A Lexicon-based Approach for Sentiment Classification of Amazon Books Reviews in Italian Language

Franco Chiavetta[1,2], Giosuè Lo Bosco[2,3] and Giovanni Pilato[4]

[1]*Istituto Tecnico Settore Tecnologico Vittorio Emanuele III, Via Duca della Verdura 48, 90143, Palermo, Italy*

[2]*Department of Mathematics and Computer Science, University of Palermo, Via Archirafi 34, 90123, Palermo, Italy*

[3]*Department of Key Sciences and Enabling Technologies, Euro-Mediterranean Institute of Science and Technology, Via Michele Miraglia 20, 90139, Palermo, Italy*

[4]*ICAR-CNR, Viale delle Scienze - Edificio 11, 90128, Palermo, Italy*

francochiavetta@gmail.com, giosue.lobosco@unipa.it, giovanni.pilato@cnr.it

Keywords: Sentiment Analysis, Opinion Mining.

Abstract: We present a system aimed at the automatic classification of the sentiment orientation expressed into book reviews written in Italian language. The system we have developed is found on a lexicon-based approach and uses NLP techniques in order to take into account the linguistic relation between terms in the analyzed texts. The classification of a review is based on the average sentiment strenght of its sentences, while the classification of each sentence is obtained through a parsing process inspecting, for each term, a window of previous items to detect particular combinations of elements giving inversions or variations of polarity. The score of a single word depends on all the associated meanings considering also semantically related concepts as synonyms and hyperonims. Concepts associated to words are extracted from a proper stratification of linguistic resources that we adopt to solve the problems of lack of an opinion lexicon specifically tailored on the Italian language. The system has been prototyped by using Python language and it has been tested on a dataset of reviews crawled from Amazon.it, the Italian Amazon website. Experiments show that the proposed system is able to automatically classify both positive and negative reviews, with an average accuracy of above 82%.

1 INTRODUCTION

Sentiment analysis has the objective of extracting from the web the current opinion toward someone or something through the massive classification of texts generated by users (Liu, 2012). The development of automatic tools for Sentiment Analysis is required by the huge and growing amount of opinionated user generated contents currently available on the Web. A component of a sentiment analysis platform is the ”sentiment classifier”, a software having the role to decide the ”sentiment polarity” of a text, i.e. if it expresses a positive or a negative opinion toward the target. An extended survey on the sentiment classification approaches is given in (Hassan et al., 2014). They are usually divided into “Machine Learning” and “Lexicon-based” approaches. Among the former ones we recall the use of Naïve Bayes classifiers (Duda and Hart, 1973): see for example (Domingos and Pazzani, 1997), (Lewis, 1998), (Dinu and Iuga, 2012), (Garcia and Gamallo, 2014) and (Shanmuganathan and Sakthivel, 2015); besides, Support Vectors Machines (SVM) (Boser et al., 1992), (Meyer et al., 2003) have been successfully employed by Pang and Lee in(Pang et al., 2002), Dave et al in (Dave et al., 2003) and in (Kennedy and Inkpen, 2006).

Lexicon based approaches (Taboada et al., 2011) use as knowledge base lexical resources named *opinion lexicon* (Hu and Liu, 2004), that associate words to their sentiment orientation represented for example by positive and negative“scores”. Their use in sentiment analysis research starts from the assumption that single words can be considered as a unit of opinion information, and therefore it can provide indications to detect document sentiment and subjectivity. The annotation can be done either manually or by automatic, semi-supervised, processes that, using linguistic resources like a corpora (Littman and M.L., 2002), a thesaurus, or a more sophisticated one like Wordnet (Fellbaum, 1998), (Dragut et al., 2010), generate the lexicon. The most popular opinion lexicon de-

Chiavetta, F., Bosco, G. and Pilato, G.
A Lexicon-based Approach for Sentiment Classification of Amazon Books Reviews in Italian Language.
In *Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016) - Volume 2*, pages 159-170
ISBN: 978-989-758-186-1

rived from Wordnet (Fellbaum, 1998) and containing posterior polarities (i.e. polarities associated to each *word sense*) is SentiWordNet (Esuli et al., 2010). In other cases the lexicon associates to each word a *prior polarity*, i.e. the polarity for its non-disambiguated meaning, out of any context, as human beings perceive by using cognitive knowledge. An example of opinion lexicon based on prior polarities is represented by MPQA (Wilson et al., 2005). In general, the effectiveness of the approach is highly dependent both on the correctness of the preprocessing steps (e.g the right "part-of-speech" detection) and on consistency and quality of the opinion lexicon and the number of terms it contains (*coverage*).

A problem in the research activities on Sentiment Analysis is that available literature is mostly focused on documents written in the English language. Moreover, most of the available resources needed in this approach to sentiment classification, like opinion lexicons, manually labelled corpora and NLP tools, are available for the English language only. The lack of linguistic resources is considered as a critical issue in most of the research works regarding Sentiment Analysis of non English languages.

Examples of sentiment analysis methods applied to non English languages texts are given in (Kim and Hovy, 2006) for the German language, in (Kanayama and Nasukawa, 2006) and (Takamura et al., 2006) for the Japanese language, in (Yi et al., 2005) and (Zagibalov and Carroll, 2008) for the Chinese language, in (Abbasi et al., 2008) for the Arabic language. In (Casoto et al., 2008) an example of sentiment analysis applied to movie reviews written in Italian language is given.

Several methods have been investigated to automatically generate resources for a new language starting from lexical resources already available for the English language. In (Mihalcea et al., 2007) authors illustrate a method that, given a cross-lingual bridge between English and the selected target language, such as a bilingual dictionary or a parallel corpus, rapidly create tools for sentiment analysis in the new language (their work is on Romanian language).

To work around the lack of resources and tools for other languages, another common approach is to make a full translation of the text during a preprocessing by "state of the art" automatic translators, and then apply all traditional stages of the sentiment analysis on the corresponding English text (see for example (Bautin et al., 2008)) where obtained results show a certain consistency of Sentiment Analysis applied to automatically translated texts across several languages. Such solutions, however, present several problems including imprecision in translation and disambiguation of words.

