Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 126 (2018) 626–635

Procedia Computer Science

www.elsevier.com/locate/procedia

22nd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A simple algorithm for the lexical classification of comparable adjectives

Matteo Cristani[1], Ilaria Chitó[1], Claudio Tomazzoli[1], Margherita Zorzi[1]

[a]*Dipartimento di Informatica, Università di Verona, Strada Le Grazie 15, Italy*

Abstract

Lexical classification is one of the most widely investigated fields in (computational) linguistic and Natural language Processing. Adjectives play a significant role both in classification tasks and in applications as sentiment analysis. In this paper a simple algorithm for lexical classification of *comparable adjectives*, called MORE (coMparable fORm dEtector), is proposed. The algorithm is efficient in time. The method is a specific unsupervised learning technique. Results are verified against a reference standard built from 80 manually annotated lists of adjective. The algorithm exhibits an accuracy of 76%.

1. Introduction

The interest onto the nature of adjectives in statistical natural language processing (SNLP) lies upon two major issues that are challenging scholars in the field.

First of all adjectives are relevant in those applications that are stricly related to the expression of *judgments*, especially in social networks. The most common application field of this notion is *sentiment analysis* [7, 12], but other investigations have focused upon this theme as well [15, 28, 2]. In other terms the most important and recent applications of SNLP found on understanding of adjectives.

Adjectives belong to many different types, and the detection of the adjective type is essential to the interpretation of expressions based upon that adjective. A part from the declination of adjectives in number, genre, possibly case and possibly person (depending on language), they also may exhibit different grades, through the adjective morphology (more, and most). Adjectives with grades (named *comparable*) represent the essence of judgments. They express indeed the relation between an individual and a *measured property*. "John is tall" means that the individual named John has a (vaguely) defined value on (vaguely) defined area of the measure of the property of height.

Adjectives are often used to *modify nouns*, or, more specifically for the adjectives used to provide judgments, to qualify them. Therefore it would be very significant to have a method that allows to detect whether an adjective is comparable or not. The benefit of such a technique is twofold, for the very existence of such an adjective in a sentence proves that the sentence expresses a judgment, and it is therefore interesting from a sentiment analysis viewpoint.

10.1016/j.procs.2018.07.297

Moreover the technique that we are defining here is able to provide a specific information about an adjective, an useful tool for those who aim at building an annotated dictionary, with specific semantic data associated to lexicon.

From the pure semantical viewpoint there are numerous investigations that give account to the specific nature of the adjectives. There are also lexical resources online, that provide the basic information pieces about usage of the words.

Nevertheless, the classification of these adjectives in comparable and incomparable, though existing in many specific languages share, in general, the following drawbacks: (a) they classify a very limited number of adjectives, claimed to be the most common, and (b) are handmade and therefore irreproducible.

In this paper a simple algorithm for lexical classification of *comparable adjectives*, called MORE (coMparable fORm dEtector), is proposed. MORE does not suffer of any of the above mentioned limits. It is completely automatic and does not require either training or supervision. It is efficient in computational terms and can be easily applied to languages different than english (see Section 5).

Statistical reliability measures upon the above mentioned algorithm are difficult to compute without a gold standard. In fact, the novel method presented here is also compared against human behaviour in order to establish accuracy, precision and recall of the method. Obtained results are quite encouraging from several viewpoints. They can be easily used to detect the behaviour of adjectives, and to classify them, even onto a language that is not known or pre-defined, under the condition of having a list of the adjective that we aim at classifying and a corpus (or set of corpora) on which we aim at looking for them.

The rest of the paper is organized as follows. Section 2 briefly introduces the background of the method and describes the algorithm MORE. Section 3 introduces the experiment design and discusses the results. Finally Section 4 reviews current literature, and Section 5 discusses results and takes conclusions.

2. MORE: an algorithm for adjective classification

In the following, some well-known notions of linguistic and text processing are used. As usual, with *lexical class* or part-of-speech, one intend a category of lexical items which have similar grammatical properties. *Suffix* and *prefix* of a word identify results of usual operations on strings. The *stemming* of a lexical item is its reduction to a particular root form (that not necessarily corresponds to the morphological root) [21]. Several stemming algorithm have been introduced, for several languages. In this paper the most famous (and efficient) English stemmer, by Porter [22], is adopted.

