The original publication is available at: Springer
DOI: 10.1007/978-3-642-24769-9_47

Determining the Polarity of Words through a Common Online Dictionary

António Paulo Santos[1], Carlos Ramos[1], and Nuno C. Marques[2]

[1] GECAD – Knowledge Engineering and Decision Support Group, Institute of Engineering - Polytechnic of Porto, Portugal
[2] Departamento de Informática, Faculdade de Ciências, Universidade Nova de Lisboa, Monte da Caparica, Portugal
pgsa (at) isep.ipp.pt, csr (at) dei.isep.ipp.pt, nmm (at) di.fct.unl.pt

Abstract. Considerable attention has been given to polarity of words and the creation of large polarity lexicons. Most of the approaches rely on advanced tools like part-of-speech taggers and rich lexical resources such as WordNet. In this paper we show and examine the viability to create a moderate-sized polarity lexicon using only a common online dictionary, five positive and five negative words, a set of highly accurate extraction rules, and a simple yet effective polarity propagation algorithm. The algorithm evaluation results show an accuracy of 84.86% for a lexicon of 3034 words.

Keywords: lexicon generation, polarity lexicon, polarity of words

1 Introduction

Entries on a polarity lexicon are tagged either as positive, negative, or neutral in some works. For instance, *good*, *beautiful*, *happiness* can be tagged as positive words, whereas words such as *bad*, *ugly*, *sadness* can be tagged as negative words. A polarity lexicon is a resource that can be used for instance to identify, classify and extract sentiment or opinions from sentences or larger units of text.

Most of the approaches to date on polarity lexicons generation have required advanced tools like part-of-speech taggers and rich lexical resources such as WordNet [16]. Most of the research has focused on English, as evidenced by the availability of lexical resources such as Harvard Inquirer [19], SentiWordNet [7], [2], Q-WordNet [1], WordNet-Affect [20].

In works such as [9], [14], [4], and [17] polarity is propagated based on WordNet [16] and graph propagation algorithms. Our approach is different because we use a common online dictionary. Esuli and Sebastiani [6] use WordNet [16] and Merriam-Webster online dictionary[1] to determine the polarity of terms through gloss classification (i.e. textual definitions classification). It is also worth mentioning the work by Rao and Ravichandran [17] which uses a common synonym dictionary (the OpenOffice thesaurus[2]) to build a graph. We focus on a common online dictionary to retrieve not only synonyms but also antonyms, the latter retrieved by a set of high accurate extraction rules (e.g. if *ugly* appears on dictionary as *opposite of beautiful*, we extract *beautiful* as an antonyms of *ugly*).

This paper presents first results of an empirical study (section 4) by using a common online dictionary to build a moderate-sized polarity lexicon with scarce resources. Beginning with a small seed set of words labeled as positive or negative and a common online dictionary we propagate the positive and negative sense to unlabeled words applying a simple and intuitive graph propagation algorithm (section 2.3). The study focuses on the polarity of Portuguese words but could be applied to other languages. The contribution of this work is to empirically show that with scarce resources and a simple yet effective semi-supervised approach it is possible to build a polarity lexicon.

2 Approach

In this section we present a graph propagation algorithm to propagate the positive and negative sense of a seed set of words labeled as positive or negative to unlabeled words. In this approach, a dictionary is viewed as a graph and the positive or negative sense of labeled words is propagated to their neighboring words.

[1] http://www.merriam-webster.com
[2] http://www.openoffice.org

2.1 Dictionary as a Graph

In a dictionary meanings are given through synonyms, antonyms, or richer semantic relations such as in WordNet [16]. We can think of a dictionary as a graph in which nodes correspond to words and edges correspond to synonyms, antonyms or other semantic relations between words. The representation of a dictionary as a directed graph is done for instance in [4], [10] and as an undirected graph for instance in [17], in both cases using the WordNet [16]. In this study we look to a common dictionary as a directed graph as shown on figure 1.

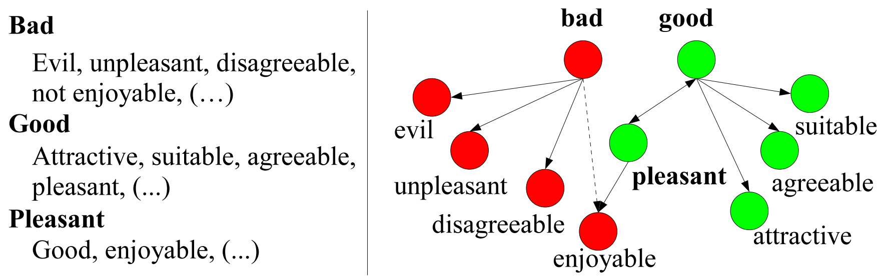

Fig. 1. Representation of the word *bad*, *good*, and *pleasant* as a directed graph (*dashed arrow = antonym, solid arrow = synonym*)

In figure 1, on the left side we have the meanings or definitions of the word *bad*, *good*, and *pleasant* and on the right side their representation as a directed graph.