There are several types of user generated textual contents on which sentiment analysis can be applied. Differences can consist in their usual average size, application domain, context, technical terms presence, linguistic level of author, number opinion holders, and much more.

This paper is focused on the analysis of books reviews, i.e. texts having in general a well defined target (the book), few technical terms, except for references to authors and book titles. The linguist level, e.g. the syntactical and grammatical correctness and the richness of vocabulary used, is usually higher with respect to reviews regarding other products. This can be in general ascribed to the different cultural profile of the reviewers: for example, the manner of expression of a teenager in reviewing a game may be very different from the way an adult with a good cultural level can comment the latest book by an affirmed Italian book writer. Moreover, books' reviewers may use domain related terms.

From the size point of view, a book review is usually shorter than a full blog's article and longer than a microblogging post (e.g. a "tweet"). While the short number of characters available to write a tweet doesn't allow, from the linguistic point of view, complicated and articulated sentences and forces the writer to explicitly use common opinion-bearing words easily revealing its sentiment, in a review the more common opinion-bearing adjectives are replaced by more complicated expressions conveying user's sentiment including irony, which cannot be easily identified and used in document representation. In designing a sentiment classification algorithm all these differences must be considered to choose the appropriate strategy. For example, if sentiment classification of tweets can be successfully performed by a probabilistic Naïve Bayes classification scheme, where the shortness of the text and the use of opinion-bearing words allow to neglect its grammatical structure, the order and relative position of words, in favor of the multiplicity of occurrences (according to a "bag of word" model) (Shanmuganathan and Sakthivel, 2015), this strategy can not properly works on longer and more sophisticated texts. For these reasons we have developed a sentiment classification scheme taking into account the structure of the sentences to be analyzed, using a natural language processing approach based also on linguistic resources.

This paper proposes a method for Sentiment Analysis applied to documents written in Italian language. In particular, we have developed and evaluated a linguistic algorithm aimed at classifying books reviews by using a lexicon-based approach.

The system takes into account both the context of words, and the presence of "valence shifters" (Polanyi and Zaenen, 2006), that can change the polarity of a given word. Furthermore, the methodology tries to consider the *concepts* represented by a word in its context more than the simple presence of a "word form", i.e. simple strings of letters.

In order to support the experimental activities a complete interactive framework has been designed and implemented; it provides a toolbox for document analysis, classifier refinement and evaluation. The framework has been used to evaluate the proposed methodology for opinion polarity analysis on both domain dependent and independent environments. The results give an accuracy of 85.5% for positive reviews and 84.7% for negative reviews of the proposed approach with a significant improvement with respect to 78.1% for positive reviews and 49.6% for negative reviews obtained by using a baseline classifier based on a lexicon of Italian words.

Since the most popular website collecting books reviews is Amazon, we have applied our classification approach to a dataset extracted from the Italian Amazon site (http://www.amazon.it). The dataset has been created by means of a properly developed grabber that is a part of the realized framework.

2 OVERVIEW OF THE APPROACH

The overall architecture of the developed system is shown in Figure 1. The core element is a classifier tuned to perform a sentiment classification task on books' reviews written in Italian language. Our approach classifies a document on the basis of the average sentiment strengths of its sentences. The sentiment expressed in the sentences is obtained through the use of tools suited for the Italian language, in all the steps preceding a "scoring phase", i.e. the part of the process that attributes scores to each word in any given sentence of the text. In Section 2.1, we describe the approach to obtain scores for Italian words. To this purpose, we will make use of a "concept-based" scoring technique by using Wordnet related lexical resources. In Section 2.2 we illustrate the classification algorithm. Experiments and result are illustrated in the Section 3.

2.1 Resources for Concept-based Scoring

As mentioned above, the developed approach obtains the sentiment polarity expressed in a review by calculating the average sentiment strengths of its sentences. We state that the efficacy of this approach strongly depends on the possibility of computing the polarity scores for the largest number of Italian terms as possible. To reach this goal we have investigated a strategy we named *concept-based scoring*, where the key idea is that human beings associate sentiment to "concepts" and not to words, i.e. strings depending from the language used[1].

Moreover, since human beings associate sentiments to words by mediating between different meanings, we have chosen to exploit proper linguistic resources to retrieve meanings. In particular we make use of semantic relations like "synonymy" or the "IS-A" relationship or similar correspondences to better calculate a sentiment score for a given word.

These capabilities are available in resources like Wordnet, in which the linguistic knowledge is represented "by concepts" by means of two parallel components:

- a *lexical* component, collecting words (understood as character strings separated from their meaning) by organizing them into syntactic categories (nouns, verbs, adjectives, adverbs).
- a *semantic* component, clustering words into "synsets", i.e. list of synonyms expressing the same concept and representing also other semantic relationships between concepts (hyperonymy, hyponymy, antinomy, meronymy, etc.);

This set of relations between synsets allow us to consider WordNet as a lightweight ontology (Liu and Özsu, 2009) where lexically or conceptually related synsets are connected: for example, nouns can be linked through hyperonymy/hyponymy and meronymy/holonymy relations which can also be inherited determining a hierarchy. Verbs are organized via troponym, hypernym and entailment relations. Adjectives are linked to their antonyms, and relational adjectives point to their related nouns. Since, at the best of our knowledge, there are not freely available and well tested high coverage opinion lexicon containing posterior polarities of italian words, we have investigated the possibility to build a "layered" opinion lexicon by coupling WordNet-like existing resources in a sort of "stack".

