon*

dataset. As a consequence, F_1 values are also slightly lower in comparison to those in the *BestBuy* dataset.

Conclusions and future work

In this paper, we presented a method to enrich product catalogs, which traditionally include only objective data provided by manufacturers or retailers, with subjective information extracted from reviews written by customers. Our motivation is the need users have to know other users’ impressions of specific product attributes, at the time they are making their purchase decisions. We claim that, while objective data on product attributes (e.g., the clock speed of a processor) is easy to obtain, subjective information on these attributes (e.g., what users think about the speed of this processor) are much harder to gather and to keep updated.

In our method, called *AspectLink*, attributes of products of a given category in a catalog are represented as aspect classes. Then, the problem of enriching product catalogs reduces to the task of mapping aspects extracted from user opinions to the corresponding attribute classes.

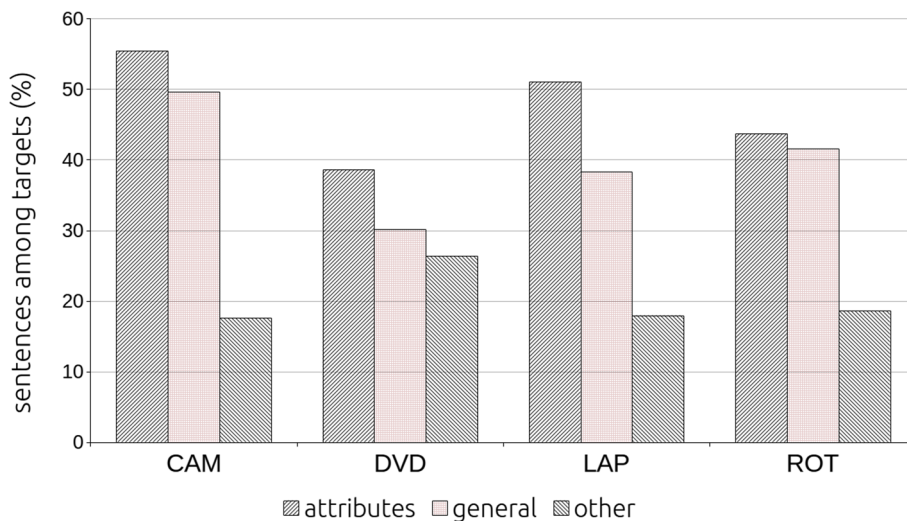


Fig. 9 Distribution of sentences among the kinds of aspect classes

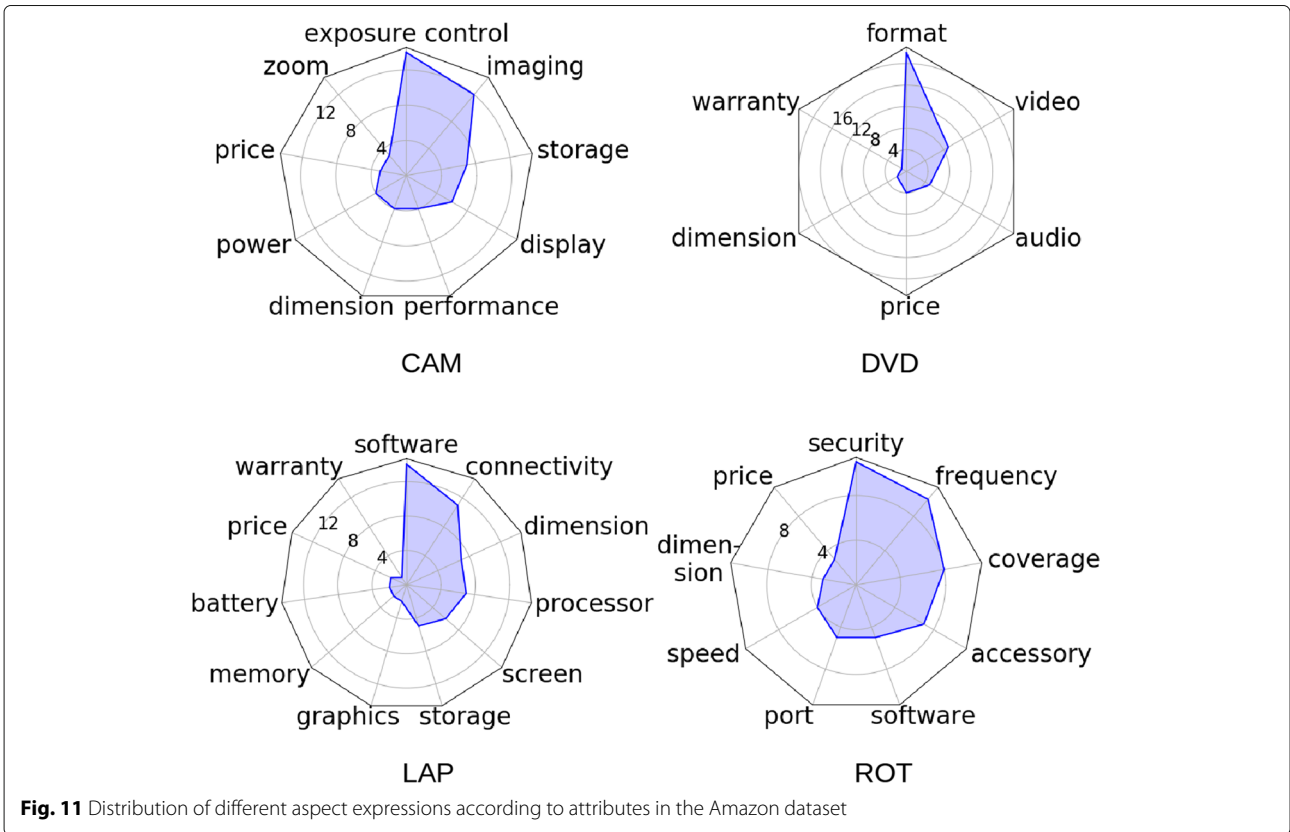
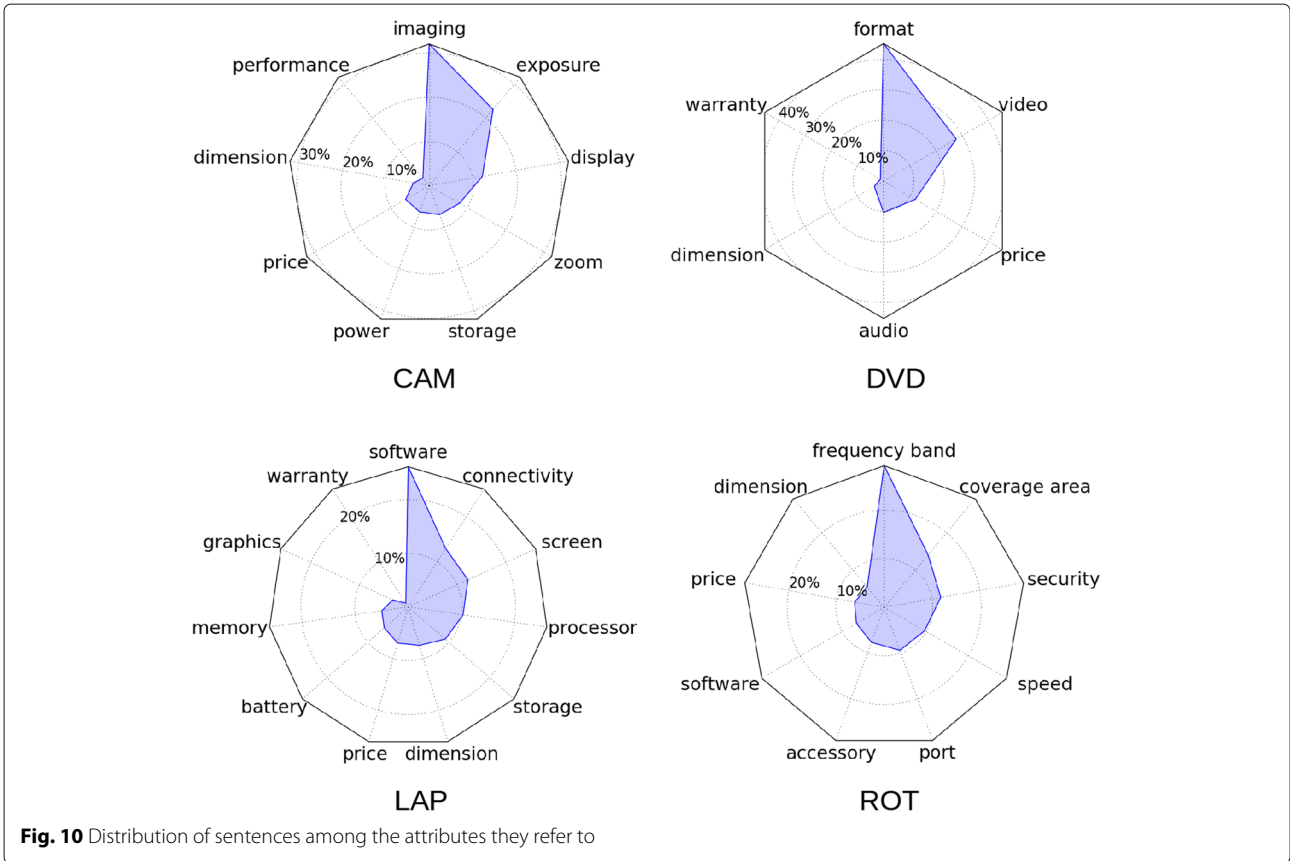