2.2 Intuitive and Simple Polarity Propagation - Key Ideas

The first key idea is that viewing a dictionary as a directed graph and starting with the seed set of words manually classified as positive and negative we can propagate their polarity to unlabeled words applying a graph breadth-first traversal. E.g. on figure 1 assuming that good is a positive seed word we can propagate its positive sense to *attractive*, *suitable*, *agreeable*, and *pleasant*. We can then proceed the same way firstly for all seed words (e.g. *bad* and *pleasant*) and then for all other remaining words until we have reached all possible nodes. The positive and negative polarity is propagated based on the assumptions that:

1. Two synonymous or related words have the same or a close sense, so they should have the same polarity. E.g. if *good* is positive the synonymous or related words *attractive*, *suitable*, *agreeable*, and *pleasant* should also be.
2. Two antonyms words have opposite sense, so they should have opposite polarity. E.g. if the word *bad* is negative their antonym *enjoyable* should be positive.

The second key idea is that the closer a word is to a seed word, the higher the probability of the propagation being right (section 4.2, table 8). Knowing this, the described sense propagation approach can be extended assigning different weights to words according to their distance to the closest seed word. For instance, supposing that word A and Z (figure 2) are seed words. A is manually labeled as positive and Z as negative. The word B and Y are those which we want to know whether they are positive or negative (figure 2 at left).

Fig. 2. Polarity propagation

Following the simple unweighted approach we should: 1.) propagate the positive sense of seed word A to B which is the only neighbor of A; 2.) propagate the negative sense of Z to Y which is only neighbor of Z; and finally 3.) propagate the positive sense of B to Y. While in this approach, word Y should be considered both positive and negative because it has a positive and a negative neighbor (B and Z respectively). In the weighted propagation approach, Y should be considered more negative than positive (or even only negative), because their neighbor Z should be considered more important than B since Z is closer than B from a seed word.

2.3 Intuitive and Simple Polarity Propagation - Algorithm

In this section we present the algorithm applied to propagate the positive and negative sense of a seed set of words and at the same time to compute the short distance between each word and the closest seed word.

Assuming that we have a dictionary represented as a direct graph (figure 3 at left), we get the graph on the right (the polarity lexicon) by applying the following algorithm:

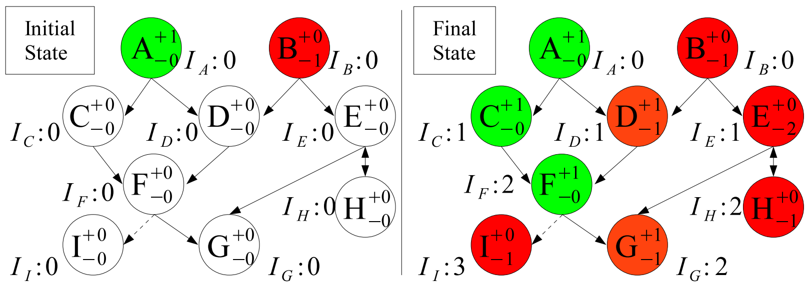

Fig. 3. Polarity propagation *(dashed arrow = antonym, solid arrow = synonym, Ix = Iteration or shortest distance from the closest seed word to word X)*

1. The first step is to label a set of words $W = \{w_1, \ldots, w_n\}$ as positive or negative, and for each word initialize an iteration counter I to 0. Finally add the words in a queue Q.
Taking as example the graph on the left on the figure 3, suppose that word A is labeled as positive by adding 1 unit to its positive counter and 0 to its negative counter (A^{+1}_{-0}). In a similar way word B is labeled as negative (B^{+0}_{-1}). The iteration counter of each word is initialized to 0 ($I_A : 0$ and $I_B : 0$), meaning they are seed words. Finally we add both words to a queue $Q = \{A^{+1}_{-0},\ B^{+1}_{-0}\}$.
2. Retrieve the first word w_1 from the queue Q and we get all their neighbors $Nb_{w1} = \{nb_1, \ldots, nb_m\}$. For each neighbor nb_i visited for the first time, set the iteration counter to *value of iteration counter of* w_1 *+ 1.*
Continuing the previous example, word A^{+1}_{-0} is retrieved from Q and we get all their neighbors $Nb_A = \{C^{+0}_{-0},\ D^{+0}_{-0}\}$. Since iteration counter of word A^{+1}_{-0} is set to 0 ($I_A : 0$) and we are visiting their neighbors C^{+0}_{-0} and D^{+0}_{-0} for the first time, the iteration counter of each one is set to $I_A +\ 1$ leaving us with $I_C : 1$ and $I_D : 1$. The value 1 means that the shortest distance from the closest seed word to word C^{+0}_{-0} and D^{+0}_{-0} is 1.
3. Propagate the positive or negative sense of the word w_1 to all their neighbors Nb_{w1} according to their semantic relation (e.g. synonymous or antonymous). The propagation is done increasing the positive (pos.) or negative (neg.) counter of each neighbor according to the following rules:

$$\begin{array}{llll} \text{If } w_1{>}0 \bigwedge w_1 \longrightarrow nb_j & \text{Then} & \text{pos.} \leftarrow \text{pos.} + & 1{\times}\text{Weight} \\ \text{If } w_1{<}0 \bigwedge w_1 \longrightarrow nb_j & \text{Then} & \text{neg.} \leftarrow \text{neg.} + & 1{\times}\text{Weight} \\ \text{If } w_1{>}0 \bigwedge w_1 \dashrightarrow nb_j & \text{Then} & \text{neg.} \leftarrow \text{neg.} + & 1{\times}\text{Weight} \\ \text{If } w_1{<}0 \bigwedge w_1 \dashrightarrow nb_j & \text{Then} & \text{pos.} \leftarrow \text{pos.} + & 1{\times}\text{Weight} \end{array}$$

The *Weight* variable should not be used or set to 1 if we want to apply an unweighted propagation approach. As discussed on section 2.2, the *Weight* should decrease as iteration increases.
Continuing the previous example, since word A^{+1}_{-0} is positive ($w_1{>}0$) and has two synonymous neighbors C^{+0}_{-0} and D^{+0}_{-0} ($w_1 \longrightarrow nb_j$), the first rule is applied to both (assuming Weight = 1*). This means that the positive counter of C^{+0}_{-0} and D^{+0}_{-0} is increased 1 unit leaving us with C^{+1}_{-0} and D^{+1}_{-0}.*

4. Mark word w_1, as visited, by adding him to a list of visited words V and for each neighbor nb_i of w_1:
   (a) If nb_i already exists on queue Q, update him (replace him);
   (b) If nb_i does not exists on queue Q nor list V, add him to the end of Q.
   Continuing the previous example, $V = \{A^{+1}_{-0}\}$, $Q = \{B^{+0}_{-1}, C^{+1}_{-0}, D^{+1}_{-0}\}$.
5. If the Q is not empty go to step 2 or else the algorithm ends.

Note that to prevent infinite loops, each node is visited just once following a breadth-1rst traversal. Note also that the polarity of a word can be propagate back for words that already have a polarity. For this reason, we may want to apply a weighted propagation approach where closer a word is from a seed word the greater it is their importance (greater is the positive or negative value to propagate to its neighbors). See step 5 on above algorithm to know how to apply a weighted propagation approach, and section 2.2, second paragraph a broader discussion. Note also that a word may spread its polarity for words from which it received their polarity (e.g, on figure 3, word H first recieves the negative polarity of E and then propagate this negative polarity back to H). This last case should not be a problem because on a real dictionary (graph) a word tend to have several neighbors, moreover the final polarity of a word is given not by the polarity of just one neighbor word, but by the balance of all their direct neighbors, for instance, table 5 shows that in our study the word *destruir* (to destroy) is strongly negative because have 22 negative and 5 positive direct neighbors.

At the end of the algorithm we get a polarity lexicon (Figure 3 at right) where each word has a positive, a negative and an iteration counter. We can use the positive and negative counters to compute not only the polarity but also their relative strength. For instance, word A^{+1}_{-0} is positive with strength 1 (*1/(1 + 0)*). Word E^{+0}_{-2} is negative with strength 1 (*2/(2 + 0)*). Word D^{+1}_{-1} is positive with strength 0.5 (*1/(1 + 1)*) and negative with same strength.

3 Polarity Lexicon - Pilot Experiment

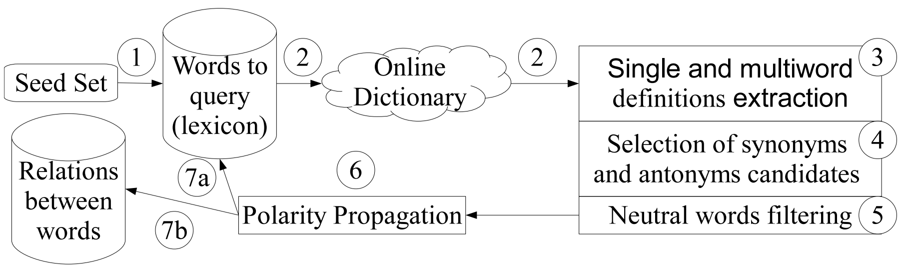

Fig. 4. Lexicon construction from an online dictionary

In section 2.3 we presented the algorithm to propagate the positive and negative sense of the seed words assuming that we already have a dictionary (lexicon) represented as a graph. In this section we present an approach to propagate the positive and negative sense of each seed word while building the graph at a same time. The short distance between the closest seed word and each word is also computed. The algorithm is:

Step 1. Proceed as described on section 2.3 step 1. In our empirical study we start with a seed set of 5 positive and 5 negative Portuguese words. We tried to choose pairs of opposite words and with a strong positive and negative sense, hoping to minimize the propagation errors on a later stage. We initialized the positive, negative and iteration counter of each seed word as described on 2.3 step 1, and we stored them in the queue “Words to Query”.

Step 2. Retrieve the first word not yet visited from the queue ”Words to Query”, and get all their list of definitions. In our study we query a public online dictionary for Portugese and we get a web page for each queried word containing their meaning. For instance, for the queried seed word *injusto* (unfair) we get the list of meanings illustrated by Figure 5.

Table 1. Queue "words to query" (*"+" = number of positive senses, "-" = number of negative senses, It. = iteration, V. = visited*).

Word	+	-	It.	V.	Word	+	-	It.	V.
Bom (good)	1	0	0	no	Mau (bad)	0	1	0	no
Positivo (positive)	1	0	0	no	Negativo (negative)	0	1	0	no
Certo (right)	1	0	0	no	Errado (wrong)	0	1	0	no
Alegria (joy)	1	0	0	no	Tristeza (sadness)	0	1	0	no
Justo (fair)	1	0	0	no	Injusto (unfair)	0	1	0	no

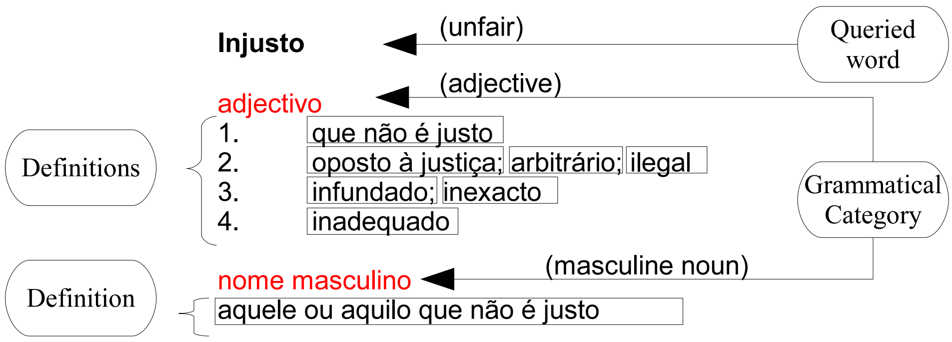

Fig. 5. Part of html page showing the definition (or definitions) of word *injusto* (unfair) *Definitions meaning: 1) opposite of justice; 2) that is not fair; arbitrary; illegal; 3) baseless; inaccurate; 4) inadequated; 5) who or what is not fair.*

Step 3. Extraction of definitions from the html document (web page). In Figure 5 the definitions to be extracted for the queried word *injusto* (unfair) are rounded by a square. We have 5 single word expressions and 3 multiword expressions.

Step 4 a. Get all single words either by taking all single words, e.g. *arbritário* (arbitrary), *ilegal* (illegal), or by extracting them from multi word expressions, e.g. *justiça* (justice) can be extracted from *oposto à justiça* (opposite of justice).

Step 4 b. Determination of the semantic relation (e.g. *synonym*, *antonym*) among the queried word and each extracted single word. Note that single words as opposed to multiword expressions can be used later to obtain new words from the online dictionary.

In our study, step 4 a) and 4 b) are performed by extraction and classification rules as those on table 2.

Table 2. Most frequent rules for single word extraction and semantic relation classification

Extraction and classification rule	Semantic Relation	E.g. based on Figure 5	Freq.
<single word>	Synonym	ilegal (illegal)	20605
fazer <single word> (*to do <single word>*)	Synonym	—	222
que tem <single word> (*which has <single word>*)	Synonym	—	112
acto de <single word> (*act of <single word>*)	Synonym	—	95
não <single word> (*not <single word>*)	Antonym	—	225
que não é <single word> (*that is not <single word>*)	Antonym	que não é <justo> (*that is not <fair>*)	35

Each extraction and classification rule extracts a <single word> and classifies it as synonym or antonym. For instance, for the word *injusto* (unfair) (Figure 5), the rule <*single word*>, among others, extracts *ilegal* (illegal) and classifies it as synonym. The rule *que não é <single word>* (that is not <single word>) extracts the single word *justo* (fair) and classify it as antonym.

