Mining Domain-Specific Dictionaries of Opinion Words

Pantelis Agathangelou[1], Ioannis Katakis[2]

Fotios Kokkoras[3], Konstantinos Ntonas[4]

[1] Open University of Cyprus - pandelis.agathangelou@st.ouc.ac.cy
[2] National and Kapodistrian University of Athens - katak@di.uoa.gr
[3] Technological Educational Institute of Thessaly - fkokkoras@teilar.gr
[4] International Hellenic University - k.ntonas@ihu.edu.gr

Abstract. The task of opinion mining has attracted interest during the last years. This is mainly due to the vast availability and value of opinions on-line and the easy access of data through conventional or intelligent crawlers. In order to utilize this information, algorithms make extensive use of word sets with known polarity. This approach is known as dictionary-based sentiment analysis. Such dictionaries are available for the English language. Unfortunately, this is not the case for other languages with smaller user bases. Moreover, such generic dictionaries are not suitable for specific domains. Domain-specific dictionaries are crucial for domain-specific sentiment analysis tasks. In this paper we alleviate the above issues by proposing an approach for domain-specific dictionary building. We evaluate our approach on a sentiment analysis task. Experiments on user reviews on digital devices demonstrate the utility of the proposed approach. In addition, we present `NiosTo`, a software that enables dictionary extraction and sentiment analysis on a given corpus.

1 Introduction

Sentiment analysis is the task of extracting valuable, non-trivial knowledge from a collection of documents containing opinions. Most of the times, extracted knowledge represents a summary of the opinions expressed in the collection. Opinions can refer to products, services or even political figures. Advanced algorithms are able to discover opinions of multiple features for the entity under investigation. These techniques are applied to data extracted from various sources like discussion boards, social networks, blogs or video sharing networks.

Sentiment analysis is utilized in different levels of granularity: at sentence-level, document-level or corpus-level. A key-role to all these levels play the so-called list of "opinion-words". This is a dictionary containing terms of known polarity. In most cases, the list is dual. A *positive* word-list ("beautiful", "astonishing") is coupled with a *negative* word-list ("ugly", "slow"). Opinion dictionaries are critical to various steps of sentiment analysis since algorithms depend their initialization, enhancement and operation on them. Therefore, high-quality dictionaries are required for these type of analytics.

Such dictionaries are already available for the English language [4]. Many of them are manually constructed. However, the availability of opinion word-lists for less popular languages is very limited. Another issue with such lists is that they are domain dependent. For example the words 'cool' or 'low', might have different meaning in different domains. Therefore approaches that are able to extract domain-specific opinion dictionaries are necessary.

In this paper, we provide a method that mines domain-specific dictionaries given a corpus of opinions. This is a multiple-stage iterative approach that gets a small seed of generic opinion words as input and extends it with domain-specific words. It utilizes language patterns and takes advantage of a double propagation procedure between opinion words and opinion targets. The effectiveness of the approach is estimated in a sentiment analysis task. Results justify the utility of all steps of the proposed algorithm. In addition, we present a software that enables the use of the above techniques under a user-friendly interface.

The *advantages* of the proposed algorithm are: a) it is domain independent, b) it can operate with a very small initial seed-list, c) it is unsupervised and, d) it can operate in multiple languages provided the proper set of patterns. The *contribution* of this work can be summarized in the following points.

- Introduces a novel resource-efficient approach for building domain-specific opinion dictionaries.
- Provides an experimental study where all steps of the proposed algorithm are evaluated through a sentiment analysis task.
- Provides a dataset of opinions in the Greek language containing user reviews. This dataset can be exploited in various opinion mining tasks.
- Offers `NiosTo`, a free application that integrates opinion dictionary discovery and sentiment analysis tasks.

The rest of the document is structured as follows. Section 2 summarizes and highlights related work. In Section 3, we outline our approach and provide detailed description of all steps. After that, the experimental evaluation is presented (Section4) followed by results and discussion (Section 4.2). The reader can learn about the basic feature-set of the `NiosTo` software at Section 5, while the last section highlights significant conclusions and suggests future work.

2 Related Work

In [1] a probabilistic method is presented that builds an opinion word lexicon. The method uses a set of opinion documents which is used as a biased sample and a set of relevant documents as a pool of opinions. In order to assess the effectiveness of the algorithm a dictionary made up of $8K$ words is used, built by [9, 10]. Certain probabilistic functions such as Information Content, Opinion Entropy and Average Opinion Entropy are used as extraction tools. The method is based upon the observation that nouns contain high information value, while adjectives, adverbs and verbs (usually opinion words) provide additional information to the context. Upon these observations and the probabilistic tools they extract the opinion word lexicon.