In this section the procedure MORE, designed to classify adjectives is described. Following the same approach adopted in previous investigations in SNLP [9, 30, 32], the proposed procedure is able to couple feasibility and efficiency. MORE is completely automatic, unsupervised and does not need any training phase. These properties make the software computationally efficient (see Section 3.5) and very easy to specialise to different data sets and languages. To simplify the presentation, the algorithm is splitted into two distinct phases (subroutines), called here *acquisition phase* and *sifting phase* respectively.

The *acquisition* phase exploits the powerful of a web search engine as Bing to retrieve information about a list of adjectives. The sifting phase try to correct the information by means of suitable thresholds, described below. In the following, each phase is accurately explained. The main procedure MORE is built from the sequential application of the two phases.

The input of the global procedure (and in particular of the acquisition phase) is represented by the following:

- **A list $\mathcal{L}$ of English adjectives.** This is the list of word to classify. It does not contain adjective representing nationalities (e.g. Italian, English, Spanish...).
- **Two support lists $\mathcal{S}_u$ and $\mathcal{S}_c$ of adjectives**. These lists constitute the support knowledge base for the classification algorithm. In particular, the first list includes uncomparable adjectives and the second list includes adjectives in comparative or superlative forms. These lists are included in several reference handbooks of style, such as [6] as the *most common* English adjectives, and are split up into comparable and uncomparable.

Let describe now in details the two phase of MORE.

Acquisition phase: preprocessing and web-based information retrieval

For each candidate adjective a from the input list $\mathcal{L}$, the following steps are performed:

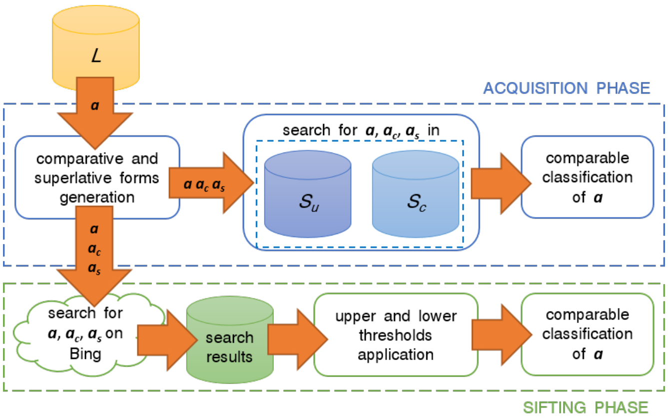

Fig. 1. MORE procedure, overall picture.

1. Verify if a is contained in one of the two support input lists $\mathcal{S}_u$ or $\mathcal{S}_c$. If so, then classify the adjective according to the membership list and repeat step 1 with a new adjective. Otherwise, proceed to the next step.
2. According to the grammatical rules of the English language, verify if its **suffix** is "-er" (resp. "-est"). If so, check if there is another adjective with *the same stem* but with suffix "-est" (resp. "-er"). If the latter is present then classify the adjective a as comparable and repeat step 1 with a new adjective. Otherwise proceed to the next step.
3. Generate **comparative** and **superlative** forms of a by following the grammatical rules of the English language. Denote comparative and superlative forms as a_c and a_s respectively. Verify if at least one of the forms a_c or a_s belongs to the list of comparable adjectives $\mathcal{S}_c$. If so, then classify the original adjective a as comparable and repeat step 1 with a new adjective. Otherwise, proceed to the next step.
4. Search for the adjective a and its two comparable forms a_c and a_s on the **Bing search engine**[1]. For each version of the adjective, store the value of the search result, i.e. the number of results obtained by the web query. This information will be used in the sifting phase for the classification of the original adjective.

Sifting phase

The second phase of MORE is devoted to the analysis of the results of the web search. Indubitably, the web is a powerful source of knowledge, but the information retrieved has to be carefully processed. The main goal of the sifting phase is to exploit the web query result to divide comparable adjective from non comparable ones.