The freq. column indicates the number of matches of a rule. Note that each rule needs to do an exact match to ensure that the extracted single word has meaning. A non exact match could increase the number of noisy single words, for instance, the extraction rule *que tem <single word>* (which has <single word>) applied to *que tem o poder de dicidir* (which has the power to decide) would extract the noisy word *o* (the).

Step 5. Filtration of neutral words (words which are neither positive nor negative). A word is not taken into consideration on next step if it is filtered.

In our study we used a small list of neutral words obtained by observing the dictionary. On example of figure 5, none of the words is filtered in this step.

Step 6. Propagation of the positive or negative polarity of queried word (qw) to all their directed neighbors according the rules presented on section 2.3, step 3. Determining the distance between the closest seed word to each single words sw_i as described on section 2.3, step 2.

Continuing the previous example, the negative polarity of *injusto* (unfair) is propagated according to the rules above, for instance to *ilegal* (illegal), *justo* (fair), and all others single words extracted on step 4 and not filtered on step 5. Since the word *justo* (fair) already exists on queue "words to query", its iteration counter remains unchanged (value 0). For all other single words the iteration counter is set to 1 (0 + 1, where 0 is the value of iteration counter of the word *injusto* (unfair)).

Step 7 a. Mark the queried word as visited and save all its meanings to be queried in future iterations.

Continuing the previous example, the word *injusto* (unfair) is marked as visited and its meanings saved on queue "Words to Query". The result of applying this step is shown in table 3.

Table 3. State of queue "Words to Query" after this step (*It. = Iteration, V. = Visited, ... = Other Seed Words)*

Word	+	-	It.	V.	Word	+	-	It.	V.
...	..	..	..	..	Arbitrário (arbitrary)	0	1	1	No
Injusto (unfair)	0	1	0	Yes	Ilegal (illegal)	0	1	1	No
Justo (fair)	2	0	0	No	...	..	..	..	..
Justiça (justice)	1	0	1	No					

Table 3 maintains a list of words with their positive and negative sense and it also maintains the short distance from the closest seed word to each word. This list forms our polarity lexicon.

Step 7 b. Save the queried word and its synonyms and antonyms. In our study we saved this information in the form of [*queried word*, *meaning word*, *semantic relation*].

Table 4. State of "relations between words" list after this step

Queried Word	Semantic Relation	Meaning Word
injusto (unfair)	Synonym	ilegal (illegal)
injusto (unfair)	Antonym	justo (fair)
...	...	...

Table 4 forms an adjacency list (a data structure for representing the dictionary as a graph).

Step 8: Finally, we check if there are more words to query. If there are, we retrieve the first word not yet visited from the queue "words to query" and return to step 2, if not the process ends. In the end, after applying the algorithm, the output is a polarity lexicon (table 5) and an adjacency list (table 4).

Table 5. Sample output of polarity words (*+% = positive strength, -% = negative strength*)

Word	**PoS**	+	-	**It.**	+%	-%
força (strength)	noun	39	1	4	97.5	2.5
conveniente (convenient)	adjective	17	0	2	100	0
inconstante (fickle)	adjective	0	18	2	0	100
perturbar (to disturb)	verb	4	21	4	16	84
destruir (to destroy)	verb	6	22	4	21.43	78.57
confundir (to confuse)	verb	0	15	4	0	100

4 Evaluation

To evaluate the ability of the propagation algorithm to classify a word as positive or negative (algorithm presented on section 3), we first asked two humans to classify a sample of 524 words and to measure the agreement between them (section 4.1). Afterwards, we measured the agreement between them and the algorithm (4.2).

4.1 Inter-Human Agreement

The following steps here performed for measure the inter-human agreement:

1. We randomly selected 524 words from the generated lexicon of 9107 words.
2. Initially we asked two native speakers of Portuguese to do a domain and context independent classification of each word to one of four categories: *positive* (+), *negative* (-), *neutral* (0), and *ambiguous* (A) for words positive and negative at same time. Soon there was the need to create the *unknown* category to words which the annotators do not know its meaning.

Table 6. Contingency table for inter-human classification agreement (diagonal elements show the agreement between Humans, off-diagonal elements show the disagreement)

		H1				**Total**
		+	-	**0**	**A**	
H2	+	188	5	1	4	198
	-	13	157	2	12	184
	0	75	13	6	22	116
	A	6	4	**0**	1	11
Total		282	179	9	39	509

		H1		**Total**
		+	-	
H2	+	270	18	288
	-	15	157	172
Total		285	175	460

Of the 524 words, 15 were *unknown* to one or both annotators, therefore they were immediately discarded. For the remaining 509 words belonging to the remaining four categories, the inter-human agreement was 69.16% (352 words) as shown on table 6 on the left (sum of diagonal elements). This is a similar agreement reported by Kim and Hovy [14] for English, and similar to Jijkoun and Hofmann [10] for Dutch, in both cases considering the categories: positive, negative and neutral. Kim and Hovy [14] reported an agreement of 76.19% for adjectives and 62.35% for verbs. Jijkoun and Hofmann [10] reported an agreement of 69%.