The authors of [12] tackle the problem of opinion target orientation and summarization. The method uses an opinion lexicon [4] from WordNet. A list of content dependent opinion words such as nouns, verbs and word phrases that are joined together is utilized. The algorithm uses a score function, which is a formula that calculates opinion target orientation, by exploiting coexistence of opinion words and opinion targets in a sentence and the variance of distance among them. Linguistic patterns and syntactic conventions are used in order to boost the efficiency of the proposed method.

[3] proposes an unsupervised lexicon building method for the detection of "polar clauses" (clauses that can be classified as positive or negative) in order to acquire the minimum syntactic structures called "polar atoms" (words or phrases that can be classified as positive or negative opinion modifiers). This part of process includes a list of syntactic patterns that helps the identification of propositional sentences. Moreover the method uses an opinion lexicon and statistical metrics such as coherent precision and coherent density in order to acquire true polar atoms from fake ones.

The authors of [7] exploit a model called partially supervised word alignment, which discovers alignment links between opinion targets and opinion modifiers that are connected in bipartite graph. Initially some high precision low recall syntactic patterns are used as training sets for generating initial partial alignment links. Then these initial links will be feeded into the alignment model. The selection of opinion target candidates is based upon a factor called confidence. Candidates with higher confidence will be extracted as the opinion targets.

Our approach combines various components of the above methods, refines them and introduces new processes to overcome their disadvantages. We propose a multi-stage approach that includes conjunction based extraction and double propagation. The latter is applied more than once. However, in parallel with word extraction we employ *polarity disambiguation* after every step in order to identify the sentiment of the newly discovered words. Finally, we utilize a word validation process that takes advantage of opinion-words - opinion-target relationships by introducing parameters that improve retrieval performance.

3 Our Approach

The approach comprises of a series of steps that gradually detect opinion words. Each step creates a pool of opinion words that will constitute the feed for the next detection step. More specifically the construction of the proposed algorithm can be summarized in the following steps: 1) Opinion Preprocessing 2) Auxiliary List Preparation, 3) Seed Import and Filtered Seed Extraction, 4) Conjunction Based Extraction, 5) Double Propagation, and 6) Opinion Word Validation. In the following subsections, we present the above steps in more detail. Figure 1 presents an overview of the proposed approach.

Opinion Preprocessing At this step the designed algorithm receives user opinions in raw form. We implement some form of preprocessing in order to filter-out

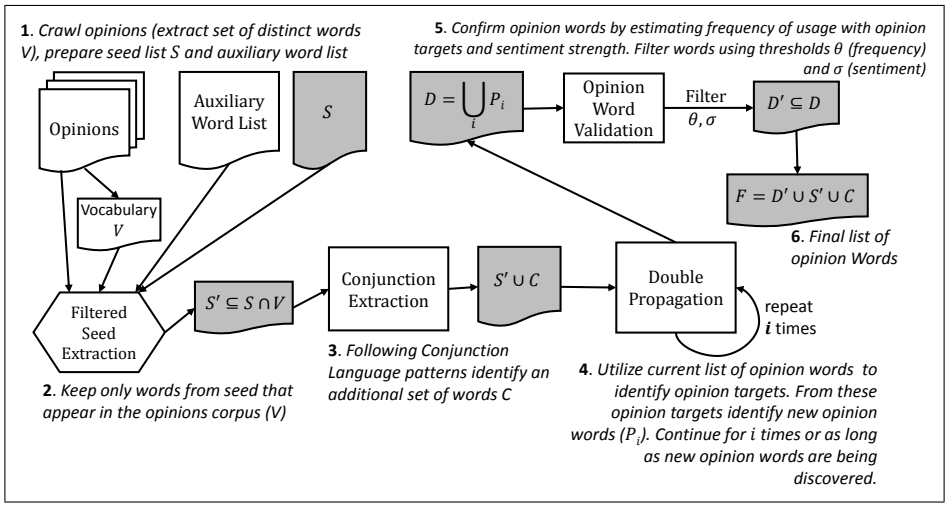


Fig. 1: Overview of the Proposed Approach. Gray shaded items represent the opinion word list that is improved in each step. Please follow the numbers (1-6).

noise. Sentence splitting is a critical step in this module (opinion delimitation) since double propagation takes into account neighbourhood sentences in order to propagate sentiment. Additionally in order to increase the efficiency of the extraction process we have adopted an on-line stemmer engine for the Greek Language[5]. At this step the set of distinct words V is constructed.

