Knowl Inf Syst
DOI 10.1007/s10115-017-1072-y

REGULAR PAPER

Learning patterns for discovering domain-oriented opinion words

Pantelis Agathangelou[1] · Ioannis Katakis[1] · Ioannis Koutoulakis[1] · Fotis Kokkoras[2] · Dimitrios Gunopulos[1]

Received: 23 May 2016 / Revised: 29 April 2017 / Accepted: 29 May 2017

Abstract Sentiment analysis is a challenging task that attracted increasing interest during the last years. The availability of online data along with the business interest to keep up with consumer feedback generates a constant demand for online analysis of user-generated content. A key role to this task plays the utilization of domain-specific lexicons of opinion words that enables algorithms to classify short snippets of text into sentiment classes (positive, negative). This process is known as dictionary-based sentiment analysis. The related work tends to solve this lexicon identification problem by either exploiting a corpus and a thesaurus or by manually defining a set of patterns that will extract opinion words. In this work, we propose an unsupervised approach for discovering patterns that will extract domain-specific dictionary. Our approach (DidaxTo) utilizes opinion modifiers, sentiment consistency theories, polarity assignment graphs and pattern similarity metrics. The outcome is compared against lexicons extracted by the state-of-the-art approaches on a sentiment analysis task. Experiments on user reviews coming from a diverse set of products demonstrate the utility of the proposed method. An implementation of the proposed approach in an easy to use application for extracting opinion words from any domain and evaluate their quality is also presented.

✉ Ioannis Katakis
katak@di.uoa.gr

Pantelis Agathangelou
pandelisagathangelou@gmail.com

Ioannis Koutoulakis
ikout@di.uoa.gr

Fotis Kokkoras
fkokkoras@teilar.gr

Dimitrios Gunopulos
dg@di.uoa.gr

1 University of Athens, Athens, Greece

2 Technological Educational Institute of Thessaly, Larissa, Greece

Published online: 14 June 2017

1 Introduction

Sentiment analysis is the process of extracting valuable information from user-generated opinionated content. This content may exist in various online sources and formats like product reviews, discussion forums or social networks. As the amount of online content grows, the importance of tools that automatically analyze such information sources becomes a necessity. Sentiment analysis lies in this category of tools, and it has high commercial value.

In this area, many approaches have been developed. The dictionary-based approach is a representative example. In this approach, there is an opinion word lexicon which consists of two opinion word classes. Positive words such as "*good, handy, impressive*" followed by negative like "*bad, useless, trash*". Such generic lexicons are already available but, as expected, fall short in adapting to data coming from diverse domains. This requires the utilization of *domain-specific* dictionaries of opinion words.

Currently, there are three main lines of research for tackling the task of opinion word extraction: (a) semantic thesaurus-based, (b) corpus-based and (c) pattern-based methods (for more details, see Sect. 2). These techniques although successful in some cases, they lack in some critical points. The first group relies on an information-rich semantic thesaurus that is not available for every language and domain. The disadvantage of the second group is that it captures only shallow dependencies. Finally, the third group (pattern-based methods) is using mostly predefined patterns that have limited applicability. Moreover, in most cases, these patterns are used solely for extracting opinion words and are not utilized in sentiment analysis.

Alleviating the disadvantages of the above methods, our approach (DidaxTo) dynamically explores the pattern space by analyzing the corpus in an unsupervised manner. Moreover, the extracted patterns have the following advantages: (a) they are corpus and domain specific and (b) they can be utilized directly in sentiment analysis.

We evaluate the proposed approach and compare it against the state-of-the-art with two complementary evaluation processes: First, we compare the classification ability of the extracted patterns against a method that utilizes predefined patterns (`NiosTo`) [1]. The evaluation is based on a sentiment analysis task applied on product reviews. We collect the ground truth sentiment (positive, negative) of each review by automatically extracting the users' rating (i.e. stars given to the product).

In the second process, we exploit these patterns to extract domain-specific lexicons and evaluate them against lexicons provided by the method that uses fixed patterns (`NiosTo`) [1] and a state-of-the-art approach [35] on a set of diverse domains of product reviews. Then, we let a set of human annotators evaluate all three groups of lexicons in terms of sentiment accuracy and domain specificity.

The *contribution* of this work can be summarized in the following points:

- Provides an unsupervised, multistage, resource efficient pattern extraction method. The extracted patterns can be utilized in extracting domain-specific opinion words and in sentiment classification. Our comparative study demonstrates that our approach outperforms two state-of-the-art methods in extracting opinion words and sentiment analysis.
- Offers a software (`DidaxTo`[1]) that implements and integrates the above method in an easy to use graphical user interface.
- Provides a dataset of opinions in the English language (product reviews). The dataset can serve as a benchmark in a variety of opinion mining tasks.

[1] http://deixto.com/didaxto.

This work takes advantage of our previous efforts in the field. In particular, NiosTo [1] is a methodology that uses a fixed set of patterns (conjunction-based, double propagation-based, etc.) in order to discover opinion words. On the other hand, in the approach introduced in this paper (DidaxTo) the patterns that are utilized are learned from the corpus *automatically* (see Sect. 3). Another difference is that DidaxTo applies the double propagation process (Sect. 3.5) both for opinion words and opinion targets. This results in improved accuracy when extracting domain-specific opinion words (see Sect. 5). More specifically, DidaxTo has higher precision in extracting opinion words (see Sect. 6). On top of that, DidaxTo utilizes a more elaborate method for polarity assignment, whereas NiosTo follows the polarity of the fixed patterns. This provides DidaxTo an advantage in sentiment classification (Sect. 5.3). Finally, the DidaxTo GUI is extended and implements a lot of additional features (Sect. 7).

The rest of the document is structured as follows. Section 2 summarizes the related work. In Sect. 3, we outline our approach and provide detailed description of all steps. Next Section (Sect. 4) provides details regarding patterns evaluation. Our experimental evaluation is presented in Sect. 5 followed by results and discussion (Sect. 6). The reader can learn about the basic feature-set of the `DidaxTo` software at Sect. 7, while the last section highlights significant conclusions and suggests future work.

2 Related work

In recent years, many research studies focus on the problem of sentiment lexicon construction. Most of them utilize some opinion seed words and word similarities to construct the sentiment lexicon. According to the way in which the word similarities are obtained, these studies can be categorized into three types of approaches: (a) the semantic thesaurus-based approaches, (b) the corpus-based approaches and (c) the pattern-based approaches. Below we discuss representatives from each category and in parallel discuss their main disadvantages.

2.1 Semantic thesaurus-based approaches

Some studies have proposed to utilize existing semantic thesauri for sentiment lexicon construction, like Word-Net and General Inquirer for English, Hownet for Chinese. This category of studies mainly depends on the synonym and antonym relationship among sentiment terms and the lists of related words in thesauri to expand the polarity lexicon from a sentiment word seed list. Hu and Liu [10] used adjective synonym and antonym lists in Word-Net to predict the semantic orientations for adjectives. Kamps et al. [11] built a polarity lexicon by linking synonyms provided by thesauri, and the sentiment polarity was defined by the distance from seed words ("good" and "bad"). Esuli et al. [3,6,7] proposed to use the synsets in Word-Net as the sentiment words, and annotated their positive, neutral and negative polarity scores. The annotation process was divided into a semi-supervised learning stage and a random walking learning stage. Esuli and Sebastiani proposed the inverse and bidirectional model of random walking algorithm in [6], where they also demonstrate the improvements over the work done in [7]. The above methods rely on the assumption that adjectives tend to have the same polarities with their synonyms and opposite polarities with their antonyms.

The first disadvantage of these set of approaches is that they totally rely on prior rich and complicated resources and thus cannot be applied to languages where such resources are not available. The second disadvantage is that they do not consider domain-dependent characteristics of sentiment lexicons, such as polarity variations.

可以做一个hotel review的
corpus

2.2 Corpus-based approaches

The corpus-based approaches are built on the assumption that there is co-occurrence of sentiment words that share same polarity. One of the most representative papers in this field is that of Turney [30], that focuses on learning polarities from a corpus. The adjective and adverb phrases were firstly extracted as candidate sentiment terms using pattern rules and their polarity was determined based on the co-occurrence of words like "excellent" or "poor" (which are considered seed words). The co-occurrence was measured by the number of hits returned by a search engine. Hatzivassiloglou and McKeown [9] approach relies on an analysis of textual corpora that correlates linguistic features, or indicators, with semantic orientation. No direct indicators of positive or negative semantic orientation have been proposed but demonstrated that conjunctions between adjectives provide indirect information about orientation. Popescu and Etzioni [23] introduce an unsupervised, high-precision information extraction system which mines product reviews in order to build a model of product features and their evaluation by reviewers. The method iteratively assigns polarity to words by using various features including intra-sentential co-occurrence and synonyms/antonyms of a thesaurus. Wilson et al. [33] proposed a supervised two-stage classification algorithm and analyzed the feature sets that affect polarity classification. Lau et al. [16], Qiu et al. [24,25], Weichselbraun et al. [32] manually analyzed and summarized eight dependency rules between opinion words and opinion targets, and proposed a double propagation algorithm to expand the opinionated targets and sentiment lexicon iteratively. Broß and Ehrig [4] and Lu et al. [21] create a domain and opinion target-specific polarity lexicon, creating a tree-like structure based on the attributes of the examined product. The word polarities are calculated via the use of a soft constraint [21] and the partially supervised word alignment method [4]. Amiri and Chua [2] and Xu et al. [35] treated the problem of detecting polarity of words as a semi-supervised label propagation problem on the graph, and prune the final lexicon with random walking on that graph [2,22]. Also [2] argues that the lexicon can contain both slang and misspelled words, which is one fundamental advantage over the semantic thesaurus-based approaches.

Corpus-based approaches like the ones mentioned above are intuitive, but they are quite limited since they capture only one type of simple relationship (co-occurrence). Our method captures more complicated syntactic relationships that form the extraction patterns. These patterns can be utilized not only for discovering opinion words but also for sentiment classification.

2.3 Pattern-based approaches

In this section, we review techniques that utilize patterns in order to extract opinion words. Note that there is an overlap with corpus-based approaches described in the previous section since there are techniques that analyze a corpus in order to discover such patterns.