At the basis of this stack we searched for a "multilingual WordNet", i.e. a linguistic resources containing at least a database for the Italian language and a database for the English language, where the

[1] *'What's in a name? that which we call a rose by any other name would smell as sweet...'*, W. Shakespeare, *Romeo and Juliet*

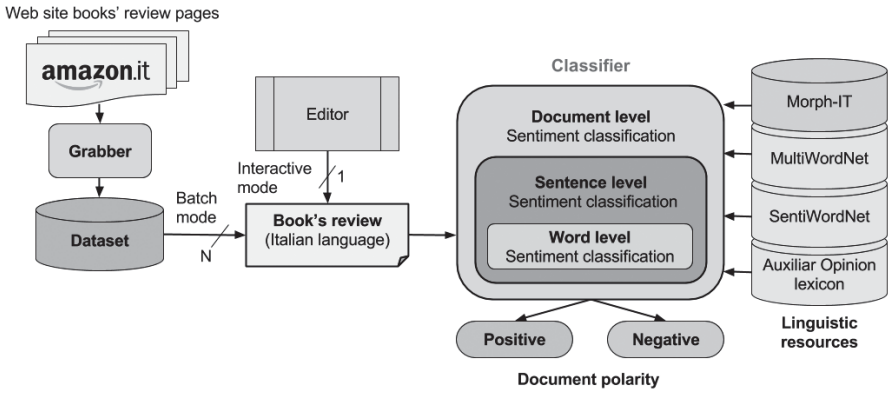

Figure 1: Architecture of the system.

databases are both structured on the basis of the original Princeton WordNet (PWN) (Fellbaum, 1998) and, moreover, they are also aligned to support multilingualism, and they are sufficiently extended to assure high probability to find Italian terms and the corresponding English translations. There are at least two models for building a multilingual WordNet in literature. The first one, named "merge model", has been adopted within the EuroWordNet project (EWN) (Vossen, 1998) and consists in building language specific wordnets independently from each other, trying to find their correspondences in a successive phase (Vossen, 1998).

The second model, "the expand model" adopted within the MultiWordNet project (Bentivogli et al., 2002), consists in building language specific Wordnets starting as much as possible from the synsets (concepts) and the semantic relations available in the PWN. The Wordnet for a foreign language is built by adding synsets in correspondence with the PWN synsets, whenever possible, and importing semantic relations from the corresponding English synsets; i.e. we assume that, if there are two synsets in PWN and a relation holding between them, the same relation holds between the corresponding synsets in the other language. This building strategy makes sense only if there exist few structural differences between English and the lexicon of the other language, i.e. there are relatively few cases when the synset of one language has no correspondent in the other language. A situation like this is called "*lexical gap*". A recent work on the lexical gap existing between English and Italian languages is given in (Bentivogli et al., 2002).

MultiWordnet contains an Italian Wordnet strongly aligned to the English PWN with a percentage of lexical gap between Italian and English synsets around 1 % only. These characteristics have determined the choice of MultiWordNet as a component of our stack to realize a concept-based scoring. Figure 2 shows in the middle the three dimensional structure of the so called "lexical matrix" of MultiWordnet: in the figure, words in a language are indicated by W_j; meanings are indicated by M_i; languages are indicated by L_k. Moreover, the main lexical and semantic relations are also shown. The E^I_{ij} represents intersections.

For a more detailed description of the Multiwordnet scheme we remand to the paper (Bentivogli et al., 2002) and reference therein. Concerning the second component of the stack, we have analyzed and compared several opinion lexicons (Agerri and Garcia-Serrano, 2010),(Cambria et al., 2014),(Strapparava and Valitutti, 2004), (Compagnoni et al., 2007). The one we consider is SentiWordnet, the best to couple with MultiWordnet (Esuli et al., 2010), an opinion lexicon obtained from the annotation of all 117659 synsets of the English PWN, representing hence a very high coverage opinion lexicon. Its elements are named *senti_synsets* because each one is a synset associated with a triple (P,N,O) of scores, i.e. a positive, a negative and a objective polarity scores having values in [0.0, 1.0] and sum equal to one. Finally, the structure of the stack is represented in figure 2: being based on MultiWordnet as first component, and on SentiWordNet as second component, we get a minimal lexical gap and a maximal coverage because their Wordnet-Like structures are both aligned whenever possible with Princeton WordNet English synsets, assuring the possibility to realize a concept-based scoring.

As illustrated in figures 1 and 2 our system includes also a third component we named "Auxiliar Opinion Lexicon", used to take into account misclassifications, missing terms, domain specific terms.

2.1.1 The Resource for the Morphological Normalization/Lemmatization

Most of the sentiment lexicons described in literature

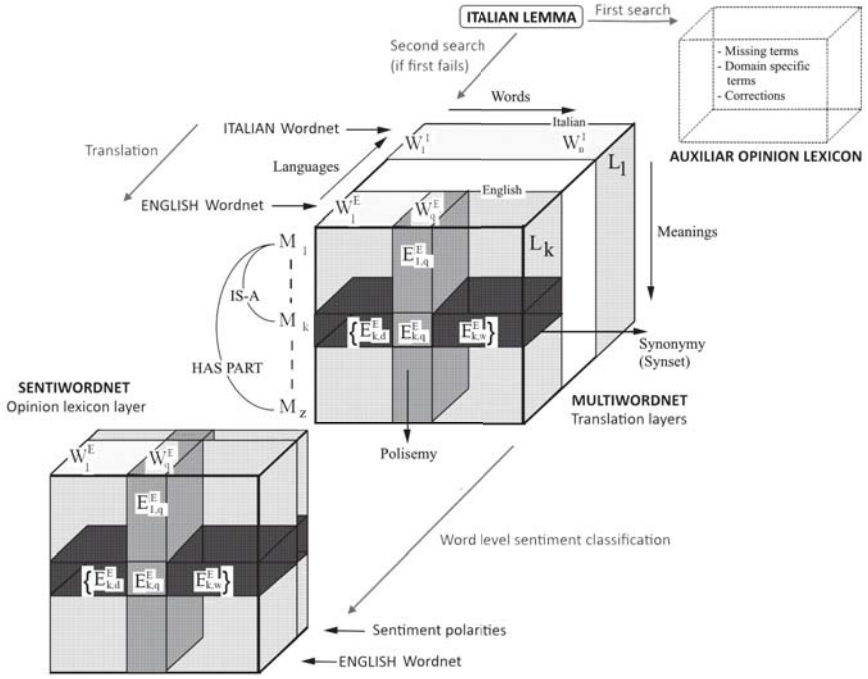

Figure 2: The stack of lexical resources used as opinion lexicon.