To this end, some iterations of a subroutine are perfomed to progressively classify adjectives that are not yet defined after the acquisition phase. Roughly speaking, one verifies if the number of result retrieved by Bing for the adjective a and its comparable forms a_s and a_c "survive" to two variable thresholds, called **U** (*upper*) and **L** (*lower*), along different iterations.

The upper threshold is defined as $labelupperU = \frac{RES\_BASE}{10^i}$, where: RES_BASE indicates the search result *of the original adjective a*, i is initially set to 1 and, at each subsequent application of the formula i is increased by one.

The lower threshold is determined as $L = \frac{RES\_BASE}{10^{DIGITS-i}}$ where DIGITS represents the number of digits of RES_BASE; RES_BASE and i behave as above. According to the increasing of i, U decreases and L increases. This step is repeated as long as adjectives are still indefinite. At the first iteration both U and L are applied. At the following iteration a single threshold, alternately, is applied, until the number of the indefinite adjectives is zero.

Let's see in detail the steps performed in the sifting phase for each still undefined adjective of the list $\mathcal{L}$. $B(a_c)$ and $B(a_s)$ denote the search results of comparative (a_c) and superlative forms (a_c) respectively.

1. Verify if the search results of comparative and superlative forms are both greater than or equal to the search result of the original adjective $((B(a_c) \geq RES\_BASE) AND (B(a_s) \geq RES\_BASE))$. In this case, conclude that the original adjective is comparable, classify it and repeat step 1 with a new adjective. Otherwise, go to the next step.

[1] between double quotes

ITER	COMP.	UNC.	IND.
0	1,664	591	25,065
1_U_L	2,587	10,351	14,382
2_U	4,345	10,351	12,624
3_L	4,345	14,430	8,545
4_U	8,952	14,430	3,938
5_L	8,952	16,553	1,815
6_U	10,371	16,553	396
7_L	10,371	16,760	189
8_U	10,557	16,760	3
9_L	**10,557**	**16,763**	**0**

Table 1. Upper threshold: first run.

ITER	COMP.	UNC.	IND.
0	1,664	591	25,065
1_H_L	2,587	10,351	14,382
2_L	2,587	14,430	10,303
3_H	4,345	14,430	8,545
4_L	4,345	16,757	6,218
5_H	8,748	16,757	1,815
6_L	8,748	17,077	1,495
7_H	10,054	17,077	189
8_L	10,054	17,258	8
9_H	**10,062**	**17,258**	**0**

Table 2. Lower threshold: first run.

2. At this step results of web search are verified against the threshold U or L, according to the current iteration.

   - If the upper threshold is applied, check whether the search results of the comparative **and** superlative forms are both greater than or equal to the search result of the original adjective divided by the upper threshold ($B(a_c) \geq \frac{RES\_BASE)}{U}$ AND $B(a_s) \geq \frac{RES\_BASE)}{U}$). In this case, classify the original adjective a as comparable, otherwise a remains undefined and repeat step 1 with a new adjective.
   - If the lower threshold is applied, verify whether the search result of the comparative **or** superlative form is less than or equal to the search result of the original adjective divided by the lower threshold ($B(a_c) \geq \frac{RES\_BASE)}{L}$ OR $B(a_s) \geq \frac{RES\_BASE)}{L}$). If so, the original adjective a is classified as not comparable. Otherwise, it considered still undefined and repeat step 1 with a new adjective.

Since the application order of the thresholds leads to different classification results, in experiment performed in the following the sifting phase is applied twice: in the first, one applies first the upper threshold and in the second one applies fist the lower threshold.

3. The Experiment

This section describes an experiment performed on a list $\mathcal{L}$ of about 27k candidate adjectives. To evaluate MORE performances, a *gold reference standard*, collecting manual classifications of a subset of adjectives by a sample of students in *foreign languages* subject is defined.