Turney [30], for example, uses part of speech (pos) tagging in each review sentence. Some predefined patterns like adjective–noun and adjective–adjective are considered, and after that, for each extracted phrase, the semantic orientation is inferred by comparing its similarity to a positive reference word (e.g. "excellent") and a negative reference word (e.g. "poor"). In [9], the focus is on syntactic patterns. Adjectives are considered opinion words, and word orientations are calculated via the use of conjunctions links and with the help of some adjectives from the corpus with already known polarities. Then, the lexicon is expanded with the use of antonym–synonym relations. In [23], Word-Net relationships and morphological cues (e.g. suffixes) are being used, noun phrases are extracted from already defined patterns and some domain-independent rules for the extraction of potential opinion phrases are introduced. In

[33], the polarity of words from predefined lexicons is taken into account and with the help of morphological cues, adverbs and adjectives are being searched within each sentence and within the previous and next sentences. Qiu et al. [24,25] exploit direct and indirect relations between sentiment words and topics (or product features). Adjectives are considered opinion words, nouns opinion targets, while specific patterns are introduced. Moreover, in [24] extracted sentiment words and product features are used to extract new sentiment words. In [16], domain-independent opinion words are used to statistically extract domain-dependent opinion words, extending the Point-wise Mutual Information (PMI) measure. In [32], the bag of words approach is used. The authors consider negation by scanning for negation triggers such as "not" and "without". After that, classification methods are used to create the final lexicon. Lu et al. [21] use the Stanford parser while [35] uses Minipar to create dependency trees and to exploit sentence structure and relations. Morphological cues and negation rules are being utilized, and an already created general purpose lexicon is considered to estimate the opinion word polarity. Next, in [2], a seed list is created using general purpose opinion lexicons such as SentiWordNet. Linguistic rules such as affixes "dis" and "mis" for opposite polarities are used to expand that seed list. On top of that, Word-Net's first layer of synonyms and antonyms for the already obtained seed words is added to the seed list, while considering negations and disjunctive clauses. The PMI is calculated to find the words in the corpus related to the above-mentioned seed list, and a polarity graph is created for a final word refinement. Broß and Ehrig [4] identify nine high-precision patterns with adjectives exclusively to be considered as opinion words. Rao et al. [26] create a matrix with rows representing unique words from corpus and columns representing one of seven predefined emotions. Each cell of the matrix contains the probability a word portrays in that specific emotion. Peng and Park [22] expand seed word lists via Word-Net, uses pos-tagging to extract all adjective pairs linked with "and" and "but", and correlate them with the lexicon created by the seed word list and Word-Net. Liu et al. [20] study syntax-based methods that exploit syntactic patterns to extract opinion targets and alignment-based methods that use the word alignment model. A comparison of both approaches is presented. In the evaluation, the size of the corpus is being considered. Finally, our previous work [1] falls into this category. NiosTo, through a fixed set of patterns, mainly double propagation patterns, identifies domain-specific opinion words. Its effectiveness is evaluated on a sentiment analysis tasks, and the results demonstrate its sentiment analysis capabilities in multiple domains. The differences between NiosTo and DidaxTo are highlighted in detail in the previous section.

The main disadvantages of this set of approaches are that they either utilize a predefined set of patterns or target at a specific set of patterns (negation patterns). Moreover, in most cases the discovered patterns are not utilized in sentiment analysis but only in the extraction of opinion words. We argue and demonstrate experimentally that pattern-based sentiment analysis is beneficial since it captures complex sentiment dependencies in the use of language. For this reason, we develop a mechanism that assigns polarity for each pattern our method discovers. This enables the utilization of our patterns directly in sentiment analysis without the need of extracting opinion words first.

2.4 Related problems

In this section, we present a set of problems that are related in multiple ways with the task under study. For each problem category, we provide a brief overview of representative works in the context of opinion mining.

Text and opinion summarization In the task of opinion word extraction, we discover representative terms that users utilize while expressing opinions for items of specific domains. In a sense, the task has a similar effect with *text summarization* since, it identifies, representative positive and negative opinion words. Moreover, our approach identifies opinion targets which can be also used for summarization (see Sect. 8). In [8], the authors present a method for contrastive opinion summarization focusing on opinion about controversial issues like political discussions. The method mostly relies on a PLSA (Probabilistic Latent Semantic Analysis) model. Moreover, Kim et al. [13] provide a technique that selects texts that are explanations of why an opinion holder has a particular polarity of sentiment about an entity. Hidden Markov models are exploited for the solution of the problem. In a similar context, Kim et al. [14] provide a method for scoring the "explanatoriness" of a sentence. Three methods are explored that use tf-idf weighting and probabilistic modeling. Our approach is not directly comparable with these techniques since DidaxTo applies dynamic pattern discovery for the extraction of high-quality opinion words and opinion targets and does not output a natural language summary.

Topic modeling Given that our method extracts descriptive opinion words and targets, the output is similar with topic modeling. In the context of opinion mining, for example, *aspects* (opinion target and their attributes) [31] correspond to *topics* in topic modeling. A representative example can be found in [31] where multiple dimensions of opinion mining are modeled. These dimensions include aspect identification, opinion word detection, polarity assignment and the distinction between general or aspect specific opinion words. The paper utilizes topic modeling to represent these dimensions. In the same line of research, Thonet et al. [29] present a topic model that discovers viewpoints (opinion holders), topics (aspects, targets) and opinions. Lastly, another topic modeling approach is presented in [18]. This approach is tailored for Twitter since it takes advantage of elements like hashtags and mentions.

Deep learning During the last few years, deep learning and word embeddings have started to appear in research efforts related to sentiment analysis. In [27], for example, the authors take advantage of a neural language model to initialize the embeddings and a convolutional neural network to refine them. The approach is applied to phrase-level and message-level sentiment analysis of Twitter data. Moreover, Li et al. [17] combine a Recursive Neural Network with a Recursive Neural *Tensor* Network. The method is evaluated on the Stanford Sentiment Treebank [28].

3 The DidaxTo approach

The approach comprises of four pattern extraction steps. Each step creates a pattern list that will be utilized, extended and refined in a following step. More specifically, the proposed method can be summarized in the following steps:

(A) Seed Patterns Extraction
(B) Conjunction Patterns Extraction
(C) Opinion Target Patterns Extraction
(D) Double Propagation Patterns

In addition, there are some extra steps that complement the whole process like the preprocessing of the corpus, validating and selecting patterns as well as the polarity assignment to the patterns. The reader can find details about the above steps and sub-processes in the following sections.

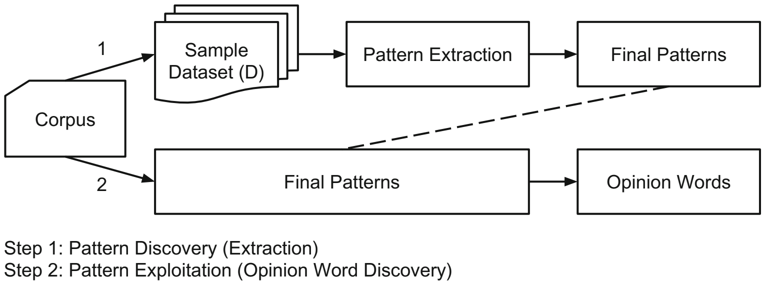

Fig. 1 Overview of the Pattern Discovery and Pattern Exploitation Process. At first, the algorithm extracts the (final) patterns (Step 1). These patterns are used (even in the same corpus) to extract opinion words (Step 2). This paper focuses on Pattern Extraction (Step 1), but, naturally, the proposed approach is evaluated on how good it performs in Step 2 (Pattern Exploitation)

Figure 1 presents an abstract overview of the whole process, and Fig. 2 presents a detailed description of how Pattern Extraction is being achieved. Table 1 presents the notation that is used throughout the paper.

Note All examples in the following sections are based on the corpus that we describe in our evaluation section (Sect. 5).

3.1 Pre-processing

The pre-processing part of our approach includes the following steps:

Sample dataset preparation The extraction of patterns is based on the analysis of a subset of the original corpus. Essentially, this set consists of the larger opinions of the original corpus. We followed this approach mainly to avoid massive extraction of patterns that would solely prolong the extraction process without improving patterns utilization, as this is something we observed by experimenting with varying size of datasets.

Part of speech word list filtering Having an initial list of auxiliary words organized by part of speech (articles, auxiliary verbs, comparatives, adverbs, quantifiers, prepositions, interjections), the algorithm analyzes the sample dataset and records the frequency of each one of them. It then filters out low-frequency words since, after preliminary investigation, we observed that they increase the problem's complexity without offering any significant benefits to the outcome of the approach.

3.2 Seed Patterns Extraction

At this stage, the algorithm utilizes the sample dataset, the preprocessed part-of-speech word list and a set of `Seed` (S) opinion words (see Fig. 2). This is a domain neutral list of opinion words (e.g. "bad", "ugly", "wonderful"), along with their polarity (positive, negative), provided by [10,34]. The process looks for `Seed` words in the sample dataset. When it locates one, it stores its location and applies part of speech tagging (based on the POS word list (L)) to the words before and after this location. The tagging is applied up to a certain distance from the location of the seed word. We refer to this distance as "depth" since the algorithm does a depth search in the sentence using the seed word as a starting point. When the tagging

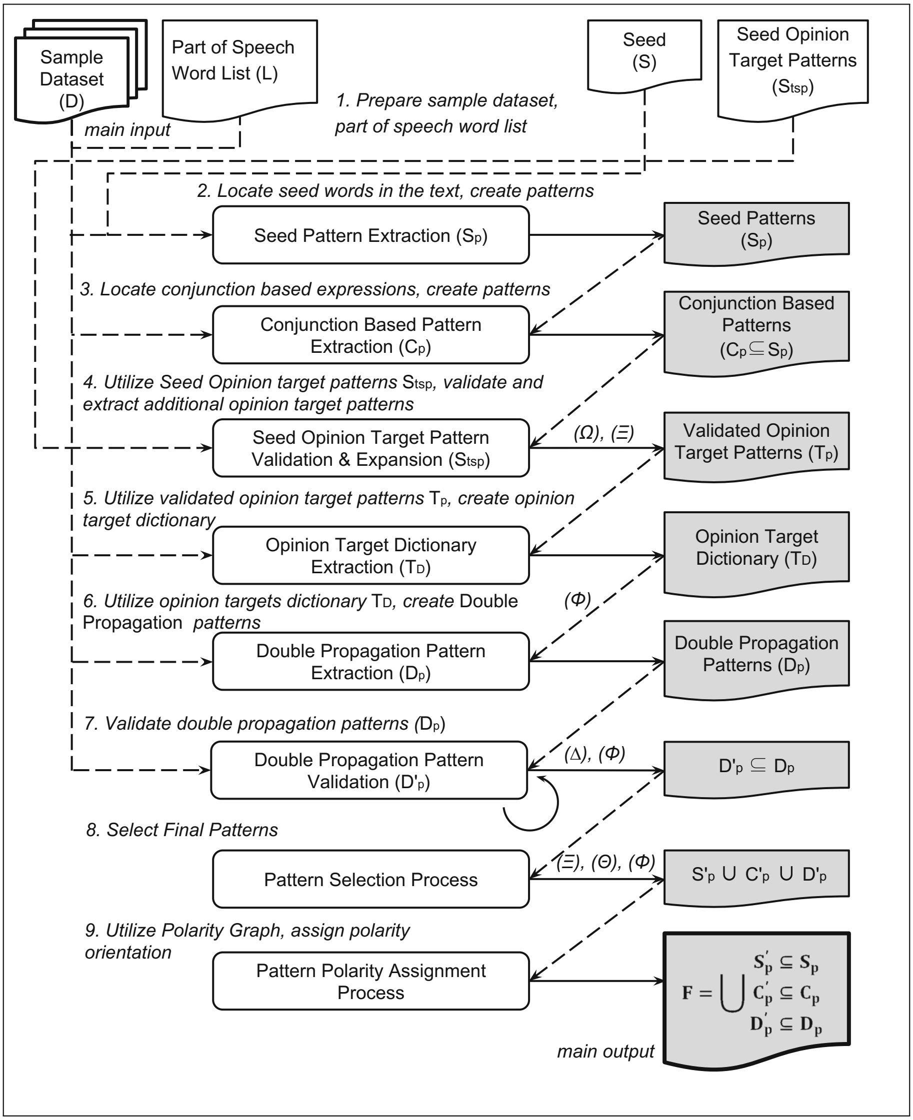

Fig. 2 Overview of the Proposed Approach. *Gray shaded items* represent the extracted pattern set that is enhanced in each step. The reader can find details about each step (1–9) as well as the parameters used in each one (*greek letters* in parentheses) in later sections. The main input (**D**) of our algorithm is the Sample Dataset, and the main output is the set final extracted patterns (**F**). These patterns can be exploited for the extraction of opinion words

does not recognize a word, then it annotates it as unknown (we use the "?" symbol for this element). Table 2 presents a small list of extracted seed patterns along with an example sentence.