contain lists of tagged lemmas, i.e. the canonical form (or dictionary form) of a word. For instance, the latest version of MultiWordNet (1.39) contains around 58,000 Italian word senses and 41,500 lemmas organized into 32,700 synsets aligned whenever possible with Princeton WordNet English synsets. When using such kind of resources in sentiment analysis it is necessary to operate a previous step of sense disambiguation in order to identify the corresponding lemma of a word. In ontologies like Wordnet and MultiWordent the synonyms contained within a synset are called *lemmas*. The retrieval of synonyms associated to a word requires therefore a previous morphological transformation named "lemmatization" that, given a word, returns its inflected form. Before searching the meaning of a word in the Italian Wordnet we thus perform a lemmatization by using "Morph-it!" a morphological resource for the Italian language (Zanchetta and Baroni, 2005) . "Morph-it!" is a lexicon of inflected forms with their lemma and morphological features.

These resource currently contains 505,074 entries and 35,056 lemmas and therefore it can be used as a data source for a lemmatizer / morphological analyzer / morphological generator.

2.2 Sentiment Classification of a Document

Here we describe the approach used to perform the sentiment classification of a document (a review), The classifier, i.e. the main function *DocumentLevelSC(D,τ)* described in the Algorithm 1 receives as parameters the document D, and a threshold $\tau \in \mathbb{R}$ used to decide the positive or negative overall polarity as described below.

The classifier starts with a preprocessing of the review by using the *TextCleaner(D)* function. The purpose of this function is to obtain a text with fewer ambiguities and errors: it converts all the letters to lowercase, unescapes "htmlentities", deletes some escape sequences as "\n", "\r", "\t", reduces letters repeated more then three times, recode accented vowels, corrects "chat style" terms to the corresponding italian words and other "cleaning" operations.

After the preprocessing step, the document D is splitted into a list of sentences[2] $s_1, s_2, ..., s_k$ by the *GetSentences(D)* function detecting sentences boundaries on the basis of punctuation marks: '.', '!', '?'.

The overall polarity of the review D is hence obtained comparing the mean of the polarity strengths of its sentences (average sentence polarity, ASP) with the positivity threshold τ calculated on the basis of some experiments. The classifier returns the label string 'POS' if the computed average is above or equal τ, else the label string 'NEG' is returned. The most important sub-task here is the *Sen-*

[2] A sentence is a linguistic unit consisting of one or more words that are grammatically linked.

tenceLevelSC(s) function ,which assigns a polarity strength score, s_score_i to a sentence $s_i \in D$.

Algorithm 1: Sentiment Classification of a document.

procedure DOCUMENTLEVELSC(D, τ)
  $D \leftarrow TextCleaner(D)$ ▷ Preprocessing
  $\{s_1, s_2, ..., s_k\} \leftarrow GetSentences(D)$ ▷ Tokenization
  for $i \leftarrow 1, k$ **do** ▷ Classification of all sentences
    $s\_score_i \leftarrow SentenceLevelSC(s_i)$ ▷ (Algorithm 2)
  end for
  $asp \leftarrow (\sum_{i=1}^{k} s\_score_i)/k$ ▷ Average polarity
  if $asp \geq \tau$ **then**
    $res \leftarrow POS$ ▷ Positive label
  else
    $res \leftarrow NEG$ ▷ Negative label
  end if
  return *res* ▷ Document polarity class
end procedure

2.2.1 Sentiment Classification of Sentences

A sentence can be composed by one or more clauses separated by text elements that can be conjunctions("ma" = "but", "e"="and"...), punctuation (',' ";", ":"), or both. We assume that each clause is a portion of text that can express a sentiment independently from other clauses in the sentence. This assumption obliges us to search clauses separators during the sentence analysis. The classification of a sentence s is done by means of the function *SentenceLevelSC(s)*.

Its first step is a Parts of Speech tagging of s with a tool specifically designed for the Italian language. The *POS_Tagger(s)* returns a list *TS* of r pairs *(p,t)* where each p is a word of the sentence s and t a *tag* indicating the part of speech p represents.

The second step is a parsing of the sequence of pairs (p_i,t_i), $i=1,2,...r$. The overall sentence score is given summing positive and negative scores, named pt_scores_i, associated to the pairs (p_i,t_i), with $i = 1,2,...r$. Each pt_scores_i is the product of three factors:

1. the result of the $Weight(p_i,t_i)$ function based on a lookup table (see Table 3) to give different enhancements according to the part of speech tag t_i or to give a proper amplification to a negation term in p_i;
2. the result of the *Sign(F,sp)* function that, during the parsing, at each step $i=1,2,...r$, keeps or inverts the sign of the polarity based on the *local context* extracted from s and *TS*.
3. the result of the $WordLevelSC(p_i,t_i)$ function that calculates a positive or negative score for each word in the sentence independently by its local context (Algorithm 3).