3.1. Experiment design

The input data of the experiment are the following instance of lists $\mathcal{L}, \mathcal{S}_u, \mathcal{S}_c$:

- 27,320 English adjectives: this list has been created by Ashley Bovan using Moby Part-of-Speech II, a list that contains 233,356 words fully described by part(s) of speech[2], and the UK Advanced Cryptics Dictionary, a word list compiled for the Crossword Community[3]. In particular the author verified whether adjectives with 1, 2, 3 and 4 syllables on the Moby Part-of-Speech II list were contained in the UK Advanced Cryptics Dictionary and included they in the list accordingly. Adjectives representing nationalities are excluded);
- 718 incomparable adjectives from the Oxford English Dictionary;
- 678 adjectives in comparative (339) and superlative (339) forms from the Oxford English Dictionary.

The instances described above are processed by the subroutine of the acquisition phase, that collect information about adjectives in $\mathcal{L}$ by querying Bing. After the information retrieval, two runs of the sifting phase of MORE are

[2] http://www.gutenberg.org/ebooks/3203
[3] http://www.crosswordman.com/wordlist.html

performed since, as previously said, the order in which thresholds U and L are applied matters. Data about the number of adjectives classified as comparable, incomparable and undefined, for each iteration of the sifting phase of MORE are in Tables 1 and 2. Table 1 describes results for the first run, where at the second iteration (2_U) the upper threshold is firstly updated.Similarly, Table 2, describes results for the seconf run, where at the second iteration (2_U) the lower threshold is firstly updated. Obtained results are verified through a reference standard, described in Section 3.2. Results are in Section 3.4.

3.2. *Testing* MORE *performances:the definition of a gold standard*

To prove the accuracy of the designed method, a gold standard, i.e. a reference truth against to we compared MORE results has been created.

The idea is simple: to take a subset of the adjective in list $\mathcal{L}$ and provide a *manual revision*, i.e. a manual annotation of adjectives with their lexical classification. After the manual revision, the sample of annotated adjectives is classified by some evaluation methods, to assign to each lexical item some *predicted class* representing the reference standard to compare MORE performances. Test generation and adjective classification are described in Section 3.2.1 and Section 3.2.2.

3.2.1. *Test generation*

72 "test" starting from a subset of **672** adjectives from the list $\mathcal{L}$ are automatically created.

In the generation of reference tests, a number of aspects have to been considered. Main steps of the generation are summarized here.

1. The **25,065** adjectives in $\mathcal{L}$ have been decreasingly ordered according to their search results on Bing (the number of results retrieved by the search engine).
2. *Quartiles* have bben calculated on this ordered list. The quartiles allow us to divide the list into 4 equal groups. The first contains the most widespread and known adjectives, while the last contains the rarest and least used ones. For this purpose one has performed the following calculations.

   (a) Both the *absolute frequencies*, i.e. the number of adjectives corresponding to the same search result, and the *cumulative frequencies* have been calculated i.e., for each search result, its absolute frequency *plus* the absolute frequency of its predecessor.
   (b) One determines the positions of the quartiles according to the formula $posQ_i = (\frac{N+1}{4}) \cdot i$, where $N = 25,065$ is the total number of adjectives in $\mathcal{L}$ and $i = 1, 2, 3$. Thus one has: $posQ_1 = 6,266$, $posQ_2 = 12,533$ and $posQ_3 = 18,799$.
   (c) Quartiles are defined as Q_i = *first search result with cumulative frequency greater than or equal to the value of* $posQ_i$, (i = 1, 2, 3).
   One obtained the following values:Q_1 = **13,800,000**, Q_2 = **1,380,000**, Q_3 = **41,200**.

3. The ordered list of adjectives is divided into 4 equal groups according to quartile values.
4. One randomly mixed the adjectives within each group.
5. 72 tests have been built. Each test is composed by **40** adjectives, coming from different groups. In particular: **24** of the 1st group; **8** of the 2nd group; **4** of the 3rd group; **4** of the 4th group.
6. 72 textual format (.txt) tests have been generated. They contain:

   - a unique identification number of the test
   - a list of 40 adjectives, each with its own unique identifier and followed by the choices:
     - **?** = adjective classified as unknown;
     - **C** = adjective classified as comparable;
     - **U** = adjective classified as incomparable;
     - **I** = adjective classified as indefinite.