What this step accomplishes is to identify the appearances of patterns of positive and negative words. Multiple appearances of a pattern are counted and considered. Obviously,

Table 1 Notation

L	Part of speech word List	l	Low-frequency pattern
D	Sample dataset	Ω	Confidence threshold
S	Seed	Ξ	Similarity threshold
S_{p}	Seed patterns	Δ	Depth search threshold
C_{p}	Conjunction patterns	Φ	Frequent threshold
S_{tsp}	Seed opinion target patterns	Θ	Similarity hit threshold
D_{p}	Double propagation patterns	τ_i	Pair of pattern labels (transition)
v	Validated seed opinion target pattern	τ_{i1}, τ_{i2}	First, second label in a transition
a	Additional opinion target pattern	T_D	Opinion target dictionary
T_{p}	Validated opinion target pattern	γ	Secondary polarity term
p	Candidate pattern	λ	Primary polarity term
i	Transition index	$op\text{-}word(p, n)$	(posw, negw) opinion words labels
$\omega()$	Confidence score	df	Distance factor
$inf()$	Information score	δ	Distance between secondary, primary polarity terms
$s()$	Similarity score	N	Number of labels in a pattern
h	High-frequency pattern		

Table 2 Sample of seed extracted patterns

Pattern												
Sp_i	$-l = -3$	...	-2		-1		**Centroid**		$+1$		$+2 = +r$	$Freq$
1.	?	–	?	–	**art**	–	**posw**	–	?	–	?	10
e.g:	*features*		*were*		*a*		*good*		*idea*		*but*	
2.	**pronoun**	–	**verb**	–	?	–	**posw**	–	?	–	?	6
e.g:	*i*		*would*		*highly*		*recommend*		*this*		*tv*	
3.	**verb**	–	?	–	?	–	**negw**	–	**fut**	–	?	1
e.g:	*set*		*are*		*very*		*irritating*		*to*		*listen*	

(Candidate) Patterns are appearing with bold-typeface. r and l are the right and left boundaries, respectively (where we limit the annotation process). The annotation starts from the centroid that in this case is a (positive or negative) seed word. Words of unknown type are marked with ? The last column ($freq$) is the number of times this pattern is found in the (sample) corpus. Targeting at a compact representation of the examples, we only present a fragment of the sentences in our examples

the frequency and length of the pattern relate to its information value. These patterns will be utilized in the next steps.

This procedure of annotating a sentence of the corpus in order to identify patterns is described in Algorithm 1. It is a general procedure used throughout our approach. The `Centroid` word is the point where the algorithm initiates the labeling process (before and after this location). For each step of the approach, the `Centroid` is different, and therefore, we explicitly define it in the next sections. For this step (Seed Patterns Extraction), the centroid is the positive or negative seed word found in the sentence.

Algorithm 1: Patterns Extraction Process

```
input  : centroid location (i), sentence (s), pattern (p-initially set to null)
output: p - extracted patterns

extractPatterns(i, s, p)
if there are unlabeled words left of the centroid then
    label ←labelDistribution(left word)
    // create new pattern;
    p ←createPattern(label , p)
    if p exists then
        addFrequency(p)
    else
        storePattern(p)
if there are unlabeled words right of the centroid then
    ...
    (do exactly as above)
    ...
if haven't reached end of depth search limit then
    // increase depth search;
    i ← i + 1
    // keep extracting patterns;
    extractPatterns(i, s, p)
else
    End Process
```

3.3 Conjunction Patterns Extraction

This is the second pattern extraction step. At this step, we employ theories of sentiment consistency in order to discover new patterns. According to these theories [9] opinion words followed by conjunctions indicate the existence of new opinion words of the same polarity. Take, for example, the phrase "beautiful and practical". If "beautiful" is a positive opinion word, so is "practical". The process looks for <seed conjunction> word pairs and, when it locates one, it initiates Algorithm 1. The `Centroid` in this case is the location of the conjunction.

This step accomplishes to extract what we call "conjunction pattern" dependencies that will be utilized by the following processes. Table 3 shows a sample list of these type of patterns along with an example sentence.

3.4 Opinion Target Patterns Extraction

One of the main objectives of the proposed method is to identify patterns capable of extracting domain opinion words in an unsupervised manner. In this step, we utilize opinion targets, e.g. "phone", "screen", "size", "chair" in order to extract domain opinion words. For example, in the expression "*these chairs are barely comfortable*", "chair" is a candidate opinion target and "comfortable" is a candidate domain opinion word. The method that exploits opinion targets to extract opinion words and then, opinion words to extract new opinion targets, is known as Double Propagation [12,25] (see Fig. 3). To tackle this task, we start from the

Table 3 Sample of conjunction patterns, where r and l are the right and left boundaries, respectively

Pattern												
Cp_i	$-l=-3$	...	-2		-1		**Centroid**		$+1$		$+2=+r$	*Freq*
1.	?	–	**adverb**	–	**posw**	–	**conj**	–	?	–	?	4
e.g:	*actually*		*quite*		*good*		*and*		*provide*		*lots*	
2.	?	–	**verb**	–	**posw**	–	**conj**	–	**art**	–	?	3
e.g:	*fabric*		*is*		*great*		*and*		*the*		*multi*	
3.			?	–	**negw**	–	**conj**	–	**adverb**	–	?	2
e.g:			*very*		*disappointed*		*and*		*even*		*surprised*	

Patterns are indicated with bold-typeface

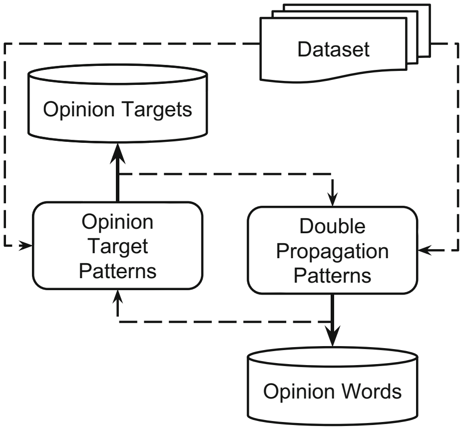

Fig. 3 A flowchart describing how this step of our approach (Opinion Target Patterns Extraction) is enhancing the Double Propagation process. Opinion Target Patterns aid in discovering new opinion targets that are utilized in Double Propagation

opinion targets. Initially, we extract a set of opinion target patterns that we validate and then identify more of them by calculating pattern similarities.

Validating opinion target patterns We start from the observation that some part of speech elements such as pronouns, articles and opinion words are usually followed by opinion targets [25]. That is: “that chair” <pronoun op-target>, “nice screen” <op-word op-target>, “the size” <article op-target>. These strong opinion target expressions are usually part of a larger sentence and thus create dependencies with nearby part of speech elements. Our goal in this step is to locate these opinion target expressions along with the nearby part of speech elements, create patterns and select the most promising ones.

We first employed a small list of short length opinion target detection patterns, e.g: <art comp art op-target> (that we call Seed Opinion Target Patterns—S_{tsp}) and validated them applying the following method. We calculate all the information (frequency) scores of the part of speech elements relations that existed in these patterns, e.g: art→(?):58%, posw→(?): 65%, art→posw:7%. The information score expresses the frequency of that specific part of speech element relation against all other possible elements, in our corpus. The unknown element (?) in these part of speech relations declares the candidate opinion target.

In the validation process, we applied the information scores of the seed pattern list in the elements of the seed opinion target detection patterns. Table 4 illustrates a sample of validated opinion target detection patterns along with calculation examples. Confidence (ω) is the sum

Table 4 Validated seed opinion target detection patterns

v_i	Pattern						**Candidate opinion target**	Confidence $\boldsymbol{\omega}$
1.	int	→	art	→	posw	→	?	
e.g:	*for*		*an*		*adequate*		*durance*	
		20%		5%		57%		82%
2.	posw	→	int	→	art	→	?	
e.g:	*impressive*		*for*		*the*		*money*	
		5%		7%		65%		77%
3.	art	→	comp	→	art	→	?	
e.g:	*the*		*greater*		*the*		*screen*	
		3%		5%		65%		73%

of the information scores of the opinion target pattern elements and expresses the validity of a pattern according to our corpus. The calculation is shown in Eq. 1. A pattern $\mathbf{v}$ of length N consists of $N-1$ transitions (label pairs) $\mathbf{v} = \{\tau_1, \tau_2, \ldots, \tau_{N-1}\}$. Each transition has its own information score $inf(\tau_i)$. Thus, $\omega(\mathbf{v})$ is equal to $\sum_i^{N-1} inf(\tau_i)$.

In the pattern evaluation process, we utilize the confidence threshold (Ω). Patterns that overcome this threshold are finally selected. By applying this filtering, we target to identify which of the seed opinion target detection patterns fit better the data and the domain. We call this set of patterns as Validated Seed Opinion Target Detection Patterns.

$$\omega(\mathbf{v}) = \sum_{i=1}^{N-1} inf(\tau_i) \tag{1}$$

Extracting additional opinion target detection patterns The validated seed opinion target patterns that were extracted in the above step promise high extraction accuracy of opinion targets, but they are only a few in number and consequently unable to capture all candidate opinion target expressions in the corpus of opinions. However, since these patterns are manually defined and validated in the particular domain, they are very accurate. To take advantage of their accuracy, we search for additional patterns by following a pattern matching approach.

The method we employ is based on the observation that the unknown label (?) in the already extracted patterns stands most probably for a candidate domain opinion word or candidate opinion target. This is based on the fact that the approach already is aware of auxiliary and generic (seed) opinion words, and hence, there is a high chance that the remain unknown words (?) will be domain-specific opinion words or opinion targets. The following approach tries to validate this assumption focusing only at the opinion targets and confirm that these candidates are indeed opinion targets.

We employed the sample dataset and located words that can be classified as "unknown (?)". Then we called Algorithm 1 and inserted the unknown (?) word as centroid and the candidate sentence to create unknown (?)-based patterns dynamically. We also constrained the amount of extracted patterns based on pattern length. We also filtered out candidate patterns that start from an unknown label (?). The rest of the unknown (?)-based patterns were filtered in a similarity approach with the validated opinion target patterns.