Algorithm 2: Sentiment Classification of a sentence.

procedure SENTENCELEVELSC(s)
  $\{(p_1,t_1),(p_2,t_2),...,(p_r,t_r)\} \leftarrow POS\_Tagger(s)$
  $TS \leftarrow \{(p_1,t_1),(p_2,t_2),...,(p_r,t_r)\}$ ▷ tagged sentences
  $s\_score \leftarrow 0$ ▷ initialization
  $sp \leftarrow +1$ ▷ initial polarity sign (positive)
  for $i \leftarrow 1, r$ **do** ▷ sequence parsing
    $a \leftarrow Weight(p_i,t_i)$ ▷ amplifications
    $F \leftarrow GetLocalContext(i,s,TS)$ ▷ current window
    $ns \leftarrow Sign(F,sp)$ ▷ calculate new sign
    $w\_score \leftarrow WordLevelSC(p_i,t_i)$ ▷ (Algorithm 3)
    $pt\_score_i \leftarrow a \times ns \times w\_score$ ▷ score for(p_i,t_i)
    $s\_score \leftarrow s\_score + pt\_score_i$ ▷ accumulation
    $sp \leftarrow ns$ ▷ polarity sign update
  end for
  return *s_score* ▷ sentence polarity strenght
end procedure

2.2.2 Local Context Window

In the Algorithm 2, during the parsing, the function *GetLocalContext(i, s, TS)* maintains a window F of elements from *TS* and from s.
At each step $i = 1,2,...r$, the function puts in F the following elements of *TS*:

- the unigram (p_i,t_i) if $i = 1$,
- the bigram $(p_{i-1},t_{i-1}),(p_i,t_i)$ if $i = 2$,
- the trigram $(p_{i-2},t_{i-2}),(p_{i-1},t_{i-1}),(p_i,t_i)$ if $i > 2$

In addition, the function adds to the F clause separators punctuation marks in s, if any, preceding the word p_i. This window is used as local context of the currently analysed term.

2.2.3 The Function Sign(F, sp)

While some terms may seem to be inherently positive or negative, other lexical items near them in a text can change their base valence according to context. A "valence shifter" (Polanyi and Zaenen, 2006) is a combination of items in the sentence that flip the polarity of a term on the basis of its local context. According to the Italian grammar, there are several lexical items combinations acting as valence shifters.

Common examples are:

- negation at different levels of syntactic constituency;
- lexicalized negation in the verb or in adverbs;
- conditional, counterfactual subordinators;
- double negations with copulative verbs;
- modals and other modality operators.

Since polarities are represented as signed scores, the parser starts with a positive sign and flips it every time a valence shifter occurs. In this way, valence shifters detection gives a proper sign to the pt_score_i at each parsing step *i*. Sign inversions are decided by calling the function *Sign(F, sp)* passing as parameters the local context *F* and the *sp* variable storing the sign of the polarity up the preceding position *i-1*.

The function uses these arguments and a set of lists of Italian words to detects items combinations representing valence shifters by using a sequence of IF-THEN/IF-THEN-ELSE rules. Examples of words' lists used are:

LNEG={"non"},LN={"né","neppure","neanche", "nemmeno"}, L1={"solo","soltanto","solamente"}, L2={"meno"}, L3={"niente","nulla"},L4={"affatto"}, L5={"di"}, L6={"per"}, L7={"ma","anzi", "eppure", "peró", "tuttavia", "bensí"}, ...

If the presence of a valence shifter is detected in the current position, the sign to be given to the pt_score_i is flipped, else it is maintained. The sign is also reset each time the function find either conjunctions or punctuation marks in the local context, as colon and semicolon, individuating the start of a new independent clause into the analysed sentence.

2.2.4 Sentiment Classification of Words by Concept-based Scoring

The function *WordLevelSC(p,t)* on Algorithm 2 assigns a positive or negative prior polarity score to a single Italian word *p*, regardless of its local context, by using the stack of lexical resources. When it is not possible to perform Word Sense Disambiguation, then all senses of a word must be considered (or one can use heuristics such as taking the most frequent sense). Many words are *polysemous*, that is, they have multiple meanings. Moreover, the meanings of a word can convey sentiments with opposite polarities. Algorithm 3 is a sketch of the approach we use to find the prior polarity of an Italian word by using the concept-based approach. We remember that in the used resources both positive and negative polarity scores are values in the range [0.0,1.0]. At the beginning the word *p* is "normalized", i.e. it is transformed in a *lemma* (step 3). This morphological transformation is done either by means of Morph-IT database (see subsection 2.1.1) and some heuristics and it is required to correctly perform the successive searches in the lexical resources.

First of all, the algorithm tries to search a pair of positive and negative scores corresponding to the input (p,t) into the Auxiliar Opinion Lexicon (AOL), in which corrections and terms missing in the lexical resources are stored . The AOL database can also contain domain specific terms: in our application we have stored in it terms typically used in literary criticism and books reviews.

During the development of the proposed methodology, we have found that some terms in SentiWordNet have opposite polarity signs with respect to the corresponding Italian terms. Moreover, some errors are due to the POS tagger which in some cases applies wrong tags labelling as verbs some adjectives and some verbs as nouns. If the search in AOL is successful the function returns the greater of the two values, providing it with a proper sign (steps 4-12). If the term is missing in AOL, then the function *SearchLemmas()* search a list of lemmas synonyms of that given in input, with and without using also its postag *t*, in the Italian Wordnet.

Then, the corresponding English synsets are extracted from the English Wordnet and finally their scored versions, if any, are taken from SentiWordNet (step 13). If this attempt fails, our approach uses other semantic relations represented in the ontology trying to search "meanings" in the set we call the "cloud" of *p*. We define the cloud of *p* a group of terms semantically close as linked by relationships of type "IS-A", the set of hyperonyms (and secondarily) of *p* (step 15) Finally, if also this search gives an empty list of senti-synsets, the algorithm attempts to find some sense, if any, starting from alternative translations of *p* (step 18). Each one of the three used "search functions" can return a list of senti-synsets (or an empty list).