Auxiliary List Preparation This is a particularly crucial step. Auxiliary word list comprises a series of word sets like articles, verbs, comparatives, conjunctions, decreasers (e.g. "less"), increasers (e.g. "extra"), negations (e.g. "not") and pronouns. These words will constitute a main feed of the algorithm. The proposed approach utilizes this seed in the construction of all extraction patterns.

3.1 `Seed` Import and `Filtered Seed` Extraction

The initial `Seed` S of the system is a set opinion words with known polarity (e.g. "bad", "ugly", "wonderful", etc) [4, 11]. The `Seed` is *generic*, meaning that it is not domain-specific. The `Filtered Seed` S' is the set of `Seed` words that also appeared in the collection of opinion documents $S' = S \cap V$. In other words we filter-out words from the `Seed` that don't appear in the corpus. In `Seed` list, the polarity of each word is provided. However, depending on the way the word is used, it might alter its polarity ("this phone is definitely not lightweight"). Hence, we apply a step of *polarity disambiguation* using a set of language patterns. Some examples of polarity patterns are presented in Table 1. Word abbreviations stand for {fut}: future word, {pos}: positive word, {neg}: negative word, {conj}:

[5] `http://deixto.com/greek-stemmer/`

Table 1: Examples of patterns used for sentiment disambiguation

						Pol							Pol
	{fut}	{?}	{fut}	{verb}	{pos}	-1		{neg}	{verb}	{conj}	{comp}	{neg}	+1
	{neg}	{verb}	{conj}	{comp}	{pos}	-1		{neg}	{fut}	{art}	{?}	{neg}	+1
	{neg}	{pron}	{art}	{pos}	{neg}	+1		{art}	{neg}	{verb}	{decr}		+1
	{decr}	{comp}	{pos}			-1	e.g.	the	noise	has	depleted		
e.g.	little	more	useful					{neg}	{verb}	{neg}			+1
	{neg}	{pos}				-1		{pos}	{neg}				+1
	{pos}					+1		{neg}					-1

conjunction, {art}: article, {decr}: decreaser, {comp}: comparative word, {? }: any word. In total, we've used 21 positive and 12 negative polarity patterns. At the end of this process we have a pool of newly discovered opinion words along with their polarity. Algorithm 1 and 2 provide the logic of this step.

Algorithm 1: Filter seed extraction process

input : List of Opinions, initial opinion Lexicon
output: List of Extracted Opinion Words {FilteredSeed}

```
extractFilteredSeedOpinionWords()
foreach Opinion in Opinions do
    foreach sentence in Sentences do
        foreach word in sentence do
            if opinion word then
                // Get opinion word orientation;
                Orientation←getOpinionWordOrientation(w, p, s)
                if new opinion word then
                    // Add found opinion word & orientation
                    FilteredSeed←add(opinion word, Orientation)
                else
                    // Add orientation only
                    FilteredSeed←addOrientation(opinion word, Orientation)
Return FilteredSeed
```

3.2 Conjunction-Based Extraction of Opinion Words

At this step we exploit the assumption of *sentiment consistency* [2] that applies in conjuncted words (e.g. "lightweight and well-built device"). That way our algorithm discovers new opinion words by making use of certain conjunction patterns that have been selected to fit the sentiment consistency theories. Table 2 provide examples of such language patterns. Word abbreviations stand for {cpos}: candidate positive word {cneg}: candidate negative word.

Algorithm 2: Filter seed & Conjunction based polarity exploration

input : Polarity patterns
output: Word orientation

```
getOpinionWordOrientation(word, position, sentence)
if match polarity pattern then
    Return pattern orientation
else
    Return default orientation
```

Table 2: Examples of positive and negative word conjunction dependencies

Positive						Negative					
	{pos}	{conj}	{comp}	{art}	{cpos}		{neg}	{conj}	{fut}	{verb}	{cneg}
	{pos}	{conj}	{neg}	{cpos}			{neg}	{conj}	{incr}	{cneg}	
e.g.	thin	and	not	sticky		e.g.	expensive	and	too	small	
	{pos}	{conj}	{cpos}				{neg}	{conj}	{cneg}		

For this step we have utilized 6 positive and 4 negative extraction patterns. Candidate opinion words of this step also go through a polarity disambiguation process like the previous step. At the end of this process we have an extended list of opinion words $S' \cup C$ where C is the list of opinion words extracted from this step. Algorithm3 and 4 provide the logic of this step.