Table 5 Validated additional opinion target detection patterns (based on similarity)

Pattern										
a_i	$-l = -3$		...		-1		**Centroid**		$+1 = +r$	$s(a_i)$
1.			**posw**	–	**art**	–	?	–	**int**	96.39%
e.g:			*worth*		*the*		*price*		*of*	
2.	**negation**	–	**verb**	–	**art**	–	?			93.78%
e.g:	*not*		*touch*		*the*		*wall*			
3.	**negw**	–	**verb**	–	**art**	–	?			93.45%
e.g:	*scratchy*		*on*		*the*		*skin*			
4.			**verb**	–	**art**	–	?	–	**verb**	91.21%
e.g:			*on*		*the*		*floor*		*are*	

Just as in previous subsection we assume patterns **v** (validated seed opinion target detection) and **a** (patterns with unknown words) consist of a number of transitions $\mathbf{v} = \{\tau_1^v, \tau_2^v, \ldots, \tau_{N-1}^v\}$ and $\mathbf{a} = \{\tau_1^a, \tau_2^a, \ldots, \tau_{N-1}^a\}$. We exploit the information score of each transition in a pattern ($inf(\tau_i)$) in order to create an *information vector* for each pattern as follows:

$$\mathbf{inf}(\mathbf{v}) = \left\{inf(\tau_1^v), inf(\tau_2^v), \ldots, inf(\tau_{N-1}^v)\right\}, \text{ and}$$
$$\mathbf{inf}(\mathbf{a}) = \left\{inf(\tau_1^a), inf(\tau_2^a), \ldots, inf(\tau_{N-1}^a)\right\}$$

Note that the two patterns might not be of the same length. However, in order to calculate their similarity we keep only the $N - 1$ common elements. We utilize these vectors and calculate the cosine similarity as shown in Eq. 2. We validate a pattern based on cosine similarity function and a threshold (Ξ). The patterns that overcome that threshold are finally selected.

$$\text{score}(\mathbf{a}) = \cos(\mathbf{inf}(\mathbf{v}), \mathbf{inf}(\mathbf{a})) == \frac{\sum_{i=1}^{N-1} inf(\tau_i^v) \cdot inf(\tau_i^a)}{\sqrt{\sum_{i=1}^{N-1} (inf(\tau_i^v))^2}\sqrt{\sum_{i=1}^{N-1} (inf(\tau_i^a))^2}} \tag{2}$$

These two groups of patterns, the Validated Opinion Target Patterns (Table 4) and the Additional Opinion Target Patterns (based on similarity) (Table 5), constitute the basis for the extraction of the candidate opinion targets that will be utilized at the next pattern extraction step. We will refer to these two groups of patterns as Validated Opinion Target Patterns.

The previous steps and the two groups of patterns they provide aim at having a successful double propagation step at the pattern exploitation process.

3.5 Double propagation pattern extraction

This process is implemented in the following steps.

- Opinion target extraction: In this step, we extract the opinion target set that will be utilized for the double propagation pattern extraction. We employ the validated opinion target patterns and the sample dataset to extract candidate opinion targets. Then we apply a *frequency threshold* (Φ), e.g. $\Phi \geq 10$, and select those that overcome this threshold. These opinion targets are considered as *validated opinion targets*, since they appear multiple times in the corpus. Table 6 shows a sample of extracted and validated opinion

Table 6 Sample of validated opinion targets after exploiting the opinion target detection patterns on the tablets sample dataset

Candidate opinion targets and number of occurrences [frequency]											
Tablet	[302]	Sony	[96]	Ipad	[79]	Market	[57]	Android	[41]	Device	[35]
Video	[31]	Battery	[28]	Sd	[28]	Galaxy	[28]	Samsung	[27]	Gb	[26]
Wifi	[26]	USB	[25]	Software	[24]	Big	[22]	Music	[21]	Price	[21]

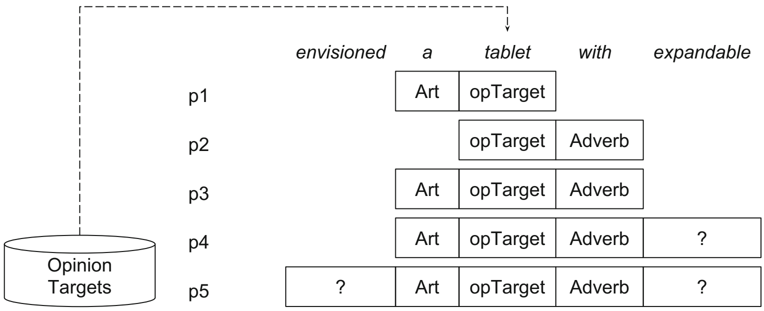

Fig. 4 Double Propagation Pattern Extraction Process on an example sentence

targets. The next step utilizes the *validated opinion targets* to extract double propagation patterns.

- Pattern extraction: Double propagation is the process that extracts opinion words utilizing opinion targets and vice versa [12,25]. At this step, the process looks for validated opinion targets in the sample dataset. When it locates one, it initiates Algorithm 1 providing it with information about the location of the opinion target word as well the candidate sentence. The Centroid in this case is the location of the opinion target. This opinion target-based pattern extraction process builds up the double propagation patterns (see Fig. 4).
  Next we employ these patterns utilizing the opinion target label location and annotate adjacent labels where candidate opinion words can be explored. This process is the validation of the double propagation patterns (see next section). Table 7 shows a sample of the double propagation patterns that have been validated using this process.
- Pattern validation: Double propagation pattern validation is the process that employs the double propagation patterns, analyzes them through a clustering process and keeps only the most promising ones. As will demonstrate in our experimental evaluation, these patterns are highly accurate in identifying opinion words. To alleviate this task, we also employed the validated opinion target list and the sample dataset.
  The process initially employs a candidate double propagation pattern from the pattern list and looks for validated opinion targets in the sample dataset. When it locates one, it initiates Algorithm 2 providing it with information about the location of the opinion target word, the sentence as well the candidate double propagation pattern.
  The process analyzes the candidate pattern and the candidate sentence exploring an appropriate pair matching. It utilizes the Opinion Target from the label and the opinion target from the sentence as Centroids and assumes all the unknown labels (?) in the pattern and the sentence up to a certain *depth threshold* (Δ) as candidate opinion words. When there is a match between the <optarget - unknown: (?)> pair in the pattern and

Table 7 Sample of validated double propagation patterns

Pattern										
	$-l=-5$	...	...	...	-1	**Centroid**	$+1$	$+2=+r$	*freq*	Opinion word position
1.	**art**	?	?	**int**	**art**	**optarget**			[5]	[−3]
e.g:	*the*	*slanted*	*angle*	*of*	*the*	*chair*				
2.	?	**verb**	**art**	?	?	**optarget**			[5]	[−1]
e.g:	*really*	*like*	*the*	*multi*	*colored*	*buttons*				
3.			?	?	**art**	**optarget**	**int**	**art**	[4]	[−2]
e.g:			*thus*	*making*	*the*	*height*	*of*	*the*		
4.				**vconj**	**art**	**optarget**	?	?	[4]	[+2]
e.g:				*when*	*the*	*channel*	*does*	*display*		

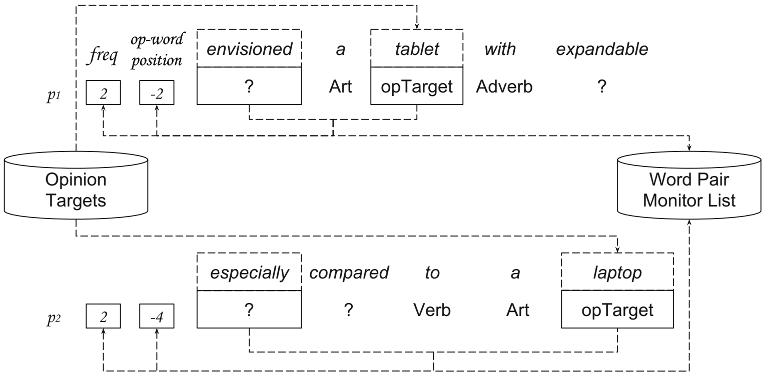

Fig. 5 Two examples of the Double Propagation Validation process. Sentences of the corpus and the candidate patterns (to be validated) are aligned on the opinion target (actual words in the sentences and opinion Target label in the pattern). The number of successful alignments of a pattern with sentences in the corpus is indicative of the utility of the pattern. This is how we validate the double propagation patterns

the sentence it stores this information in the pattern and also in a list where we monitor the frequency that this pair appears (see Fig. 5). The process iterates for the double propagation patterns and the sentences of the sample dataset.

Finally, we select the patterns that have high frequency and we place the <optarget - unknown: (?)> pair in the monitor list. At Table 7, the column "*opinion word position*" declares the relative distance from the opinion target where a candidate opinion word exists. The sign [−/+] stands for left and right of the Centroid, and the number represents the relative distance, respectively.

3.6 Patterns selection process

The pattern extraction steps as described above lead to the discovery of a great number of patterns most of them appropriate for the extraction of domain opinion words. In order to optimize the number of patterns and address complexity issues without significant information loss, we devised a two-step mechanism for filtering out redundant patterns.

Algorithm 2: Double Propagation Pattern Validation Process

input : ls: opinion target location in the sentence, lp: opinion target location in the pattern, s: sentence, p: pattern, δ: max depth search, d: current depth search
output: Validated Double Propagation Patterns

```
validateDoublePropagationPatterns(ls, lp, s, p)
d←1, Δ ←getMaxDepthThreshold()
while d≤ Δ do
    if left of ls and lp there exists candidate opinion word then
        if opinion target ot - opinion word ow pair exists in the Monitor List then
            increasePairFrequency(ot, ow)
        else
            addPair(ot, ow)
        if p and ot - opinion word ow pair then
            increasePatternFrequency(p, ot, ow)
        else
            addPattern(p, ot, ow)
    if right of ls and lp candidate opinion word then
        ...
        (do exactly as above)
        ...
    // increase depth;
    d ←d + 1
```

At first step, we selected from the seed pattern list, all those that conform to an appropriate pattern length regardless of their frequency. The pattern length is the number of labels existing in a pattern. The benefit of long patterns is that they include distant opinion word modifiers, e.g: ("***don't** just feel* ***like*** *going through*"), while shorter patterns ("***not fluent***") ensure that we do not miss polarity assignment of shorter opinion expressions. From the conjunction and the double propagation validated patterns, we selected those that overcome a manually selected frequency threshold.

At the second step, we applied a similarity-based method for the selection of low-frequency patterns. The intuition behind this idea is that high-frequency patterns are important, but low-frequency patterns also include valuable information. The similarity-based method targets to exploit this information. It is implemented as follows. Initially, we employed the high-frequency patterns of the seed pattern list, the conjunction-based patterns and the validated double propagation patterns. From these patterns, we extracted all the information scores of the part of speech elements similarly to what we did in the opinion target pattern extraction process (see Sect. 3.4). Examples of these information scores can be: art→verb: 57%, comp→negation: 23%. Next we iterated the process, but this time we included the low-frequency patterns from the conjunction-based and the double propagation validated patterns. We exploited all these information scores as if they were feature vectors.