Notice that a larger number for adjectives in the 1st group is chosen, since because one expects them to be classified correctly, since they are the most used/known. According to the same logic, a low number of adjectiveshas been chosen for the 3rd and 4th group because they are less frequent and the risk of error in their classification potentially increases. Moreover, to exploit cross opinion among reviewers, the adjectives are repeated in different tests: in particular, **3**, **9**, **18**, **18** times for the adjectives of the 1st, 2nd, 3rd and 4th group respectively.

Therefore, for the test list of adjectives (672), one has the following distribution among different groups: the number of adjectives from each group are: **576** for the 1st group, **64** for the 2nd group, **16** for the 3rd group and **16** for the 4th group.

The manual classification has been entrusted to 72 competent students, that annotated each adjective of the test according to the categories **?**, **C**, **U**, **I** described above.

3.2.2. Adjective Classification

One adjectives have been manually annotated, manual revisions require to be uniformed. In other words, to each adjective we assign a *predicted class*, i.e. the expected results against to we will compare the output of automatic classification. Since different reasonable classification of manual encodings are possible, we use four evaluation methods to classify adjectives from tests. This implies that, to each adjective, different predicted class are possibly assigned. The goal is to obtain uniform and fair reference standard and so a better understanding of the experiment results. In other words, we “weight” each adjective by the following criteria, after a complete discharging of the “unknown” category.

1. **ranked**: sums the number of votes obtained for each category and classifies the adjective according to the category that has obtained a higher number of votes. In the event that several categories have the same maximum value, the adjective is classified as “indefinite”.
2. **polling**: sums the number of votes obtained for each category and classifies the adjective according to the category that has obtained a higher number of votes. In the event that several categories have the same maximum value, the adjective is discarded.
3. **majority**: like the *polling* technique above detailed, where during the evaluation both “unknown” and ”indefinite” categories are not considered;
4. **unanimity**: classifies the adjective only if it has votes for only one category, or if there is only one category with a non-zero value.

Since the number of tests actually completed by the students is **20** (on 72 generated test) then the number of adjectives classified is**800**. After the application of the classification methods described above, the number of adjectives goes from 800 to **528** since the votes relative to each adjective repeated in more tests are added together; as a consequence each adjective from now on is considered in a univocal way.

3.3. Evaluation of MORE *performances*

The classification of the adjectives, carried out by the MORE processing, is now compare with results obtained from the evaluation methods of the gold standard described above.

In the following, for each adjective, the *actual class*, dubbed as **A**, denotes the class assigned by MORE and, generally, the *predicted* class denotes the class assigned by manual reviews. Since, as explained above, reference test can be evaluated by different techniques, the following notations for the predicted classes will be used:

- **TR** for tests evaluated with the ranked technique;
- **TP** for tests evaluated with the polling technique;
- **TM** for tests evaluated with the majority technique;
- **TU** for tests evaluated with the unanimity technique.

One will denote as $A \rightarrow \mathcal{T}$ ($\mathcal{T} \in \{\mathbf{TR}, \mathbf{TP}, \mathbf{TM}, \mathbf{TU}\}$) the comparison of the automatic evaluation (the actual classification) with the gold standard evaluation by method $\mathcal{T}$.

As usual in error classification, the comparison is represented in terms of *confusion matrices*. In particular, four confusion matrices, have been generated, one for each comparison $A \rightarrow \mathcal{T}$, combining algorithmic classification with each gold standard evaluation method. Since each adjective belongs to three possible classes (comparable, incomparable

	C	**U**	**I**
C	CC	CU	CI
U	UC	UU	UI
I	IC	IU	II

Table 3. Ternary confusion matrix

measures (%)	**A → T_R**	**A → T_P**	**A → T_M**	**A → T_U**
RE_1	41.1	51.4	62.7	51.0
PR_1	76.5	76.5	76.4	78.9
ACC_1	58.1	61.2	65.0	60.5

Table 4. Evaluation of the test classifiers: first transformation.

and indefinite), one first creates four **ternary confusion matrices** similar to the one in the Table 3. Each row of the matrix represents an instance of the actual class (i.e. the class assigned by MORE)while each column represents an instance of a predicted class assigned, by the ranked, polling, majority or unanimity method. One denotes possible values of the matrix in the gold standard, so that **CC**, **UU** and **II** are the number of adjectives correctly classified as "comparable" by the predicted class, whilst the crossed ones represent the false measures. For instance **CU** is the number of "comparable" adjectives classified as "incomparable" by the predicted class, textbfCI is the number of "comparable" adjectives classified as "indefinite" by the predicted class and so on.