Table 8 Precision, recall and F_1 for *AspectLink* and the baseline in the *Amazon* dataset

Method	CAM			DVD			LAP			ROT		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Baseline	0.36	0.12	0.18	0.50	0.16	0.25	0.51	0.27	0.35	0.67	0.12	0.21
<i>AspectLink</i>	0.80	0.70	0.76	0.88	0.70	0.78	0.81	0.66	0.73	0.82	0.48	0.61

Our method carries out this task by means of similarity functions that compare lexical features of attributes and products from the catalog with features from the text of the user's opinions.

We have extensively evaluated our method comparing it against a baseline and also analyzing the impact of several parameters on the effectiveness of our method. We conduct experiments in four different electronic product categories, using two datasets with different scales. Our experimental results indicate that using *descriptors* in our method made *AspectLink* superior to the baseline in all metrics used. Our results also show that using *expanded descriptors* led to higher recall values in all categories, with a comparatively small loss in precision and, consequently, made the values for F_1 higher or equal to those obtained with common descriptors.

The results we have reached so far are already quite good, however, in future work, we plan to investigate the use of semi-supervised and supervised learning to further improve them. We also plan to test *AspectLink* in other domains, such as hotels and restaurants. Our results in this paper show that users comment a lot on the attributes that already exist in electronic product catalogs. However, in the domain of hotels, for instance, there is no representative catalog with the attributes that are commented on by the users. Therefore, a significant portion of aspect expressions could be mapped to the *Other* aspect class. In addition, we plan to study whether the opinions that were mapped to the *Other* aspect class could be clustered to later transform the most relevant clusters as new attributes in a database that represents the target entity. For instance, "keyboard" is the fifth most frequent aspect expression in the LAP category, yet there is no attribute representing it in product catalog commonly provided by e-commerce web stores. Considering that the most important attributes of the products are usually represented in the catalog, this analysis could help manufacturers and retailers find out what other product attributes are well commented on the web and are not represented in the catalog of their products.

Endnotes

¹ <https://developer.bestbuy.com>

² Available at <http://jmcauley.ucsd.edu/data/amazon>

Abbreviations

CAM: Cameras; CRF: Conditional random fields; DVD: DVD players; LAP: Laptops; POS: Part of speech; ROT: Routers

Acknowledgements

Work partially funded by projects eSpot (CNPq 461231/2014-0), SocSens (CAPES/PCGI 88887.130299/2017-01), CARECO (PROCAD/CAPES 88881.068507/2014-01), ATMOSPHERE (EC/H2020 grant no. 777154 & RNP/MCTIC acordo 51119), and by authors' individual grants from CNPq. The authors are grateful to the University of the State of Amazonas for the support to carry out this work.

Funding

Funding information is not applicable for this paper.

Availability of data and materials

The datasets used in experiments of this article are available at webpage: <https://goo.gl/uZQJjb>.

Authors' contributions

All authors have contributed to the methodological and experimental aspects of the research. The authors have also read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Institute of Computing, Federal University of Amazonas, Manaus, AM, Brazil.
²School of Technology, Amazonas State University, Manaus, AM, Brazil.