The similarity Eq. 3 exploits these information scores and calculates a similarity value $s(l)$. At the selection process, we employed two thresholds. The *similarity threshold* (Ξ) which is the minimum value a low-frequency pattern (**l**) must be similar to a high-frequency pattern (**h**) (see Eq. 2). The *hit threshold* (Θ) is the percentage of times a low-frequency pattern's similarity score ($s(l)$) must overcome the similarity threshold (Ξ) for a number of high-frequency patterns. For example, if a low-frequency pattern has a similarity value over $s(l) = 90\%$ with all high-frequency patterns, while the threshold is $\Xi = 90\%$ and hits 100% (that means that $s(l) > \Xi$ against all high frequency patterns **h**) while the hit threshold is $\Theta = 95\%$, it is finally selected (see Eq. 3, for the flow of calculations).

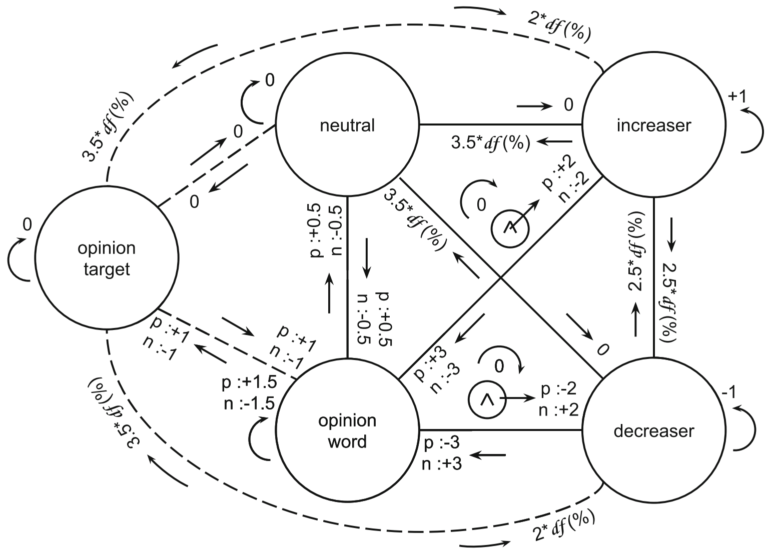

Fig. 6 Patterns Polarity Decision Graph. The graph represents only pairwise relationships. In other words, only paths of length 2 are allowed. Multiple 2-step paths are utilized in order to assign polarity to a sentence. Edges with df notation declare that this pair of labels requires an indirect polarity assignment based on Eq. 4. *Dashed lines* indicate the utilization of the opinion target labels in the pattern polarity, when double propagation patterns are employed from the *graph*

Please notice that in Eq. 3 we implement the cosine similarity by aligning the vectors of the low-frequency patterns (**l**) to the vector space of the high-frequency patterns (**h**) just as in Eq. 2.

$$s(\mathbf{l}) = \cos(\mathbf{h}, \mathbf{l}) == \frac{\sum_{i=1}^{N-1} h_i \cdot l_i}{\sqrt{\sum_{i=1}^{N-1} (h_i)^2}\sqrt{\sum_{i=1}^{N-1} (l_i)^2}} \tag{3}$$

Recall that **h** and **l** are expressed as vectors of transitions (h_i and l_i, respectively).

3.7 Pattern polarity assignment

The patterns extracted in the above steps represent valuable label dependencies and can help us identify opinion words and targets. However, they provide no information about the polarity of the candidate opinion words they extract. In order to alleviate this task, we devised a Polarity Decision Graph (see Fig. 6), which analyzes patterns and assigns the appropriate orientation.

The implementation of the graph follows the general linguistic rules in opinion mining [5], but the assigned weights and the calibration are the result of experimental effort in order to cover a broad range of expression cases on the corpus of opinions. Next we explain the construction and utilization of the graph.

Initially, we separated the predefined labels in five distinct categories. These are **neutral**, **opinion target**, **increaser**, **decreaser** and **opinion word**. Each of these categories includes a set of labels. The **neutral** category includes the *article, pronoun, verb, conjunction, unknown (?), adjective, intention and adverb* labels. The **increaser** category includes the *increaser and comparative* labels. The **decreaser** category has the *decreaser* and *negation* labels, whereas the **opinion target** category is represented by the *opinion target* label. Finally, the **opinion word** category consists of the *positive* and *negative* labels. These categories—and hence their including labels—are expressed in the polarity graph (see Fig. 6).

The graph is utilized as follows. Given a pattern p that is expressed as a sequence of transitions $\tau_1, \ldots, \tau_n$, and each transition is a pair of labels $\tau_i = \{\tau_{i,1}, \tau_{i,2}\}$, we start with the first transition $\tau_1 = \{\tau_{1,1}, \tau_{1,2}\}$. This two-step path $(\tau_{1,1}, \tau_{1,2})$ is applied to the graph, and the pairwise polarity is extracted from it (the value from the edge in the graph). For example, if the first transition of the pattern is $\tau_1 =$ {opinion-word (positive), neutral}, then the value obtained from the first τ_1 would be $+0.5$. The same procedure iterates for each transition of p, and hence, a sequence of polarity scores is the outcome of this process. These scores are then summed in order to extract the polarity of the pattern—at least in this simple case. In the graph, three types of polarity assignments are expressed: Direct, Indirect and the Split Pattern Case.

Direct polarity assignment Let us consider, for example, the pair of categories <decr op-word(p)>. The retrieved value from the graph is -3. Similarly <decr op-word(n)> would lead to the value $+3$. This value represents a *partial pairwise polarity score* (e.g. $+3$ or -3 in our examples) that will aid in calculating the polarity of the sentence. This is achieved by summing all pairwise polarity scores. Positive values are normalized to $+1$ (positive orientation) and negative values are normalized to -1 (negative orientation). Zero (0) stands for neutral polarity. For example, in the sentence of Table 8 the algorithm retrieves the polarity values as shown in steps 1 through 5, calculates their sum and finally assigns -1 which stands for negative polarity. This value will also be inherited to the candidate opinion word (in this case "readable") when the pattern is utilized.

Indirect polarity assignment In this case, we follow a very similar procedure. However, when we apply a transition (pair) to the graph, the outcome will be a calculation formula including the term df. For example, in the sample sentence of Table 10 "*few android tablets will work wonderfully*", when the algorithm employs the pair of labels $< decr\ \ neutral >$ and locates it in the graph, it retrieves the calculation formula $3.5 \times (df)$. df stands for distance factor (see Eq. 4).

The distance factor (df) takes into account the polarity impact of a label which lies at a distance from an opinion word and hence even remotely affects the polarity of the pattern. The intuition behind this parameter is that the larger the distance from the opinion word, the lesser the impact will be.

$$df = \frac{1}{\delta} \times \gamma \times \lambda \tag{4}$$

$$\lambda = \left\{ \begin{array}{ll} +1 & \text{if } \tau_{i,2} \in op\text{-}word(p) \\ -1 & \text{if } \tau_{i,2} \in op\text{-}word(n) \\ \ \ 0 & \text{Otherwise} \end{array} \right\} \quad \forall\, i \in \{\tau_\gamma \ldots \tau_\lambda\} \tag{5}$$

The formula (Eq. 4) consists of three parameters: The *distance* (δ) between the first label of the *activating transition* (τ_γ) (the transition where the distance factor (df) encountered in the graph) and the second label of the *concluding transition* (τ_λ) (the transition that includes an opinion word label, if that exists). The *secondary polarity term* (γ) is a value obtained from Table 9 and takes into account the polarity of the activating transition (τ_γ). The *primary polarity* term (λ) is a value obtained from Eq. 5 and takes into account the polarity of the concluding transition (τ_λ). Here activating and concluding transitions refer to the (first and last) transitions of the pattern that activated the indirect polarity processes (see Fig. 7). In

Table 8 Application of direct polarity assignment in a sample sentence

Steps		1		2		3		4		5		Polarity
Example	not	–	readable	–	in	–	bright	–	sunlight	–	and	
Graph encoding	decr	–	op-word(p)	–	neutral	–	op-word(p)	–	neutral	–	neutral	
Partial values		−3.0		+0.5		+0.5		+0.5		0		(−1.5) → −1

Table 9 Definition of the secondary polarity term (γ)

		Increaser	*Decreaser*	*Neutral*	*op-target*
1.	*Increaser*	0	−1	+1	+1
2.	*Decreaser*	+1	0	−1	−1
3.	*op-target*	+1	−1	0	0

secondary (γ) [------ (δ) ------] primary (λ)

Fig. 7 Primary (λ) and secondary polarity (γ) terms in relation to distance (δ) in indirect polarity implementation

Table 10 Application of indirect polarity assignment in a sample sentence

Steps		1		2		3		4		5		Polarity
Example	few	–	android	–	tablets	–	will	–	work	–	wonderfully	
Graph encoding	decr	–	neutral	–	neutral	–	neutral	–	neutral	–	op-word(p)	
Partial values		−0.7		0		0		0		+0.5		(−0.2) → −1

the example of Table 10, the activating transition is the <decr neutral> and the concluding transition is the <neutral op-word(p)>.

In our example, in Table 10 the distance factor (df) is calculated as follows. Initially, the process employs the activating transition <decr neutral> and from rule (2) in Table 9 retrieves value ($\gamma = +1$) for the secondary polarity term. The algorithm continues by using the next transition and checks if an opinion word label (posw, negw) exists. It iterates for every transition in a pattern and if such label exists, it retrieves the appropriate value from Eq. 5 or else returns zero. In the example, this transition is the <neutral op-word(p)> and the retrieved value is ($\lambda = -1$) for the primary polarity term.

The distance between "decr ... op-word(p)" is 5, and the calculation for the (df) in this indirect polarity assignment is: $df_{\text{decr...op-word(p)}''} = 3.5 \times \frac{1}{5} \times \big(-1\big) \times \big(+1\big) = -0.7$. The rest of calculations in the sample sentence are according to the direct type and are illustrated in Table 10.

Split pattern case The third case is when an opinion word refers to another opinion word indirectly through an increaser or a decreaser. In this case, we assume a pattern nests two independent opinion expressions and we split it in two parts. In Graph 6, we use the symbol "⊘" to indicate the split point. For example, in the example of Table 11 "*the mobile is great doesn't make any noise*", we split the corresponding pattern in two parts and calculate the polarity of each part independently. The sum defines the polarity of the pattern. An example is presented in Table 11.

In Table 12, we provide examples of sentences along with their polarity assignment.