Since the ternary confusion matrix does not allow to perform a detailed analysis of the results, then a transformation into a (standard) **binary confusion matrix** is performed. Notice that several transformations are allow. In particular, each ternary matrices can be mapped in four possible binary ones.

Overall, each transformation corresponds to a possible "aggregation" of predicted classes. As for the ternary matrix, each row represents an instance of the actual class while each column represents an instance of a predicted class. The values contained in the matrix are classified in a standard way as *true positives, false positives, false negatives* and *true negatives*. In our setting, these common taxonomy is defined as follows:

- **TP** (true positives) is the number of adjectives correctly classified as "comparable" by the expected class;
- **FP** (false positives) is the number of "non-comparable" adjectives erroneously classified as "comparable" by the expected class;
- **FN** (false negatives) is the number of "comparable" adjectives erroneously classified as "not comparable" by the expected class;
- **TN** (true negatives) is the number of adjectives correctly classified as "non-comparable" by the expected class.

Mappings onto binary confusion matrices can be done by means of four aggregation rules. The first aggregation transforms

For the sake of space one omits to show all output binary matrices and related values.

3.4. Results

Results have been quatified by meas of well-known metrics:

- **Recall (RE)**: is the ratio of correctly predicted positive observations to the all observations in actual positive class, $RE = \frac{TP}{TP+FN}$
- **Precision (PR)**: is the ratio of correctly predicted positive observations to the total predicted positive observations, $PR = \frac{TP}{TP+FP}$
- **Accuracy (ACC)**: is the ratio of correctly predicted observation to the total observations, $ACC = \frac{TP+TN}{TP+TN+FP+FN}$

The tables 4, 5, 6 and 7 shows the evaluation of the test classifiers with respect to the four binary confusion matrix generated by the four transformation methods described in the previous section.

First and fourth transformation method of a ternary confusion matrix into a binary one, show the values of the best performances. The first and the fourth transformation methods both provide the classification of the comparable adjectives into to the positive class and the incomparable adjectives into the negative class. However, the first method

measures (%)	**A → T_R**	**A → T_P**	**A → T_M**	**A → T_U**
RE_2	37.1	49.4	68	49.6
PR_2	47.2	47.2	49.1	44.3
ACC_2	59.5	60.2	63.5	60.7

Table 5. Evaluation of the test classifiers: second transformation.

measures (%)	**A → T_R**	**A → T_P**	**A → T_M**	**A → T_U**
RE_3	42.9	11.1	0	12.5
PR_3	3.0	1.2	0	1.3
ACC_3	61.6	78.0	97.4	78.3

Table 6. Evaluation of the test classifiers: third transformation.

measures (%)	**A → T_R**	**A → T_P**	**A → T_M**	**A → T_U**
RE_4	61.7	61.7	62.7	61.2
PR_4	77.4	77.4	77.3	79.9
ACC_4	63.6	63.6	64.6	63.4

Table 7. Evaluation of the test classifiers: fourth transformation.

ITER	**TIME (ms)**
1_U_L	1989
2_U	2084
3_L	2488
4_U	1424
5_L	1694
6_U	1635
7_L	2434
8_U	2680
9_L	1691

Fig. 2. Time for upper threshold first run.

ITER	**TIME (ms)**
1_U_L	1989
2_L	1474
3_U	2180
4_L	2115
5_U	1736
6_L	1603
7_U	2234
8_L	2398
9_U	2220

Fig. 3. Time for lower threshold first run.

also includes indefinite adjectives in the negative class, and this has a minimal effect on the results of the calculated measures.

In addition to experimental results, one believes that the best performing aggregations are preferable also because they are more consistent with the goal to classify an adjective as comparable or not.