Received: 16 January 2018 Accepted: 15 October 2018

Published online: 28 November 2018

References

- Park DH, Kim S (2008) The effects of consumer knowledge on message processing of electronic word-of-mouth via online consumer reviews. *Electron Commer Res Appl* 7(4):399–410
- Racherla P, Friske W (2012) Perceived 'usefulness' of online consumer reviews: an exploratory investigation across three services categories. *Electron Commer Res Appl* 11(6):548–559
- Yan Q, Wu S, Wang L, Wu P, Chen H, Wei G (2016) E-wom from e-commerce websites and social media. *Electron Commer Res Appl* 17(C):62–73
- Penn M (2009) New Info Shoppers. <http://goo.gl/NfVZmc>. Accessed 08 Aug 2016
- Price Water House (2016) They say they want a revolution - Total Retail 2016. Available: <https://www.pwc.es/es/publicaciones/retail-y-consumo/assets/total-retail-2016.pdf>
- Ong T, Mannino M, Gregg D (2014) Linguistic characteristics of skill reviews. *Electron Commer Res Appl* 13(2):69–78
- Trummer I, Halevy A, Lee H, Sarawagi S, Gupta R (2015) Mining subjective properties on the web. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15). ACM, New York. pp 1745–1760. <https://doi.org/10.1145/2723372.2750548>
- Hu M, Liu B (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. pp 168–177. <https://dx.doi.org/10.1145/1014052.1014073>
- Liu B, Cheng J (2005) Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th international conference on World Wide Web (WWW '05). ACM, New York. pp 342–351. <https://doi.org/10.1145/1060745.1060797>
- Yu J, Zha ZJ, Wang M, Wang K, Chua TS (2011) Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (EMNLP '11), Stroudsburg. pp 140–150

11. Mukherjee A, Liu B (2012) Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (ACL '12), vol. 1. Association for Computational Linguistics, Stroudsburg. pp 339–348
12. Huang SL, Cheng WC (2015) Discovering Chinese sentence patterns for feature-based opinion summarization. *Electron Commer Res Appl* 14(6):582–591
13. Liu B (2015) *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge University Press, New York
14. Carenini G, Ng RT, Zwart E (2005) Extracting knowledge from evaluative text. In: Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP '05). ACM, New York. pp 11–18. <https://dx.doi.org/10.1145/1088622.1088626>
15. Zhai Z, Liu B, Xu H, Jia P (2010) Grouping product features using semi-supervised learning with soft-constraints. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg. pp 1272–1280
16. Carenini G, Cheung JCK, Pauls A (2013) Multi-document summarization of evaluative text. *Comput Intell* 29(4):545–576
17. Zhai Z, Liu B, Xu H, Jia P (2011) Clustering product features for opinion mining. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11). ACM, New York. pp 347–354
18. Mansuri IR, Sarawagi S (2006) Integrating unstructured data into relational databases. In: 22nd International Conference on Data Engineering (ICDE'06). IEEE, Atlanta. pp 29–29. <https://doi.org/10.1109/ICDE.2006.83>
19. Yakout M, Ganjam K, Chakrabarti K, Chaudhuri S (2012) Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. ACM, New York. pp 97–108. <https://doi.org/10.1145/2213836.2213848>
20. Li F, Han C, Huang M, Zhu X, Xia YJ, Zhang S, Yu H (2010) Structure-aware review mining and summarization. In: Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics (ACL), Stroudsburg. pp 653–661
21. Zhai Z, Liu B, Xu H, Jia P (2011) Constrained Ilda for grouping product features in opinion mining. In: *Advances in Knowledge Discovery and Data Mining (PAKDD 2011)*. Lecture Notes in Computer Science, vol 6634. Springer, Berlin. pp 448–459. https://doi.org/10.1007/978-3-642-20841-6_37
22. Baeza-Yates RA, Ribeiro-Neto B (2011) *Modern Information Retrieval*. 2nd ed. Addison-Wesley Longman Publishing Co., USA
23. Qadir A (2009) Detecting opinion sentences specific to product features in customer reviews using typed dependency relations. In: Proceedings of the Workshop on Events in Emerging Text Types (eETTs '09). Association for Computational Linguistics, Stroudsburg. pp 38–43
24. Liu B (2010) Sentiment analysis and subjectivity. *Handb Natl Lang Process* 2:627–666
25. Schouten K, Frasincar F (2016) Survey on aspect-level sentiment analysis. In: *IEEE Transactions on Knowledge and Data Engineering (TKDE'16)*, volume 28, issue 3. pp 813–830. <https://doi.org/10.1109/TKDE.2015.2485209>
26. Poria S, Cambria E, Ku LW, Gui C, Gelbukh A (2014) A rule-based approach to aspect extraction from product reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP). Association for Computational Linguistics and Dublin City University, Dublin. pp 28–37
27. Li Y, McLean D, Bandar ZA, O'shea JD, Crockett K (2006) Sentence similarity based on semantic nets and corpus statistics. *IEEE Trans Knowl Data Eng* 18(8):1138–1150
28. Porter M (1980) An algorithm for suffix stripping. *Program* 14(3):130–137
29. Flores FN, Moreira VP (2016) Assessing the impact of stemming accuracy on information retrieval—a multilingual perspective. *Inf Process Manag* 52(5):840–854
30. McAuley J, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). ACM, New York. pp 785–794. <https://doi.org/10.1145/2783258.2783381>

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