3.3 Double Propagation Extracted Opinion Words

This process of detecting new opinion words follows the theory of double propagation [5, 8].The assumption is that each opinion word has an opinion target attached to it (e.g. "nice screen"). Here, there is a direct connection (opinion word, 'nice')→(opinion target, 'screen'). Based on double propagation and using the current list of opinion words, we are able to identify opinion targets. Using this set of opinion targets we are able to extract new opinion words following the same logic. So this process is repetitive. It iterates i times or as long as new opinion words are discovered. At each step P_i opinion words are discovered. In the end of this step we end up with list $D = S' \cup C \bigcup_i P_i$. Our experiments indicate that double propagation is repeated 4 to 8 times. For the reverse step of double propagation we use a set of patterns that can be seen in 3. The word abbreviations stand for: {copt}: candidate opinion target, {copw}: candidate opinion word. In total we have used 6 such word patterns.

The newly discovered opinion words are going through polarity disambiguation. At this step we take advantage of of intra-sentential and inter-sentential sentiment consistency [3]. The *intra-sentential* consistency suggests that if there are other opinion words in a sentence with known orientation, then, the newly found word will get the accumulated sentiment of these words. When there are no other known opinion words in the sentence, the *inter-sentential* assumption

Algorithm 3: Conjunction based extraction process

input : List of Opinions, filter seed word list, Conjunction list
output: List of Extracted Opinion Words {ConjList}

```
1  extractConjunctionBasedOpinionWords()
2  foreach Opinion in Opinions do
3      foreach sentence in Sentences do
4          foreach word in sentence do
5              if filter seed word then
6                  if next word conjunction then
7                      // Search the existence of an opinion word
8                      opinion word←getConjunctionBasedOpinionWord(w, i, s)
9                      if opinion word then
10                         // Get orientation of opinion word
11                         Orientation←getOpinionWordOrientation(w, i, s)
12                         if new opinion word then
13                             // Add found opinion word & orientation
14                             ConjList←add(opinion word, Orientation)
15                         else
16                             // Add orientation only
17                             ConjList←addOrientation(opinion word,
                               Orientation)
18 Return ConjList
```

Algorithm 4: Conjunction based opinion word extraction process

input : Conjunction based extraction patterns
output: opinion word or null

```
1 getConjunctionBaseOpinionWord(opinion word, index, sentence)
2 if conjunction based pattern then
3     Return opinion word
4 else
5     Return null
```

is applied. It suggests that users tend to follow a certain opinion orientation at succeeded sentences when forming an opinion. This way if a sentence does not have an accumulated sentiment we search at nearby sentences (up to a certain limit) and we assign to the new word, the largest sentiment score found at these nearby sentences. At the end of this process we have a pool of new opinion words and their orientation, the double propagation opinion words. Algorithms 5, 6, 7, 8 provide the details of the above processes.

3.4 Opinion Word Validation

The double propagation process makes extensive use of all possible ways to discover new opinion words, but appears to have low precision (see Experimental Section). For this reason we apply a filtering procedure, namely *opinion word*

Table 3: Double propagation opinion word dependencies

	{art}	{copt}	{pron}	{verb}	{copw
	{copw}	{conj}	{art}	{copt}	
e.g.	amazing	and	the	cost	
	{copw}	{copt}			

Algorithm 5: Double propagation extraction process

```
1 DoublePropagation()
2 while new opinion words or new opinion targets do
3 |   extractOpinionWordTargets()
4 |   extractOpinionWords()
```

validation. We employ two thresholds, the *sentiment threshold (σ)* and the *frequency threshold (θ)*. If a newly found word exceeds these two thresholds then we consider it an opinion word. The intuition behind these thresholds is the following. From the set of candidate opinion words discovered from double propagation, we consider valid opinion words only those that: a) appear more than θ times along with an opinion target *and* b) their sentiment polarity calculated (through polarity disambiguation) is larger than σ (see Equations 1 and 2).

$$w_i = \begin{cases} w_i, \text{ if } \sum_{opinion=0}^{n} (w_i \wedge t_i) \geq \theta \\ \\ \emptyset, \text{ Otherwise} \end{cases} \tag{1}$$

$$[Sent]_i = \begin{cases} [Sent]_i, \text{ if Abs}([Sent]_i) \geq \sigma \\ \\ \emptyset, \qquad \text{Otherwise} \end{cases} \tag{2}$$

Algorithm 6: Opinion target List extraction process

input : List of opinions, explicit opinion word to opinion target rules
output: List of extracted opinion targets {OpTargetList}

```
1 extractOpinionWordTargets()
2 foreach Opinion in Opinions do
3 |   foreach sentence in Sentences do
4 |   |   foreach word in sentence do
5 |   |   |   if opinion word then
6 |   |   |   |   // Explore opinion target existence
7 |   |   |   |   opinion target←getOpinionWordTarget( w, i, s)
8 |   |   |   |   if new opinion target then
9 |   |   |   |   |   OpTargetList←opinion target
10 Return OpTargetList
```