For the purpose of this study, we were only interested in evaluating the positive and negative sense assigned by the algorithm. Therefore, we discarded all words marked as ambiguous (like [10]) by one or both humans, and as [14] we merged the positive and neutral categories. In this case the inter-human agreement for the positive and negative categories was 92.83% (*Cohen's* $k = 0.85$) (table 6 at right). In the same way as [14] and as theoretically expected, the merging of positive and neutral categories increased the inter-human agreement. In our study increased from 69.16% to 92.83%.

Since 49 ambiguous words were removed, we were left with 460 words classified as positive or negative for human-machine agreement evaluation.

4.2 Human-Machine Agreement

In this section we report the agreement between each human (Human1 and Human2) and the algorithm presented on section 3. The results are shown on table 7. On run *H1:M.10seeds* and *H2:M.10seeds* it was used the 10 seed words pointed out on section 3, table 1. Table 7 shows that the agreement between Human1 and Machine is 75.43%, and between Human2 and Machine is 74.78%.

Table 7. Lexicon evaluation results *(A = Accuracy or Agreement, P = Precision, R = Recall and F1 = F-Measure, for positive and negative classes).*

		Positive Class			Negative Class		
Run	**A (%)**	**P (%)**	**R (%)**	**F1 (%)**	**P (%)**	**R (%)**	**F1 (%)**
H1:H2	92.83	n/a	n/a	n/a	n/a	n/a	n/a
H1:M.10seeds	75.43	75.22	90.63	82.20	76.11	50.00	60.35
H2:M.10seeds	74.78	74.35	90.63	81.65	76.11	49.14	59.72

Table 8 shows the accuracy of run *H1:M.10seeds* and *H2:M.10seeds* by iteration. As it is shown, the accuracy decreases at each iteration. Given these results, we concluded that we should stop to propagate the positive and negative sense at earlier iterations.

Table 8. Human-Machine agreement by iteration for the 460 words *(It. = Iteration, T. = Total).*

	Human1:M.10seeds		
It.	**Agreement**	**Disagreement**	**T.**
0	10 (100.00%)	0 (0.00%)	10
1	52 (92.86%)	4 (7.14%)	56
2	48 (82.76%)	10 (17.24%)	58
3	54 (81.82%)	12 (18.18%)	66
4	77 (81.91%)	17 (18.09%)	94
5	54 (60.67%)	35 (39.33%)	89
6	52 (59.77%)	35 (40.23%)	87
T.	347 (75.43%)	113 (24.57%)	460

	Human2:M.10seeds		
It.	**Agreement**	**Disagreement**	**T.**
0	10 (100.00%)	0 (0.00%)	10
1	51 (91.07%)	5 (8.93%)	56
2	50 (86.21%)	8 (13.79%)	58
3	54 (81.82%)	12 (18.18%)	66
4	70 (74.47%)	24 (25.53%)	94
5	56 (62.92%)	33 (37.08%)	89
6	53 (60.92%)	34 (39.08%)	87
T.	344 (74.78%)	116 (25.22%)	460

Considering all the 460 words and all 6 iterations, the total agreement between Human1 and the Machine was 75.43% and between Human2 and Machine was 74.78% for a lexicon of 9017 words. However, if we stop at iteration 4 we get an accuracy of 84.86% (241/284) for Human1 and 82.75% (235/284) for Human2 (stopping at iteration 4 reduces the lexicon from 9017 to 3034 words). These accuracies are similar to those obtained by [6] using the Merriam-Webster online dictionary and tested on three different test sets. These last authors have obtained accuracies of 83.71% (for 1,336 positive and negative adjectives), 79.78% (for 3,596 positive and negative terms), and 85.44% (for 663 adjectives).

5 Related Work

Several are the techniques for automatically building general wordnets. For example, Barbu et al [3] which used similar techniques for building a general wordnet for Hungarian. However we further study the polarity of words. In this section we will restrict our discussion to direct related work (polarity lexicons techniques).

There are a number of previous works which focus on building polarity lexicons. There are basically two main approaches to build it:

1. Approaches based on dictionaries. These approaches explore synonyms, antonyms, hypernyms, and hyponyms, among other relations, e.g. [12], [14], [17], or explore glosses (i.e. textual definitions) classification [6], [5]. Most of these approaches are based on WordNet [16]. Our work

differs in that we use a common online dictionary. There are also approaches that derive a polarity lexicon for a new language using an existing lexicon on another language and bilingual dictionaries [13] [23].