For example, if the Italian lemma is "bello" (with pos tag t= 'a', as adjective), then the list of senti-synsets returned by step 13 is:

$[SentiSynset('good.a.01'),$
$SentiSynset('nice.a.01'),$
$SentiSynset('beauty.n.01'),$
$SentiSynset('sweetheart.n.02'),$
$SentiSynset('good\_weather.n.01'),$
$SentiSynset('beautiful.a.01'),$
$SentiSynset('considerable.a.01')]$

In the list we can see that some postag do not match the tag t associated to p. The final part of the algorithm extracts information from each one of the *n*

senti-synsets:

- positive and negative posterior polarity scores (steps 24 and 25);
- sense_number (step 26);
- tag (step 27);

Two weighted means are then calculated:

$$poss = \frac{\sum_{t=1}^{n} w(s_t) \times pos(s_t)}{\sum_{t=1}^{n} w(s_t)}$$

$$negs = \frac{\sum_{t=1}^{n} w(s_t) \times neg(s_t)}{\sum_{t=1}^{n} w(s_t)}$$

with

$$w(s_t) = \frac{1}{2^{sense\_number-1}} \times ppt$$

where the first factor gives lower weight to less frequent meanings and viceversa, while the second factor is equal to 1.0 for senti-synsets having tag equal to tf, or act as a penalty for senti-synsets having postag different from t (we used ppt=0.75 in case of tags mismatch). The function finally returns the greater of the two weighted means, providing it with a proper sign. This value is the polarity given to the word p with tag t.

Algorithm 3: Sentiment Classification of a word.

```
 1: procedure WordLevelSC(p,t)
 2:     w_score ← 0.0
 3:     lemma ← Normalize(p,t)
 4:     (poss,negs) ← SearchAOL(lemma,t)
 5:     if min(poss,negs) ≥ 0.0 then
 6:         if max(poss,negs) = poss then
 7:             w_score ← poss
 8:         else
 9:             w_score ← −negs
10:         end if
11:         return w_score
12:     end if
13:     synsets_list ← SearchLemmas(lemma,t)
14:     if synsets_list = ∅ then
15:         synsets_list ← SearchCloud(lemma,t)
16:     end if
17:     if synsets_list = ∅ then
18:         synsets_list ← SearchTranslations(p,t)
19:     end if
20:     if synsets_list ≠ ∅ then
21:         poss ← 0.0,    negs ← 0.0
22:         ws ← 0.0       ▷ weights sum initialization
23:         for all ss ∈ synsets_list do
24:             ps ← ss.pos_score()
25:             ns ← ss.neg_score()
26:             sn ← ss.sense_number()
27:             tt ← ss.tag()
28:             if tt = t then
29:                 ppt ← 1.0
30:             else
31:                 ppt ← 0.75          ▷ tag mismatch
    penalty
32:             end if
33:             w ← 1/2^(sn−1) × ppt
34:             ws ← ws+w
35:             poss ← poss + w × ps
36:             negs ← negs + w × ns
37:         end for
38:         poss ← poss/ws
39:         negs ← negs/ws
40:         if min(poss,negs) ≥ 0.0 then
41:             if max(poss,negs) = poss then
42:                 w_score ← poss
43:             else
44:                 w_score ← −negs
45:             end if
46:             return w_score
47:         end if
48:     end if
49:     return w_score
50: end procedure
```

3 EVALUTATION OF THE CLASSIFIER

In order to evaluate the proposed method we have developed a fremework to perform experiments, including:

- a *grabber* to capture books reviews from amazon.it web pages,
- a relational database storing the *dataset*,
- the *classifier*,
- two *wrapper procedures* allowing to run the classifier in two possible modes: a *batch mode*, and an interactive *editor mode*.

3.1 The Dataset

A dataset of 8255 reviews in Italian language (related to 85 books of various authors and topics) has been created and stored in a relational database using the grabber. Each review has been written by a single Amazon user. Since Amazon reviews are accompanied by a rating (the number of stars), the number of reviews in the dataset for each rating level is shown in Table 1. The reviews in the dataset have an average number of 384 characters, with a standard deviation around 538. The longest review has 19263 characters. To evaluate and refine the classifier experimentally we

compare the sentiment polarity labels it returns with the rating associated to reviews. We consider positive reviews those where authors give a score above or equal to 4 stars. Negative reviews are those having a rating less or equal to 2 stars. The distribution of reviews per class is given in Table 2 showing that the dataset, while containing some thousands of reviews, is not perfectly balanced according to the rating. As you see the POS class has a majority (over two thirds) of highly positive reviews, while NEG class is well balanced with almost the same percentage of reviews of two stars and one star. The disproportion between the number of reviews at 4 or 5 stars (POS class) and those with 1 or 2 (NEG class) is "physiological" in the sense that on Amazon.it abundantly prevail, as is normal, the positive reviews. For classification purposes, this disproportion clearly puts the POS class in a "better position" than the NEG one because selecting randomly from the first there is a higher probability of highly positive sentiment strength.

Table 1: The dataset.

Rating (stars)	**5**	**4**	**2**	**1**
# reviews	4342	1963	508	494
Occurrences per review (Avg.)				
adjectives	6.6	7.1	7.5	6.8
nouns	13.5	13.9	15.9	14.3
verbs	8.5	8.6	11.0	10.6
adverbs	0.8	0.7	0.9	0.9

Table 2: Subdivision of the reviews in classes.

Polarity class	**Rating**	**#**	**%**
POS rating ≥ 4	5 stars	4342	68.87%
	4 stars	1963	31.13%
	Total	6305	100.0%
NEG rating ≤ 2	2 stars	508	50.70%
	1 star	494	49.30%
	Total	1002	100.0%

3.2 Creation of the Auxiliar Opinion Lexicon During Early Experiments

The lexical resources used include an auxiliary opinion lexicon (AOL) were we stored

- Missing terms
- Corrections
- Domain specific terms

This resource has been created by the following procedure. First, for each review in the dataset, we applied an Italian Part-Of-Speach tagging and, after a lemmatization, the terms tagged as adjective, nouns, adverbs and verbs has been searched in the Multiwordnet/Sentiwordnet. If the search of a term fails save it in a file (missing terms). Then, a manual annotation of each term in the file with a priori polarity has been performed. In addition, we have searched on other several websites different from Amazon, more reviews of books written in Italian language and we have added to the file terms we have considered domain specific or typically used in review writing (domain specific terms). Moreover, during early experiments we saved on a second file the polarity assigned to each distinct noun, adjective, adverb and verb during the classification and then we manually checked if their scores were consistent with corresponding sentiment polarity normally understood in the Italian language. Finally, for each term we saved only the stemmed version in order to reduce file size and to facilitate the search. As a future work we consider to investigate the costruction of the domain specific part of the AOL following the approach given in (Agathangelou et al., 2014).