Algorithm 7: Opinion word to opinion target extraction process

input : Explicit opinion target connection rules
output: opinion target or null

```
getOpinionWordTarget(opinion word, index, sentence)
if explicit opinion target connection then
    Return opinion target
else
    Return null
```

Algorithm 8: Double propagation opinion word List extraction process

input : List of opinions, List of opinion words, List of opinion targets
output: List of Extracted double propagation Opinion words {DoublePropList}

```
extractOpinionWords()
foreach Opinion in Opinions do
    foreach sentence in Sentences do
        foreach word in sentence do
            if opinion target then
                // Explore existence of opinion word
                opinion word←getDoublePropagationOpinionWord( w, i, s)
                if opinion word then
                    // Explore Orientation of opinion word
                    Orientation←getDoublePropagationOpWordOrientation(w,
                    i, s)
                    if new opinion word then
                        // Add opinion word & orientation
                        DoublePropList←add(opinion word, Orientation)
                    else
                        // Add orientation only
                        DoublePropList←addOrientation(opinion word,
                        Orientation)
Return DoublePropList
```

Algorithm 9: Double propagation polarity extraction process

input : Double propagation extraction patterns
output: opinion word or null

```
getDoublePropagationOpWordOrientation(opinion target, index, sentence)
if opinion words in the sentence then
    // Use intra-sentential sentiment consistency
    Orientation←(sum orientation of opinion words in sentence)
else
    // Use inter-sentential sentiment consistency
    Orientation←(strongest orientation from nearby sentences)
Return Orientation
```

Where w_i stands for double propagation opinion word and t_i for opinion target word. Parameters θ and σ are user defined. The higher these thresholds are, we get higher precision and lower recall. The user can try different values of these parameters through the `NiosTo` interface.

4 Experimental Evaluation

The evaluation aims at answering the following research questions: a) How well does the proposed approach extracts opinion words?, b) What is the added value of each step of the method?, c) How useful are the extracted domain-specific opinion words lists for performing an unsupervised sentiment classification task?

Dataset Creation and Description The dataset was created by extracting review data from a popular, Greek e-shopping site[6]. A total of 4887 reviews were extracted referencing 1052 different products, belonging to 7 domains: TVs (189 products/322 reviews), Air Conditioners (105/139), Washing Machines (63/83), Cameras (122/166), Refrigerators (77/103), Mobile Phones (339/3626), Tablets (157/448). For the extraction process we used DEiXTo [6], a popular free and open source web content extraction suite. The dataset as well as other assets used in this work, are available at `http://deixto.com/niosto`.

4.1 Evaluation of Opinion Word Retrieval

In this section we evaluate the overall retrieval quality of opinion words and then study the contribution of each individual module of the proposed approach. For the evaluation of this step, we consider the initial seed as the ground truth set of opinion words. We "hide" a percentage of this ground truth from the algorithm and study how well it can discover these words. Since the initial seed is generic, at this step we evaluate the ability of the approach to extract opinion words that are not domain specific. We focus on domain-specific words in the next section.

At Table 4 we present the results of the extraction processes upon various category domains. We present a different set of results with and without using the stemmer. This set of results focuses on the impact of each step at opinion-word discovery. In brief, Conjunction-based extraction is more conservative at finding new opinion words while Double Propagation tends to discover more words. Table 5 presents the results of the algorithm evaluation upon the extraction processes in terms of precision (for Conjunction-step) and recall (for the Double Propagation step). Note that this evaluation is only indicative since the approach is evaluated in terms of ability to identify opinion words from the original seed - which is not domain-specific and only a few of them appear in the extracted opinions. Another issue, is that language patterns are not always followed by the users. Reviews are most of the times just a set of keywords put together to describe advantages and disadvantages of the devices. As expected, domains with large number of opinions (mobiles, tablets) present better precision / recall.