4 Applying the patterns: domain-specific opinion word discovery

At this step, we utilize the extracted patterns in order to perform sentiment analysis and discover opinion words. As we discussed in the introduction, domain-specific words are important for accurate sentiment classification (e.g. positive vs negative review). The patterns extracted by our methodology can be utilized in two ways:

Table 11 Split pattern case and graph application in a sample sentence

Steps		1		2		3				1		2		3		Polarity
Sentence	the	–	mobile	–	is	–	great	–	doesn't	–	make	–	any	–	noise	
Graph encode	neutral	–	neutral	–	neutral	–	op-word(p)	$\oslash$	decr	–	neutral	–	neutral	–	op-word(n)	
Partial values		0		0		+0.5				+1.17		0		−0.5		(+1.17) → +1

Table 12 Examples of patterns and their polarity assignment

	Pattern						pol
(1)	*table*	*is*	*the*	*perfect*	*size*	for	
	?	?	**art**	**posw**	?	?	
		0.00	0.00	0.50	1.00	1.00	[+1]
(2)	*no*	*doubt*	*it*	*is*	*sufficiently*	*durable*	
	negation	**negw**	**pronoun**	?	**posw**	**posw**	
		3.00	2.50	2.50	3.00	4.50	[+1]
(3)	*am*	*no*	*longer*	*excited*	*about*	*this*	
	?	**negation**	?	**posw**	**adj**	?	
		0.00	−1.75	−1.25	−0.75	−0.75	[−1]
(4)	*positioned*	*the*	*mattress*	*somewhere*	*in*	*between*	
	?	**art**	**optarget**	?	**int**	?	
		0.00	0.00	0.00	0.00	0.00	[0]

- One can apply the patterns in the corpus and extract the opinion words along with their polarity. After that, the (domain-specific) opinion words can be utilized as dictionaries in order to perform sentiment analysis. To put it simple, a review should be considered positive if it contains more positive opinion words than negative. We call this type of sentiment classification "dictionary-based classification".
- In the previous section, we have suggested a method for assigning polarity not only to opinion words but also to patterns. Hence, we can identify applications of patterns in a document and assign sentiment to the document according to the aggregated polarity of all patterns applicable to the document. In case that no patterns can be applied in a sentence, the polarity is assigned by utilizing the intra-sentential sentiment consistency [12] which relies in previously discovered opinion words in a sentence. We call this type of sentiment classification as "pattern based".

The algorithm that extracts the opinion words starts from the seed pattern list. It parses the opinionated content and locates seed words; then, it extracts the polarity of the pattern. Next it continues employing the conjunction patterns. It locates <posw-conj> or <negw-conj> word pair expressions and matches the appropriate pattern in the sentence under study.

In the final utilization step, the algorithm employs the validated double propagation and the validated opinion target patterns. These two groups of patterns extract new opinion words and opinion targets in an iterative process known as double propagation. Table 13 presents samples of user-generated content along with pattern classification and Table14 presents a sample of a domain-specific lexicon that was extracted by DidaxTo. We notice that most of the words discovered by our method are indeed domain specific as many of them are not generic (e.g. amateurish, grip, restart). Note that opinion words do not need to be adjectives. Opinion word can be any term that reveals positive or negative sentiment (e.g. cheapness, durability).

Table 13 Sample of user-generated opinion content and utilization of pattern-based classification on the "tablets" domain

Opinion samples		Classification	
		User	DidaxTo
1.	*a* **nice[+1]** *little tablet, i had mine for almost half a year. the battery lasts a few days for me. also the back and front cameras are a plus. gps which we use a lot* **works[+1] well[+1]** *and* **works[+1]** *offline too. i likes it so much had to get a second one for my mom*	Positive	Positive
2.	**great[+1]** *buy, i did buy gb of memory for it, the only* **issue[−1]** *i have is that it can only be used as a storage card. you cant set it to be a default location or* **anything[−1]**, *everything must be manually transferred to it*	Negative	Negative
3.	*i had* **trust[+1]** *on samsung brand, but, despite of being worldwide, samsung does not* **assure[−1]** *abroad* **warranty[−1]** *for this device*, even to be fixed on official assistance. thanks god is just one of my *other devices of this brand, but it's* **enough[+1]** *to think about not bu*	Neutral	Neutral

On the second column (DidaxTo), we have noted the classification of our approach. On the first, we have noted the sentiment assigned by the author of the review. This was based on the "stars" that the user gave to that product

Table 14 A sample of the extracted domain-specific lexicon from the "tablets" domain

Opinion words extracted with Seed patterns							
Rich	[2]	Supports	[3]	Sharp	[5]	Vivid	[5]
Available	[41]	Top	[25]	Sluggish	[−3]	Disappointment	[−4]
Aggravating	[−1]	Terrible	[−6]	Slowed	[−1]	Downsides	[−2]
Opinion words extracted with Conjunction-based patterns							
Durability	[1]	Casually	[1]	Grip	[1]	Makes	[1]
Wiggle	[1]	Image	[1]	Amateurish	[−1]	Display	[−1]
Shade	[−1]	Cheapness	[−1]	Hickups	[−1]	Tinny	[−1]
Opinion words extracted with Double Propagation Based Patterns							
Covers	[1]	Lockup	[1]	little	[2]	Needs	[2]
Repertoire	[1]	Restart	[−3]	Interaction	[1]	Resolution	[1]
Holding	[−1]	Protector	[−1]	Sensitivity	[1]	Splashtop	[−1]

We demonstrate the contribution of each step separately (Seed, Conjunction, Double Propagation Patterns). At the brackets $[a, b]$, we illustrate the accumulated sentiment (+/−) the word collected in the domain under study

all above three methods
uses language parsing
technology

5 Experimental setup

In this section, we will evaluate the proposed approach (DidaxTo) and compare it against other state-of-the-art methods. The dataset was created by automatically crawling web data using tools developed by our research team.

5.1 Dataset

The dataset was created by extracting review data from a popular, e-shop.[2] A total of 3866 reviews were extracted referring to 202 different products, belonging to 7 domains: Bedroom-beds (28 products/373 reviews), Book-shelves (48/600), Coffee-tables (23/631), Kitchen-cabinets (16/285), Living-room-chairs (17/340), Tablets (38/978), TVs (32/659). For the extraction process, we used DEiXTo[3] [15], a free and open-source web content extraction suite. We selected a diverse set of domains in order to study if the evaluated algorithms are able to identify *domain-specific opinion words*. The dataset as well as other assets used in this work is available at http://deixto.com/didaxto.

5.2 Methods

Here, we present the methods included in the comparative study.

- A lexicon extraction method presented in [35]. We will refer to this method as MOWOT. The method suggests a two-stage framework for mining opinion words and opinion targets. In the first stage, the approach utilizes a Sentiment Graph Walking algorithm and the Mini-par tool [19] as a dependency tree parser to extract <op-word - target> pairs. Then, random walking is employed to estimate the confidence of each candidate. In the second stage, a self-learning strategy is used that refines the results of the first stage.
- NiosTo: It is the algorithm and GUI presented in [1]. It classifies opinionated content and discovers a domain-specific lexicon following similar processes as the ones presented in Sect. 4. The main difference is that NiosTo utilizes manually defined patterns whereas in our most recent approach, DidaxTo, the patterns are *discovered* from the corpus.
- DidaxTo: It is the algorithm and GUI implementation of this paper. Just as we presented at the previous sections the method is unsupervised. For all the following experiments, we used the default DidaxTo parameters. The values can be observed in the DidaxTo GUI under the tab “options”.

5.3 Explicit and implicit evaluation

We used two approaches to explicitly and implicitly evaluate the methods described in the previous section:

- **Explicit evaluation** We utilized the extracted lexicons that were produced by DidaxTo, NiosTo and MOWOT. After the three methods extracted the opinion words, we investigated directly the quality of this outcome. We evaluated each extracted word individually in terms of (a) how well their nature as opinion words was correctly identified, (b) if the algorithms successfully identified the orientation of the opinion words (positive, negative) and (c) if the words can be considered as domain specific or not. Three independent annotators undertook the above three evaluation tasks. The evaluators followed some generic classification rules: An extracted word is annotated as Opinion Word if it expresses sentiment and can be applied in many domains. Examples of such Opinion Words are *able, bad, cheap*. An extracted word is classified as “Domain Specific” if it can only be used in one domain or in a limited number of domains, e.g. *aggravating, bland, hassle*. Despite the subjective character of the classification task, our final results

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

[3] http://deixto.com/.

were based on the majority voting of the three annotators. For evaluating the algorithms, we compared the human annotation against the outcome of the three approaches.

– **Implicit evaluation** The rationale behind the implicit evaluation is the following. We assume that no ground truth is available (i.e. external knowledge about which words are opinion words, domain-specific opinion words). In order to evaluate the quality of the words extracted by each method, we apply the following process. We utilize the extracted words of each method to apply sentiment analysis on a separate set of reviews. We then assess the quality of the words based on the quality of the sentiment analysis. Since opinion words are primarily utilized in sentiment analysis, we assume that better (domain) opinion words will provide more accurate sentiment analysis. The quality assessment of the sentiment analysis is executed in two steps. First, we apply a dictionary-based sentiment analysis by utilizing the dictionaries obtained from each approach. Then we compare this outcome with the number of "stars" the user has assigned to each item. We have this information in our datasets. The assumption is that if the words are correct, then the sentiment analysis will agree with the user rating (stars). This assumption of course is not always true (e.g. a user might provide a bad review but a great number of stars). However, since this does not happen very often, we consider this approach a good alternative for evaluating the methods, especially when no ground truth knowledge is available.

  Note that for NiosTo and DidaxTo we utilized the polarity obtained from the extracted patterns—and not only based on the polarity of the discovered opinion words (see Sect. 4—pattern-based polarity assignment vs dictionary-based polarity assignment) since these two methods provide such capability. For MOWOT, we apply a dictionary-based sentiment analysis.

6 Results and discussion

In this section, we present the results of our experiments and discuss the attributes and potential of each approach. The overall outcome is that DidaxTo outperforms the competitive approaches in its ability to extract opinion words and, more importantly, domain-specific opinion words. Moreover, the patterns that are discovered by DidaxTo lead to effective sentiment classification.

This advantage can be attributed to the fact that DidaxTo is not dependent on a set of predefined patterns. On the contrary, it discovers patterns for each domain independently. NiosTo performs quite well in many domains, but because it utilizes a small number of patterns, it falls short in adapting to each domain successfully. Overall it presents the second best performance.

Table 15 presents the results of the explicit evaluation, whereas Fig. 8 summarizes these results illustrating opinion/domain word accuracy for every method and domain. Results show that the accuracy of DidaxTo exceeds the accuracy of the rest of the methods in *all* domains. This underlines the ability of DidaxTo to extract opinion/domain words.

The results of the implicit evaluation are presented in Fig. 9. In Fig. 9a, we observe that DidaxTo's pattern-based evaluation performs better than NiosTo in all datasets. This suggests that the DidaxTo approach is able to adapt to each individual domain successfully. Please recall that MOWOT does not have the ability to provide pattern-based sentiment analysis; hence, we only present results on the dictionary-based sentiment classification.