3.3 Experiments

In order to refine the classifier we have developed two wrapper procedures allowing to run it either in interactive mode on a single review or in batch mode on large lots of reviews. The set of weights and the positivity threshold τ on which our classifier depends has been experimentally determined by using two wrapper procedures of the classifier. The first wrapper allows the interactive use of the classifier by means of a GUI whose visual components realize a text Editor. By using this editor we can open an existing text file or directly type and change a text and then we can run the classifier to obtain a detailed trace of the text analysis. The interactivity allows to study misclassification errors and to correct them acting on parameters, e.g. weight used in parsing of sentences (algorithm 2) and the positivity threshold τ used in the algorithm 1. This experimental modality has allowed us to refine the set of IF-THEN-ELSE rules used in the parsing process (algorithm 2) and to find a balanced set of weights for the Sentence Level SC previously described. Experiments show that too high weights values tend to give document level average scores with too large standard deviations, not allowing to find a good separation threshold between positive and negative classes. It was also noted that one of the most influential weights for a more correct classification, are those given to adjective and adverbs, and to the negatives (the term 'no' and the first successive terms) that are more frequently present in negative sentences. The set of weights found, reported in table 3 showed

Table 3: Look up table of the Weight() function.

Part of Speech	Weight
'JJ'	1.1
'RB'	1.1
'VB'	1.06
'VBN'	1.06
'VBG'	1.03
'NN'	1.06
Negation	1.1

a good behaviour of the Sentence Level SC.

Taking in account the different review rating distribution as described in table 2 we have used the batch modality to run the classification task on lots of reviews in order to estimate the threshold τ to be used in the Document Level SC, In this set of experiments we have randomly selected :

- a set of 50 reviews with 5 stars rating
- a set of 50 reviews with 4 stars rating
- a set of 50 reviews with 2 stars rating
- a set of 50 reviews with 1 star rating

obtaining the following table of average document polarity scores for each class:

Table 4: Batch mode experiments results on samples of given rating.

Polarity class	Rating (stars)	Average document polarity
POS	5	+0.098443
	4	+0.035872
NEG	2	-0.02612
	1	-0.06583

Using these results and on the basis of the distribution of the rating in the POS and NEG class, we have calculated the expected Document polarity strength for the POS an NEG class as the weighted mean:

$$\mu_{POS} = (0.098 \times 68,87 + 0.035 \times 31.13)/100 = 0.079$$

$$\mu_{NEG} = (-0.03 \times 50,7 - 0.07 \times 49.3)/100 = -0.045$$

where weights are rating percentages.
Finally, the τ parameter has been determined as the arithmetic mean of the above values:

$$\tau = (\mu_{POS} + \mu_{NEG})/2 = (0.079 - 0.045)/2 = 0.017$$

a threshold value slightly positive that adjusts imbalance between the POS and NEG classes.

The final set of experiments we performed has been finalized to statistically estimate the accuracy of the classifier. For this purpose, we have applied the classifier in batch mode on six lots of 200 randomly selected reviews each and 3 of these lots having rating greater or equal 4, while the other three having rating lesser or equal 2. The results of these experiments are summarized in table 5, were TP, FN, TN, and FP are counters indicating True Positive, False Positive, True Negative and False Positive obtained from the classifications.

Table 5: Results of classification of random samples of reviews.

Random sample	Correctly classified	Incorrectly classified
Pos#1	168	32
Pos#2	171	29
Pos#3	174	26
Sums	**TP = 513**	**FN = 87**
Neg#1	157	43
Neg#2	162	38
Neg#3	163	37
Sums	**TN = 482**	**FP = 118**

On the basis of the results obtained by experiments the estimated *accuracy* is given by :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{995}{1200} = 82.91\%$$

while in terms of *precision* π and *recall* ρ we have:

$$\pi_{pos} = \frac{TP}{TP+FP} = \frac{513}{631} = 81.29\%$$

$$\rho_{pos} = \frac{TP}{TP+FN} = \frac{513}{600} = 85.5\%$$

$$\pi_{neg} = \frac{TN}{TN+FN} = \frac{482}{569} = 84.71\%$$

$$\rho_{neg} = \frac{TN}{TN+FP} = \frac{482}{600} = 82.0\%$$

3.4 Baseline Test

In order to have a term of comparison we have realized a baseline test using a reduced lexicon of Italian words we have found on Github[3]. This lexicon is represented by two text files: *pos.words.txt* and *neg.words.txt* containing a list of 1382 positive Italian words and a list of 3052 negative Italian words respectively. We call this lexicon OLIT here. The test consisted into the following experiment. We have randomly selected from the dataset one thousand positive reviews (i.e. having rating ≥ 4) and one thousand negative reviews (i.e. having rating ≤ 2). Each

[3]https://github.com/steelcode/sentiment-lang-italian

review has been preprocessed with the same preprocessor used in the classifier (first step of Algorithm 1). Then for each review it was done the count of positive and negative words of OLIT found in the text and it has been classified positive if the count of positive words exceeds that of the negative, classified negative vice versa, classified neutral if the two counts are equal. The run of this test has correctly classified positive reviews in 78.10% of cases (compared with 85.5% of the classifier) and correctly classified negative reviews in 49.63% of cases (compared with 84.71% of the classifier).