[6] `http://www.skroutz.gr/`

Table 4: Extracting Words using the proposed approach for various domains

			(Stemmer-out) Extracted words						(Stemmer-in) Extracted words					
Source Category	Opin	Senten Process	Filtered seed		Conj. extr.		Double prop.	Opin targ	Filtered seed		Conj. extr.		Double prop.	Opin targ
			pos	neg	pos	neg	total		pos	neg	pos	neg	total	
Televisions	322	1630	103	28	6	0	370	348	228	60	23	9	409	452
Air Conditioners	139	847	74	18	9	0	173	186	157	35	15	3	242	239
Washing Machines	83	515	46	11	4	0	119	120	103	29	8	3	153	151
Digital Cameras	166	872	79	16	6	0	172	168	170	31	8	1	188	258
Refrigerators	103	539	42	14	6	0	86	109	106	31	8	1	131	142
Mobiles	3626	20284	245	89	131	8	2633	2063	700	231	242	66	2139	2906
Tablets	448	2142	129	23	18	2	378	333	262	60	33	6	424	453

Table 5: Average Precision - Recall Metrics

Source Category	Opinions	Average Values	
		Conj. Precision	Double Prop. Recall
Televisions	322	0%	49%
Air Conditioners	139	28%	15%
Washing Machines	83	3%	5%
Cameras	166	0%	46%
Refrigerators	103	35%	10%
Mobiles	3626	19%	38%
Tablets	448	54%	35%

4.2 Evaluating Utility in Sentiment Classification

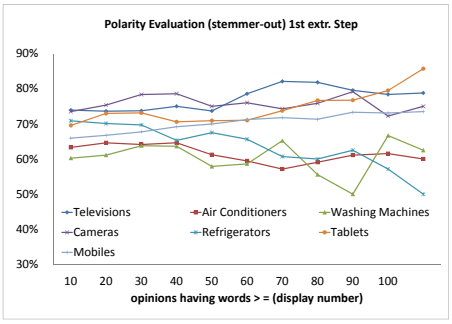
At this point we discuss the contribution of each extraction step to the sentiment classification task. To create an evaluation set we utilized the "star" ratings of user reviews. We consider an opinion positive when the user assigned 4 or 5 stars. An opinion is considered negative when the user have assigned 1 or 2 stars to the product. Table 6 presents the sentiment classification accuracy for all steps of the approach. Naturally, classification accuracy is calculated as $sent_{acc} = \frac{\#\text{correct classifications}}{\#\text{opinions}}$. In general, we observe that the accuracy of all these dictionary-based approaches can vary from $59,95\%$ to $83,09\%$. Note that these approaches are completely unsupervised, i.e. no labelled data are required. Concerning the contribution of the various steps of the approach, we observe that in most cases the double propagation step leads to an improved classification accuracy. The step of conjunction extraction has a smaller impact. The

improvement of Conjunction extraction and Double propagation over the filtered seed (generic opinion words) leads to the conclusion that the algorithm manages to identify effectively domain-specific words that aid in the task of sentiment classification. In most cases, stemming aids in classification accuracy. However, there are some domains, like refrigerators, where stemming has actually a negative impact. This can be explained by the fact that stemming unifies many different words incorrectly and identifies false opinion words.

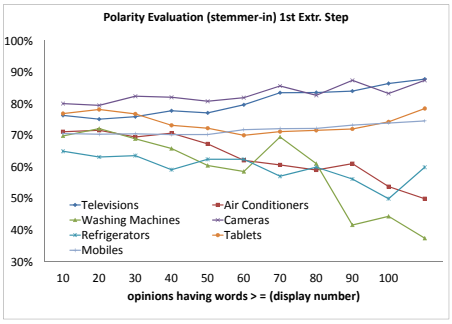
Next we study another interesting factor in our analysis which is the quality of the expressed opinion. We judge quality of opinions by terms of length (actual word count). We have observed that longer opinions are more carefully written and the 'star' rating corresponds more accurately to the expressed opinion. For example, in short casual written approaches the 'star' rating seem not to correlated well with the actual text. Hence, very short opinions will negatively affect our approach (and, in fact, any language analysis task) and the evaluation as well, since the ground truth labels (positive-negative based on 'star' rating) are inaccurate. In the following figures we present classification accuracy taking into consideration opinions of various length. In Figures 2a, 2b, 2c 2d point x_o in the x-axis presents the sentiment classification by considering only opinions with more than x_o words. This set of results confirms the results of Table 6: a) stemming has a positive impact, b) double-propagation accuracies outperform the based line (filter seed), c) Sentiment classification is more accurate in domains with more opinions. Values for more opinions with more than 50 words present higher variance since they are very few and this makes the accuracy fluctuate.