Table 15 Explicit evaluation results of DidaxTo, NiosTo and MOWOT

	Domain (number of reviews)	DidaxTo			NiosTo			MOWOT		
		Correctly identified			Correctly identified			Correctly identified		
		Discovered words	Opinion words	Domain words	Discovered words	Opinion words	Domain words	Discovered words	Opinion words	Domain words
1	Bedroom-beds (373)	700	381	231	951	400	153	300	129	36
2	Book-shelves (600)	697	418	261	1037	464	242	346	154	114
3	Coffee-tables (631)	608	486	312	1008	524	320	359	167	108
4	Kitchen-cabinets (285)	410	247	72	703	230	70	245	84	17
5	Living-room-chairs (340)	500	295	95	743	315	103	292	125	40
6	Tablets (978)	1297	621	493	1660	713	550	351	174	115
7	TVs (659)	556	331	261	870	341	301	291	140	112

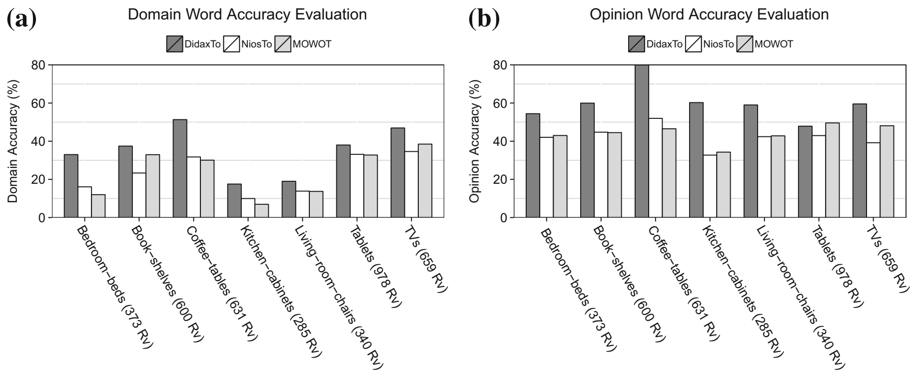

Fig. 8 Comparison of DidaxTo, NiosTo and MOWOT in the explicit evaluation. Word accuracy is the percentage of words that were correctly classified by each method, according to the human annotation. **a** Domain Word Accuracy Evaluation of DidaxTo, NiosTo and MOWOT. **b** Opinion Word Accuracy Evaluation of DidaxTo, NiosTo and MOWOT

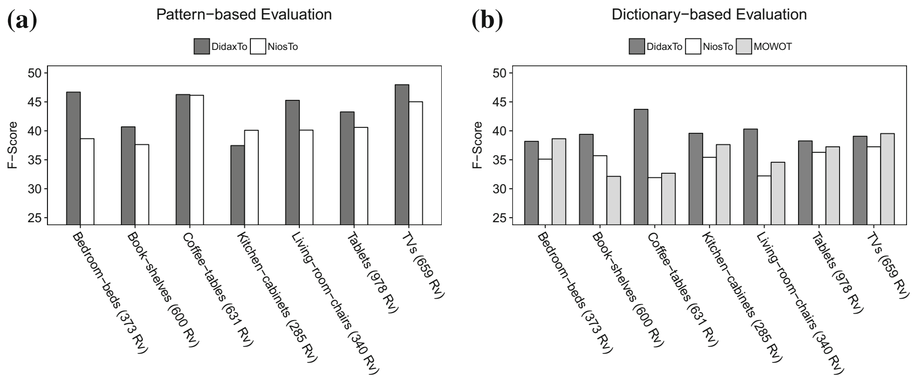

Fig. 9 Comparison of DidaxTo, NiosTo and MOWOT in the Implicit Evaluation. This sentiment analysis task is based on the comparison of the output of each method with the rating of the users. y axis is the Average F-score. In DidaxTo and Niosto, we can apply both pattern-based (**a**) and dictionary-based sentiment analysis (**b**). MOWOT can provide only dictionary-based sentiment analysis. **a** Pattern-based evaluation of DidaxTo, NiosTo. **b** Dictionary-based evaluation of DidaxTo, NiosTo and MOWOT

Figure 10 illustrates the length of patterns that are utilized by DidaxTo and NiosTo. We observe that DidaxTo utilizes patterns of different sizes, whereas NiosTo almost solely uses patterns of small length (2 and 3 labels). This result indicates that NiosTo only captures simple dependencies and hence its pattern-based capabilities are limited.

In Fig. 9b, we observe that the dictionary-based sentiment classification of DidaxTo outperforms the rest of the approaches in all domains. Moreover, in some domains like in “coffee-tables”, “living-room-chairs” the difference is substantial. The second best performance is provided by MOWOT with the exception of “book-shelves” domain.

By Comparing the two figures, Fig. 9a, b we note the advantage of the pattern-based evaluation since both DidaxTo and Niosto present a better performance in pattern-based sentiment classification. This fact justifies the usability of the dynamic (DidaxTo) or static (Niosto) syntactic patterns that better capture the sentiment of the user.

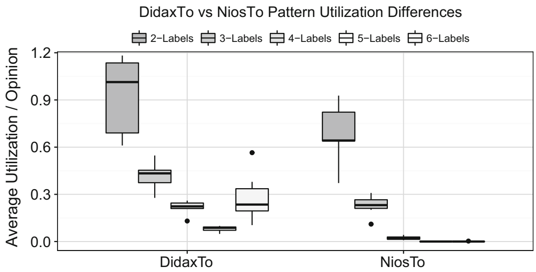

Fig. 10 Length of patterns utilized by DidaxTo and NiosTo. For each pattern length (2-labels, 3-labels, etc.), we have calculated how many times they are utilized (matched) per opinion. DidaxTo uses patterns of multiple lengths, including many patterns of length 6

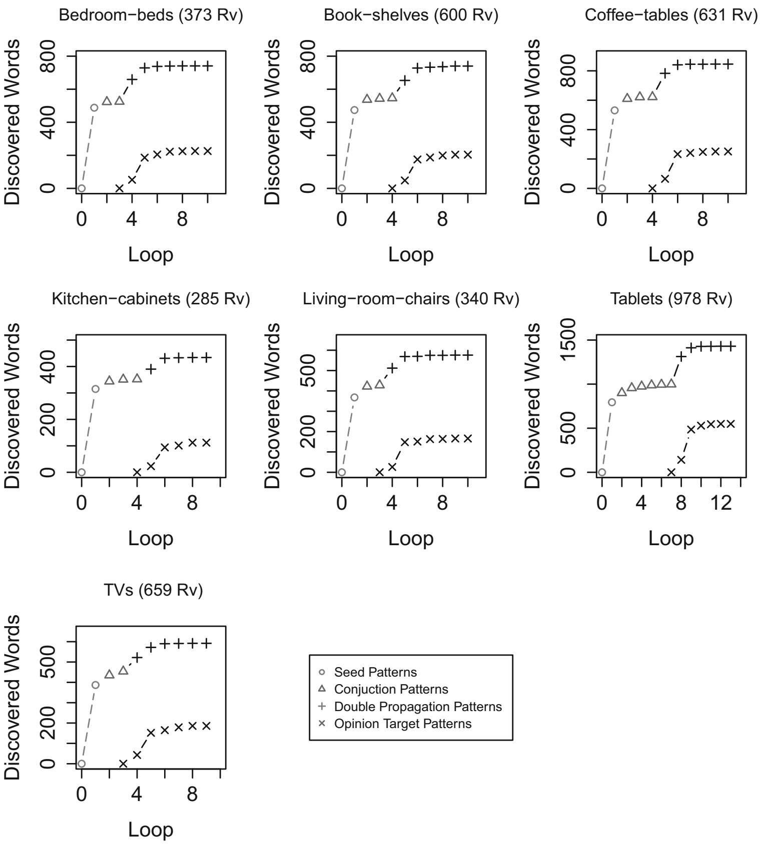

Fig. 11 DidaxTo—contribution of each pattern extraction step to the number of discovered opinion words

Finally, in Fig. 11 we observe the contribution of each pattern extraction step to the discovery of opinion words. Since some steps are repeatedly executed *in loops* (e.g. double propagation), we demonstrate the number of additional extracted opinion words. We observe that all steps have contribution and that the discovery process converges after a few repetitions (loops). In a separate line, with x-marks, we demonstrate the process of discovering opinion target words. Again the algorithm convergences after only a small number of repetitions.

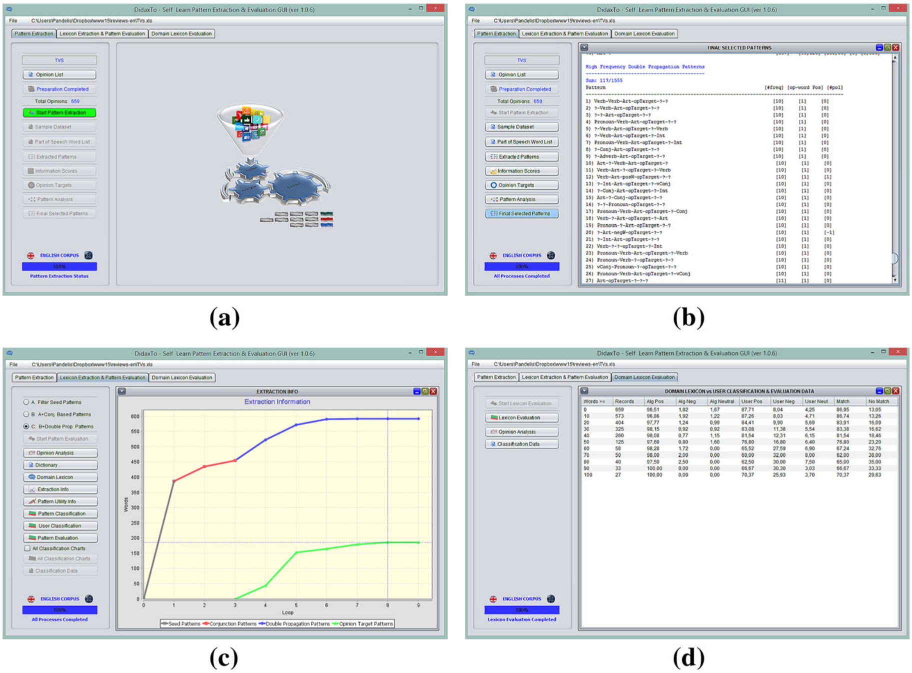

Fig. 12 The `DidaxTo` graphical user interface. **a** GUI Welcome screen and ready for pattern extraction. **b** Final extracted patterns. **c** Pattern extraction Info. **d** Domain-Lexicon evaluation data

7 Application

We have integrated the above methodology in a graphical user interface (GUI) that we make available online[4] for research purposes along with some sample datasets. A user can utilize the system to extract opinion words from a text corpus, apply sentiment analysis, visualize results and tune system's parameters. Figure 12a presents the welcome screen of the GUI. The user selects a file with opinionated text from a drop down File menu and the application makes the necessary preprocessing steps. The green button *Start Pattern Extraction* in Fig. 12a initiates the Pattern Discovery process. When the analysis is complete, the user can review all the extraction details and intermediate results. Figure 12b presents a preview of the discovered patterns, and Fig. 12c presents pattern extraction information for various extraction steps. The user can also evaluate the outcome of the method with dictionary-based or pattern-based sentiment analysis. Figure 12d presents results of the extracted dictionary (implicit) evaluation classification on the corpus under study.