3.5 Considerations on Misclassifications of Books Reviews

Several misclassifications errors have been analysed by tracing the classification process in interactive modality. We observed that some errors are due to imperfections in the components of the system or to the lack of some functionalities as a ”spelling correction” module in the framework. We noticed in fact a certain sensibility to ”typing errors” like missing spaces between words, or missing or wrong letters in them that our preprocessing step cannot resolve. Other sources of errors inside the system are given by the POS tagger that in some cases assigns erroneous tags. But the majority of misclassification we observed are due to the nature of the documents analysed. Many reviews contain ”sarcasm” or comparisons with previous works of the author of the reviewed book. Very often, users insert into negatively rated reviews positive sentences regarding Amazon’s delivery service and vice versa. Sometimes the review is on the book format rather than its contents. Moreover books reviews are different from other products reviews, like those in the ”Appstore for Android” or in” Electronics & Computers” Amazon departments. In reviewing a book in fact the user of Amazon often has a tendency to show his skills as a ”literary critic”, and hence periods are sometimes very long and articulated and generally complexes. Sometimes the ”review” is to bring the plot of the book without providing a proper contribution to the document.

4 CONCLUSIONS

In this work it has been shown the efficacy of a new proposed lexicon-based approach for the Italian language. This approach prove that multi-linguistic ontologies based on the expand-model can be used as interfaces towards opinion lexicon for the English language like SentiWordnet. Differently from other schema that use SentiWordnet simply like a dictionary to score documents as in a ”bag of words” model, our approach uses ontologies to find terms semantically close to the one to be scored, in the so called ”cloud” of the word. The sentence parsing process based on a window of up to a trigram of words, tags and punctuation marks helps in the correct sentiment classification of sentences detecting, by a proper set of rules, the presence of valence shifters based on Italian grammar. Moreover, errors and missing terms, as well as dominion specific terms can be corrected by using an Auxiliary Opinion Lexicon integrating the main resources. Important components of the whole process are represented by the morphological resource used for lemmatization and normalization of words before their search in the linguistic resources, and also the POS tagger for Italian language.

REFERENCES

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):1–34.

Agathangelou, P., Katakis, I., Kokkoras, F., and Ntonas, K. (2014). Mining domain-specific dictionaries of opinion words. In *15th International Conference on Web Information System ngineering (WISE 2014)*, pages 47–62.

Agerri, R. and Garcia-Serrano, A. (2010). Q-wordnet: Extracting polarity from wordnet senses. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*.

Bautin, M., Vijayarenu, L., and Skiena, S. (2008). International sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2008)*, pages 19–26.

Bentivogli, L., Girardi, C., and Pianta, E. (2002). Multiwordnet, developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, pages 293–302.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning (COLT92)*, pages 144–152.

Cambria, E., Olsher, D., and Rajagopal, D. (2014). Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twentyeight AAAI conference on artificial intelligence (AAAI-14)*, pages 1515–1521.

Casoto, P., Dattolo, A., and Tasso, C. (2008). Sentiment classification for the italian language: a case study on movie reviews. *Journal Of Internet Technology*, 9(4):365–373.

Compagnoni, S., Demontis, V., Formentelli, A., Gandini, M., and Cerini, G. (2007). *Language resources and linguistic theory: Typology, second language acquisition, English linguistics (Forthcoming), chapter*

Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. Franco Angeli Editore.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web (WWW '03)*, pages 519–528.

Dinu, L. P. and Iuga, I. (2012). The naive bayes classifier in opinion mining: In search of the best feature set. In *13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2012)*, pages 556–567.

Domingos, P. and Pazzani, M. J. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130.

Dragut, E. C., Yu, C., Sistla, P., and Meng, W. (2010). Construction of a sentimental word dictionary. In *ACM International Conference on Information and Knowledge Management (CIKM 2010)*, pages 1761–1764.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Esuli, A., Sebastiani, F., and Baccianella, S. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC '10)*, pages 2200–2204.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Garcia, M. and Gamallo, P. (2014). Citius: A naive-bayes strategy for sentiment analysis on english tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 171–175.

Hassan, A., Korashy, H., and Medhat, W. (2014). Sentiment analysis algorithms and applications - a survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 168–177.

Kanayama, H. and Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, pages 355–363.

Kennedy, A. and Inkpen, D. (May 2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.

Kim, S.-M. and Hovy, E. (2006). Identifying and analyzing judgment opinions. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL-06)*, pages 200–207.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the European Conference on Machine Learning (ECML-98)*, pages 4–15.

Littman, P. and M.L., T. (2002). Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical report, National Research Council Canada, Institute for Information Technology.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, San Rafael, US.

Liu, L. and Özsu, M. T. (2009). *Encyclopedia of Database Systems*. Springer.

Meyer, D., Leisch, F., and Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1–2):169–186.

Mihalcea, R., Banea, C., and Wiebe, J. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the Association for Computational Linguistics (ACL 2007)*, pages 976–983.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pages 79–86.

Polanyi, L. and Zaenen, A. (2006). Contextual valence shifters. In Croft, W. B., Shanahan, J., Qu, Y., and Wiebe, J., editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, chapter 1, pages 1–10. Springer Netherlands.

Shanmuganathan, P. and Sakthivel, C. (2015). An efficient naive bayes classification for sentiment analysis on twitter. *Data Mining and Knowledge Engineering*, 7(5).

Strapparava, C. and Valitutti, A. (2004). Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1083–1086.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Takamura, H., Inui, T., and Okumura, M. (2006). Latent variable models for semantic orientations of phrases. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 201–208.

Vossen, P. (1998). Introduction to eurowordnet. *Computers and the Humanities*, 32(2):73–89.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing (HLT/EMNLP 2005)*, pages 347–354.

Yi, H., Jianyong, D., Xiaoming, C., and Bingzhen, Pei andRuzhan, L. (2005). A new method for sentiment classification in text retrieval. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IIJCNLP-05)*, pages 1–9.

Zagibalov, T. and Carroll, J. (2008). Automatic seed word selection for unsupervised sentiment classfication of chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08)*, pages 1073–1080.

Zanchetta, E. and Baroni, M. (2005). Morph-it! a free corpus-based morphological resource for the italian language. In *Proceedings of Corpus Linguistics 2005*.