Table 6: Sentiment Classification Accuracy of Various Steps of the Approach

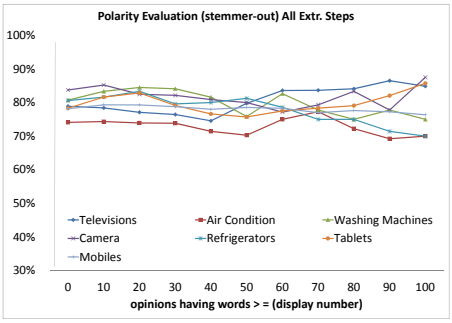
			Average polarity evaluation (stemmer out)			Classification Accuracy (stemmer in)		
Source Category	Opinions	Sentences Processed	Filtered Seed	Conj. extr.	Double prop.	Filtered Seed	Conj. extr.	Double prop.
Televisions	322	1630	77,19%	77,19%	80,71%	80,73%	80,73%	75,49%
Air Conditioners	139	847	61,47%	65,14%	72,87%	63,41%	67,08%	69,51%
Washing Machines	83	515	60,48%	60,48%	79,84%	59,13%	59,13%	62,27%
Cameras	166	872	75,75%	75,75%	81,77%	83,09%	83,09%	82,76%
Refrigerators	103	539	63,59%	63,59%	77,85%	59,95%	59,95%	66,97%
Mobiles	3626	20284	70,35%	70,39%	78,05%	71,87%	71,97%	74,96%
Tablets	448	2142	74,60%	74,60%	79,71%	74,16%	74,51%	79,73%
		Total Average:	69,06%	69,59%	78,69%	70,33%	70,92%	73,10%
		Step Contribution:		1%	12%		1%	3%



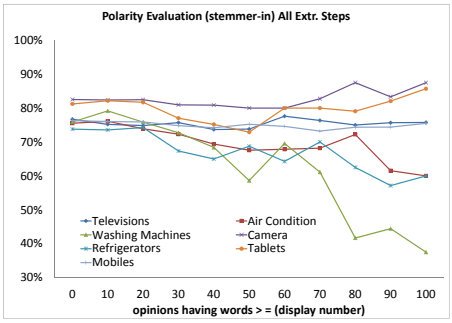
(a) Stemmer out, first step



(b) Stemmer in, first step



(c) Stemmer out, all steps



(d) Stemmer in, all steps

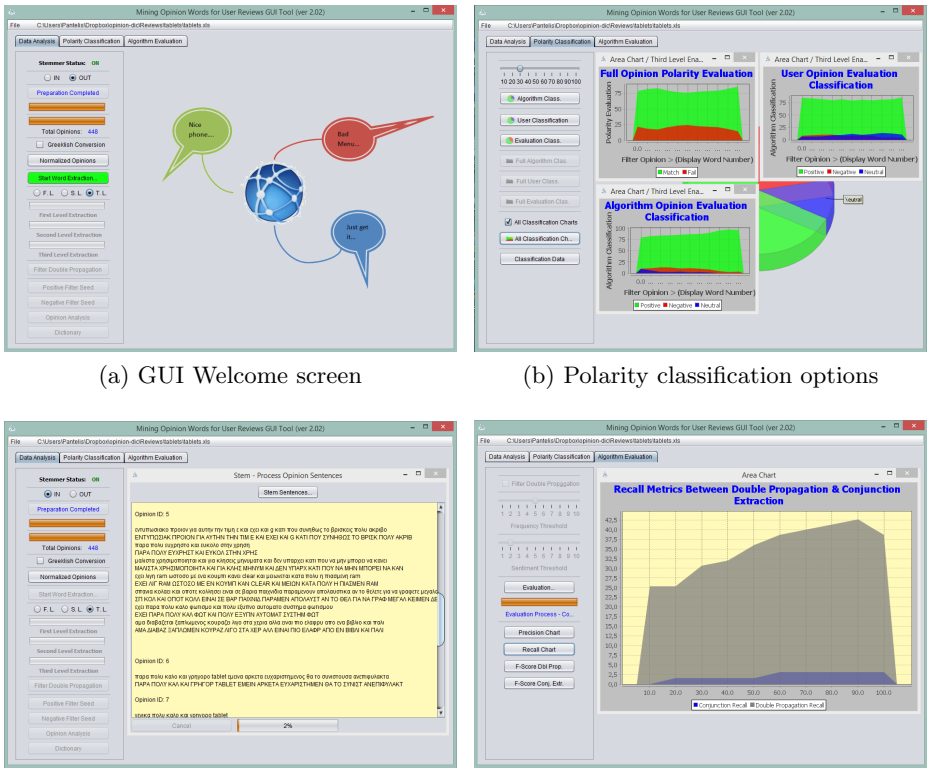
Fig. 2: Evaluation based on Sentiment Classification

5 The `NiosTo` tool

We developed a software application (`NiosTo`) that implements the above opinion word extraction algorithm as well as the dictionary-based sentiment classification. All features are accessible through an easy-to-use Graphical User Interface (see Figures 3a,3b,3c,3d). `NiosTo` takes as input opinions in csv format and presents the discovered words in each level. In addition, retrieval results as well as sentiment classification are visualized. All parameters can be changed and tuned through the various options available. In Table 7, we observe the text-outcome of the tool provided a set of opinions for mobile devices. The opinions are originally written in Greek. We have translated the outcome in English. In brackets the system outputs the sentiment strength of each word.