8 Conclusions and future work

This paper proposes a novel multistage approach that extracts patterns to be utilized in opinion mining and sentiment analysis tasks. The approach extracts three set of classification patterns: Seed, Conjunction based and Double Propagation. The overall advantages of the proposed

[4] http://deixto.com/didaxto.

technique (DidaxTo) are that it is unsupervised and can successfully identify *domain-specific* opinion words. Another benefit is that the discovered patterns can be used for sentiment classification. We assess the effectiveness of the proposed approach with an implicit and explicit evaluation procedure. The outcome of the experimental evaluation is that DidaxTo successfully identifies opinion and domain-specific opinion words outperforming two state-of-the-art techniques. The approach is integrated into an easy-to-use graphical user interface that we make available online for research purposes.

There are multiple unexplored paths in this line of research. The efficient extraction of opinion words and targets, in addition to sentiment analysis, can also aid toward opinion summarization. Our method inherently calculates significance scores for each opinion word and target, and hence, we can utilize information regarding the importance of each term. The only difference is that the documents that DidaxTo should analyze are not the full corpus of a domain but only reviews related to a single product or entity. Additionally, representative patterns can be selected in order to present a summary in the form of natural language. Another research direction is the exploitation of deep learning and word embeddings. Similar to the corpus- and thesaurus-based approaches (see Sect. 2), the use of embeddings can be explored in order to extract useful relationships among terms and extend a small seed opinion word list.

Acknowledgements This project received funding from the European Union Horizon 2020 Programme (Horizon2020/2014-2020), under Grant Agreement 688380.

References

1. Agathangelou P, Katakis I, Kokkoras F, Ntonas K (2014) Mining domain-specific dictionaries of opinion words. In: Web information systems engineering—WISE 2014—15th international conference, Thessaloniki, Greece, October 12–14, 2014, Proceedings, Part I, pp 47–62
2. Amiri H, Chua T-S (2012) Mining slang and urban opinion words and phrases from cqa services: an optimization approach. In: Proceedings of the fifth international conference on web search and web data mining, WSDM 2012, Seattle, WA, USA, February 8–12, 2012, pp 193–202
3. Baccianella S, Esuli A, Sebastiani F (2010) Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the international conference on language resources and evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta
4. Broß J, Ehrig H (2013) Automatic construction of domain and aspect specific sentiment lexicons for customer review mining. In: 22nd ACM international conference on information and knowledge management, CIKM'13, San Francisco, CA, USA, October 27–November 1, 2013, pp 1077–1086
5. Ding X, Liu B (2007) The utility of linguistic rules in opinion mining. In: SIGIR 2007: proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, Amsterdam, The Netherlands, July 23–27, 2007, pp 811–812
6. Esuli A, Sebastiani F (2006) Sentiwordnet: a publicly available lexical resource for opinion mining. In: In Proceedings of the 5th conference on language resources and evaluation (LREC06), pp 417–422
7. Esuli A, Sebastiani F (2007) Pageranking wordnet synsets: an application to opinion mining. In: ACL 2007, proceedings of the 45th annual meeting of the association for computational linguistics, June 23–30, 2007, Prague, Czech Republic
8. Guo J, Lu Y, Mori T, Blake C (2015) Expert-guided contrastive opinion summarization for controversial issues. In: WWW (Companion Volume). ACM, pp 1105–1110
9. Hatzivassiloglou V, McKeown K (1997) Predicting the semantic orientation of adjectives. In: 35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics, proceedings of the conference, 7–12 July 1997, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, pp 174–181
10. 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, Seattle, Washington, USA, August 22–25, 2004, pp 168–177

11. Kamps J, Marx M, Mokken RJ, de Rijke M (2004) Using wordnet to measure semantic orientations of adjectives. In: Proceedings of the fourth international conference on language resources and evaluation, LREC 2004, May 26–28, 2004, Lisbon, Portugal
12. Kanayama H, Nasukawa T (2006) Fully automatic lexicon expansion for domain-oriented sentiment analysis. In: EMNLP 2007, proceedings of the 2006 conference on empirical methods in natural language processing, 22–23 July 2006, Sydney, Australia, pp 355–363
13. Kim HD, Castellanos M, Hsu M, Zhai CX, Dayal U, Ghosh R (2013) Compact explanatory opinion summarization. In: CIKM. ACM, pp 1697–1702
14. Kim HD, Castellanos MG, Hsu M, Zhai CX, Dayal U, Ghosh R (2013) Ranking explanatory sentences for opinion summarization. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, SIGIR '13, New York, NY, USA. ACM, pp 1069–1072
15. Kokkoras F, Ntonas K, Bassiliades N (2013) Deixto: a web data extraction suite. In: Proceedings of the 6th Balkan conference in informatics, BCI '13, New York, NY, USA. ACM, pp 9–12
16. Lau RY-K, Lai CL, Bruza P, Wong K-F (2011) Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. In: Proceedings of the 20th ACM conference on information and knowledge management, CIKM 2011, Glasgow, United Kingdom, October 24–28, 2011, pp 2457–2460
17. Li C, Xu B, Wu G, He S, Tian G, Zhou Y (2015) Parallel recursive deep model for sentiment analysis. In: Cao T, Lim EP, Zhou ZH, Ho TB, Cheung D, Motoda H (eds) Advances in knowledge discovery and data mining. PAKDD 2015. Lecture notes in computer science, vol 9078. Springer, Cham
18. Lim KW, Buntine WL (2014) Twitter opinion topic model: extracting product opinions from tweets by leveraging hashtags and sentiment lexicon. In: CIKM. ACM, pp 1319–1328
19. Lin D (1994) PRINCIPAR—an efficient, broad-coverage, principle-based parser. In: 15th international conference on computational linguistics, COLING 1994, Kyoto, Japan, August 5–9, 1994, pp 482–488
20. Liu K, Xu L, Zhao J (2013) Syntactic patterns versus word alignment: extracting opinion targets from online reviews. In: Proceedings of the 51st annual meeting of the association for computational linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, volume 1: long papers, pp 1754–1763
21. Lu Y, Castellanos M, Dayal U, Zhai CX (2011) Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: Proceedings of the 20th international conference on World Wide Web, WWW 2011, Hyderabad, India, March 28–April 1, 2011, pp 347–356
22. Peng W, Park DH (2011) Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In: Proceedings of the fifth international conference on weblogs and social media, Barcelona, Catalonia, Spain, July 17–21, 2011
23. Popescu A-M, Etzioni O (2005) Extracting product features and opinions from reviews. In: HLT/EMNLP 2005, human language technology conference and conference on empirical methods in natural language processing, proceedings of the conference, 6–8 October 2005, Vancouver, BC, Canada
24. Qiu G, Liu B, Bu J, Chen C (2009) Expanding domain sentiment lexicon through double propagation. In: Proceedings of the 21st international joint conference on artifical intelligence, IJCAI'09, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc, pp 1199–1204
25. Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. Comput Linguist 37(1):9–27
26. Rao Y, Lei J, Wenyin L, Li Q, Chen M (2014) Building emotional dictionary for sentiment analysis of online news. World Wide Web 17(4):723–742
27. Severyn A, Moschitti A (2015) Twitter sentiment analysis with deep convolutional neural networks. In: SIGIR. ACM, pp 959–962
28. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts CP (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP
29. Thonet T, Cabanac G, Boughanem M, Pinel-Sauvagnat K (2016) VODUM: a topic model unifying viewpoint, topic and opinion discovery. In: Ferro N et al (eds) Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science, vol 9626. Springer, Cham
30. Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting of the association for computational linguistics, July 6–12, 2002, Philadelphia, PA, USA, pp 417–424
31. Wang S, Chen Z, Liu B (2016) Mining aspect-specific opinion using a holistic lifelong topic model. In: WWW. ACM, pp 167–176
32. Weichselbraun A, Gindl S, Scharl A (2011) Using games with a purpose and bootstrapping to create domain-specific sentiment lexicons. In: Proceedings of the 20th ACM conference on information and knowledge management, CIKM 2011, Glasgow, UK, October 24–28, 2011, pp 1053–1060

33. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT/EMNLP 2005, human language technology conference and conference on empirical methods in natural language processing, proceedings of the conference, 6–8 October 2005, Vancouver, BC, Canada
34. Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase dependency parsing for opinion mining. In: Proceedings of the 2009 conference on empirical methods in natural language processing, EMNLP 2009, 6–7 August 2009, Singapore. A meeting of SIGDAT, a Special Interest Group of the ACL, pp 1533–1541
35. Xu L, Liu K, Lai S, Chen Y, Zhao J (2013) Mining opinion words and opinion targets in a two-stage framework. In: Proceedings of the 51st annual meeting of the association for computational linguistics, ACL 2013, 4–9 August 2013, Sofia, Bulgaria, volume 1: long papers, pp 1764–1773

Pantelis Agathangelou holds an M.Sc. in Information Systems from the Faculty of Pure and Applied Sciences, Open University of Cyprus. His research interests include Data Mining, Sentiment Analysis, Pattern Classification, Mining Social Networks, Recommender Systems and Artificial intelligence. He has co-developed DidaxTo (http://deixto.com/didaxto/) a tool that implements an unsupervised approach for discovering patterns that will extract a domain-specific dictionary from product reviews. He is currently a research associate with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens. e-mail: pandelisagathangelou@gmail.com.

Ioannis Katakis is a senior researcher at the National and Kapodistrian University of Athens working on EU-funded research projects as a Quality Assurance Coordinator. His research interests include Mining Social and Web Data, Knowledge Discovery from Text and Urban Data Streams, Multi-label Learning, Adaptive and Recommender Systems. He has published papers in International Conferences and Scientific Journals related to his area of expertise, organized three workshops, edited four special issues and is an Editor at the journal Information Systems. More information can be found at his personal web page: http://www.katakis.eu.

Ioannis Koutoulakis holds a B.Sc. in Computer Science from the Department of Informatics and Telecommunications, University of Athens, Greece. His research interests include machine learning for data mining and information extraction, artificial intelligence and human activity recognition in health applications. He is currently an M.Sc. student in Machine Learning in the Department of Computer Science, University of Bristol, UK. e-mail: koutoulakis.ioannis@gmail.com/ik16725@bris.ac.uk.

Fotis Kokkoras holds an M.Sc. in Artificial Intelligence from the Department of Computer Science, University of Bristol, UK, and a Ph.D. in Intelligent (web) Information Extraction from the Department of Informatics, Aristotle University, Greece. His research interests include web data and information extraction, web data APIs, linked data, web intelligence and artificial intelligence. He has co-developed DEiXTo (http://deixto.com) a web content extraction framework used worldwide and has participated in many research and development projects. He is currently a lecturer in the Department of Computer Science and Engineering, at TEI of Thessaly, Greece. e-mail: fkokkoras@teilar.gr.

Dimitrios Gunopulos is a Professor at the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece. He received his Ph.D. from Princeton University in 1995 and was a Researcher in IBM Almaden Research Center and a Professor at the University of California, Riverside. His research is in the areas of Data Mining, Web Mining, Data Science, Databases, Sensor and Peerto-Peer systems, and Algorithms. He has developed clustering, classification and indexing techniques for high-dimensional and trajectory data, data analysis tools for streams and sensor networks, and on web and document analysis problems. He has co-authored over 50 journal and 100 conference papers, 7 patents and 1 patent application, 20 book chapters, and a book.