2. Approaches based on corpus. These approaches explore the co-occurrence of words, e.g. [8], [22], [21], [11].

5.1 Approaches Based on Dictionaries

Kamps et al. [12] determine the positive or negative semantic orientation of adjectives. They rely on the structure of the WordNet [16] to build a graph on the adjectives based on the WordNet *synonymous* relation. To determine the orientation of an adjective they measure the minimum distance between that adjective and two seed opposite adjectives *good* and *bad*. With this approach, only adjectives connected to any of the two chosen seed adjectives by some path in the synonymy relation graph can be evaluated.

Kim and Hovy [14] determine the strength of the positive and negative orientation of words. They label a small amount of seed words by hand and then use the *synonymous* and *antonymous* relation of WordNet [16] to expand them.

Esuli et al. [6], [5] determine the orientation of subjective terms. The orientation of a term is obtained based on the classification of glosses (i.e. textual definitions) that terms have in an online or offline glossary or dictionary. The method starts with a seed set of positive and negative words. That seed set is then expanded with WordNet [16] exploring the *synonym*, *direct antonym*, *indirect antonym*, *hypernym*, and *hyponym* relations. Then the glosses are extracted from WordNet [16] and from the online version of the Merriam-Webster[3] dictionary, and represented as vectors. Finally, it is applied a binary classifiers (naïve Bayes and Support Vector Machines).

Rao and Ravichandran [17] determine the positive and negative polarity of words. The polarity detection is treated as a semi-supervised label propagation problem in a graph. They try several graph-based semi-supervised learning methods like Mincuts, Randomized Mincuts, and Label Propagation. The study is done using WordNet [16] and OpenOffice thesaurus.

Silva et al. [18] determine the positive, negative and neutral polarity of human adjectives (adjectives that co-occur with a human subject). The authors build a graph from a seed set of adjectives manually classified and adjective synonyms from the union of multiple open thesauri. They compute the distance in the graph of each adjective whose polarity is unknown to the seed set adjectives already classified. The computed distances are then used to generate input features for an automatic polarity classifier.

5.2 Approaches Based on Corpus

Hatzivassiloglou and McKeown [8] determine the positive or negative semantic orientation of adjectives. They rely on the idea that conjunctions (e.g. *and*, *or*, *but*, *either-or*, and *neither-nor*) between adjectives provide indirect information about orientation. For instance, while *fair and legitimate* and *corrupt and brutal* have the same orientation and occurs in their corpus, the pairs *fair and brutal* and *corrupt and legitimate* would be semantically anomalous. These last two pairs of adjectives are reversed for *but*, which usually connects two adjectives of different orientations.

Turney and Littman [22] determine the positive or negative semantic orientation of two word phrases containing adjectives or adverbs. They rely on the idea that a phrase has a positive orientation when it has good associations and negative semantic orientations when it has bad associations. For determining these good and bad associations for each phrase, the authors query the Altavista[4] search engine using the phrase, the operator NEAR, and the opposite words *excellent* and *poor* (e.g. "phrase$_n$ NEAR excellent", "phrase$_n$ NEAR poor"). Based on the number of documents returned for each phrase, it is used the PMI-IR (Pointwise Mutual Information and Information Retrieval) to calculate the semantic orientation (for more details about PMI-IR, please refer to the original paper).

Takamura et al. [21] determine the positive or negative semantic orientation of words. They rely on a method that they call "spin model". The authors construct a lexical network by connecting two words and if one word appears in the gloss of the other word, they apply the spin model. The intuition behind this is that if a word has a semantic orientation, then the words in its gloss tend to have the same polarity.

[3] http://www.merriam-webster.com/

[4] http://www.altavista.com/

6 Conclusion and Future Work

This paper shows how to build a moderate-sized polarity lexicon using only a common online dictionary; a small seed set of words, a set of high accurate extraction rules, and a simple yet effective graph polarity propagation algorithm. We also show how to infer the antonymy relation using extraction rules. The acquired results are similar to those of related work, but we use less resources and a more direct method. Evidence is given regarding the interest of using more specific, but highly accurate extraction rules of single words. The applied method is language-independent and can easily be applied to other languages for which a common online or offline dictionary exists. The method tries to capture the sense that a word has in most contexts (e.g. usually *good, agreeable* is positive, and *bad, unpleasant* is negative).

Since this pilot experiment was encouraging, we plan, for instance, to: a) increase the polarity lexicon; b) study a way to extend the positive and negative lexicon to include neutral words; c) adapt this domain independent lexicon into a domain-specific lexicon; d) use the polarity lexicon to identify polarity senses of larger units of text such as sentences; e) capture the different meanings of a word, taking into account the context of the sentence using models similar to those used in part-of-speech tagging [15].