6 Conclusions

In this paper we presented a method for domain-specific opinion word discovery. It consists of a series of steps that complement each other in discovering



(a) GUI Welcome screen (b) Polarity classification options

(c) Stemmer Opinions process (d) Algorithm evaluation options

Fig. 3: The `NiosTo` Graphical User Interface

Table 7: Sample of extracted opinion words for various extraction steps from the gui and the mobiles domain source.

Positive Filterseed Opinion words and sentinent polarity weights					
good [1064]	protective [4]	well [191]	perfect [248]	functional [147]	fast [392]
exactly [28]	detailed [38]	excellent [75]	honest [8]	serious [3]	Finance [45]
incredible [92]	warm [1]	beautiful [96]	handy [91]	mild [57]	thrilled [7]
Negative Filterseed Opinion Words and sentiment polarity Weights					
bad [-63]	tragic [-5]	negative [-155]	difficulty [-4]	slow [-57]	annoying [-6]
inadequate [-3]	unhappy [-1]	expensive [-28]	Terrible [-8]	radiation [-18]	accurate [-12]
irrelevant [-2]	ridiculous [-2]	useless [-11]	unsavory [-1]	ugly [-8]	bad [-3]
Positive Conjunction extracted opinion words and sentiment weights					
responsive [2]	turns out [4]	hang [10]	practices [2]	inexpensive [1]	enough [7]
covers [1]	single [1]	great [1]	better for [1]	smooth [1]	aesthetically [1]
ease of use [1]	fast [1]	smooth [1]	Unfortunately [1]	rugged [4]	problems [1]
Negative Conjunction extracted opinion words and sentiment weights					
supersaturated [-1]	deprived [-1]	failure [-1]	appearance [-1]	packages [-1]	many [-1]
Double Propagation extracted opinion words and sentiment weights					
costing [6]	exceptional [3]	evaluation [-1]	suggests [9]	monster [3]	works [1]
heavy [21]	heavy [21]	forthcoming [1]	bigger [-1]	relaxed [4]	crazy [10]
authorities [3]	easily [1]	reluctantly [1]	important [1]	Bet [4]	demanding [7]

new words. We follow language patterns and opinion-words opinion-targets relationships to identify new words. Word polarity is calculated automatically by following a set of polarity disambiguation procedures. We evaluated the approach on a set of opinions about digital devices written in the Greek language. The experimental evaluation suggests that we can achieve satisfactory sentiment classification using this completely unsupervised approach. Finally, we presented a software tool, `NiosTo`, that implements the approach and enables the user to experiment with dictionary extraction and apply sentiment classification.

References

1. Giambattista Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. Automatic construction of an opinion-term vocabulary for ad hoc retrieval. In *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*, ECIR'08, pages 89–100, Berlin, Heidelberg, 2008. Springer-Verlag.
2. Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.
3. Tetsuya Nasukawa Hiroshi Kanayama. Fully automatic lexicon expansion for domain - oriented sentiment analysis. 2006.
4. M. Hu and B. Liu. Mining and summarizing customer reviews. 2004.
5. H. Kanayama and T. NasuKawa. "fully automatic lexicon expansion for domain - oriented sentiment analysis.".
6. Fotios Kokkoras, Konstantinos Ntonas, and Nick Bassiliades. Deixto: A web data extraction suite. In *Proceedings of the 6th Balkan Conference in Informatics*, BCI '13, pages 9–12, New York, NY, USA, 2013. ACM.
7. Kang Liu, Liheng Xu, Yang Liu, and Jun Zhao. Opinion target extraction using partially-supervised word alignment model. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 2134–2140. AAAI Press, 2013.
8. Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.*, 37(1):9–27, March 2011.
9. Wiebe J. Riloff, E. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, page 105112., 2003.
10. Vechtomova O. Skomorowski, J. Ad hoc retrieval of documents with topical opinion. *In:Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, Springer*, 4425:405417, 2007.
11. Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1533–1541, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
12. Philip S. Yu Xiaowen Ding, Bing Liu. A holistic lexicon-based approach to opinion mining. 2008.