Once our approach is applicable to English, as future work we may compare this approach, for instance, with SentiWordNet [7], [2], Q-WordNet [1].

Acknowledgments. This work was funded by the *FCT - Fundação para a Ciência e Tecnologia* (Portuguese research funding agency) through fellowship SFRH/BD/47551/2008.

References

1. Agerri, R., García-Serrano, A. Q-WordNet: Extracting polarity from WordNet senses. Seventh Conference on International Language Resources and Evaluation, Malta. Retrieved May. (2010).
2. Baccianella, S. et al. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. Proceedings of the 7th Conference on Language Resources and Evaluation (LREC'10). pp. 2200-2204 , Valletta, MT (2010).
3. Barbu, E., Mititelu, V.B. Automatic building of Wordnets. In: N. Nicolov, K. Bontcheva, G. Angelova, R.M. (Eds) (ed.) Recent Advances in Natural Language Processing IV (RANLP-05). pp. 217-226 Amsterdam; Philadelphia; J. Benjamins Pub. Co; 1999 (2005).
4. Blair-Goldensohn, S. et al. Building a Sentiment Summarizer for Local Service Reviews. Electrical Engineering. (2008).
5. Esuli, A., Sebastiani, F. Determining term subjectivity and term orientation for opinion mining. Proceedings the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-2006). p. 193–200 (2006).
6. Esuli, A., Sebastiani, F. Determining the semantic orientation of terms through gloss classification. Proceedings of the 14th ACM international conference on Information and knowledge management. p. 617–624 ACM, Bremen, Germany (2005).
7. Esuli, A., Sebastiani, F. SentiWordNet: A publicly available lexical resource for opinion mining. In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06). p. 417–422 Citeseer, Genova, IT (2006).
8. Hatzivassiloglou, V., McKeown, K.R. Predicting the semantic orientation of adjectives. 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. p. 174–181 Association for Computational Linguistics (1997).
9. Hu, M., Liu, B. Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. p. 168–177 ACM New York, NY, USA (2004).
10. Jijkoun, V., Hofmann, K. Generating a Non-English Subjectivity Lexicon : Relations That Matter. Computational Linguistics. April, 398-405 (2009).
11. Kaji, N., Kitsuregawa, M. Building lexicon for sentiment analysis from massive collection of HTML documents. Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). p. 1075–1083 (2007).
12. Kamps, J. et al. Using WordNet to measure semantic orientation of adjectives. Proceedings of LREC. p. 1115–1118 (2004).
13. Kim, J. et al. Conveying Subjectivity of a Lexicon of One Language into Another Using a Bilingual Dictionary and a Link Analysis Algorithm. International Journal of Computer Processing Of Languages. 22, 02 & 03, 205 (2009).

14. Kim, S.M., Hovy, E. Determining the sentiment of opinions. Proceedings of the 20th international conference on Computational Linguistics. p. 1367 Association for Computational Linguistics (2004).
15. Marques, N.C., Lopes, G.P. Tagging With Small Training Corpora. In Proceedings of the International Conference on Intelligent Data Analysis (IDA'01). pp. 63-72 Lecture Notes in Computer Science 2189, Cascais, Portugal. Springer Verlag. (2001).
16. Miller, G. WordNet: A lexical database for English. Communications of the ACM. 38, 11, 39-41 (1995).
17. Rao, D., Ravichandran, D. Semi-supervised polarity lexicon induction. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on EACL 09. April, 675-682 (2009).
18. Silva, M.J. et al. Automatic Expansion of a Social Judgment Lexicon for Sentiment Analysis. Technical Report. TR 10-08. University of Lisbon, Faculty of Sciences, LASIGE. (2010).
19. Stone, P.J. The General Inquirer: A Computer Approach to Content Analysis. M.I.T. Press; First Edition edition (January 1, 1966) (1966).
20. Strapparava, C., Valitutti, A. WordNet-Affect: an affective extension of WordNet. Proceedings of LREC. p. 1083–1086 Citeseer (2004).
21. Takamura, H. et al. Extracting Emotional Polarity of Words using Spin Model. Proceedings of the Joint Workshop of Vietnamese Society of AI, SIGKBS-JSAI, ICS-IPSJ and IEICE-SIGAI on Active Mining (AM2004). , Hanoi, Vietnam (2004).
22. Turney, P.D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. p. 417–424 Association for Computational Linguistics, Morristown, NJ, USA (2002).
23. Waltinger, U. German Polarity Clues: A Lexical Resource for German Sentiment Analysis. Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC). p. 1638–1642 (2010).