3.5. *On the feasibility of* MORE

MORE computational performances will be now briefly analyzed. Since MORE exploits web search, it is impossible to give its abstract complexity in a completely formal way. Not withstanding, one can sketch an reasonable evaluation of the time complexity. The implementation involves only direct access data structure (as hash tables): then, one can assume constant time $O(1)$ for all operations involved in the procedure (if then else guard and branches, basic suffix/prefix stripping on words and so on) with the exception of the Bing queries.

It is easy to observe that the procedure is linear both in the size of the adjective list $\mathcal{L}$ and in the complexity of querying Bing search engine. With a little abuse of notation one denotes as C_B the time needed to search for an adjective a. An upper complexity bound for the acquisition phase is then $O(n \cdot \mathsf{C}_B)$.

One focus now on the sifting phase. It is quite easy to observe that the subroutine is quadratic in n, i.e. the upper bound for the sifting phase is then $O(n^2)$. It is mandatory to say that this bound complains the very worst case, where the external while is performed n time.

Finally, here some statistics about MORE execution time. Tests have been performed implementing MORE in Java (without a multi-processor/multi-thread support) on a desktop PC equipped with Intel Core i7-3630QM CPU with 4 logical cores (each with a frequency of 2.40GHz) and running Fedora 22 64-bit operation system. In Tables 2 and 3 one reports the execution time (in milliseconds), for the two runs of the sifting phase.

4. Related Work

Statistical Natural Language Processing is a large interest research field. In the last twenty years a number of challenging tasks have been addressed, e.g: part-of-speech tagging [17], text classification [23], text normalization[30, 32], terminology generation [31], document classification [9] and dictionary feature construction from scratch. MORE advances this last theme, that has been actively investigated in the recent past, as in, for instance [24]. There is a long stream of investigations on topics related to understanding the nature of lexical resources, starting from the pioneering investigation [20], until [19]. Focusing on the state of art specifically devoted to adjective detection and classification, some interesting research have been proposed both for linguistic classification and as tool for sentiment analysis[7]. This is the area of adjective-related studies that mostly concerns with vagueness and meaning negotiation, as showed in [3, 4, 10, 5, 11].

In [25] an automatic technique for semantic class detection of adjective is defined studied. In [26], the authors define and apply to English adjectives in WordNet a "supersense" taxonomy, using human annotation and supervised classification, releasing a well-behaving automatic classifiers and a labeled corpus of adjectives. In [16] a semi-supervised machine-learning approach is proposed for the classification of adjectives into property- vs. relation denoting, a highly relevant goal for ontology learning. The feasibility of the classification methodology is evaluated in a human annotation experiment. Some of the authors also dedicated efforts onto semantic aspects of text processing [14, 13, 1].

Adjective classification is also investigated in [18].From the sentiment analysis perspective, adjectives play a central role in a number of papers, e.g. in [8], where verbs and adjectives are exploited to automatically classify blog sentiment and in [29], where author introduce the POAE algorithm, an efficient procedure for the collection of polarity adjectives.

Since information retrieval by web search plays a central role in MORE acquisition phase, the work presented in this paper is also related to [27], where the simple unsupervised learning algorithm PMI-IR for recognizing synonyms is proposed. PMI-IR acquires statistical data by querying a Web search engine and, similarly as for MORE, is empirically evaluated using a set of test questions for students of English as a foreign or second Language.

5. Conclusions

In this paper, a novel algorithm for the adjective automatic classification called MORE is introduced. In particular, MORE is able to detect comparability and it has ben designed for english adjectives processing. Differently from other proposals, MORE is feasible and completely unsupervised. The classifier has been applied to a significant amount of adjectives, measuring an efficient execution time.

MORE dependency on the tested language is limited to the usage of two instruments: the support lists, and the rules for syntactice provision of comparatives and superlatives. The support lists are not indispensable, although they guarantee a significant speeding up of the method. A preliminary analysis (german, italian and french) showed that similar lists are easily found in languages other than english, and similarly do the syntactic rules governing the morphology of adjectives.

Experimental results are verified against a manual gold standard, and one measured an accuracy of 76%. That initial results are hopely encouraging, since both MORE and the result evaluation can be refined.

The methodology introduced in this paper is probably robust w.r.t. language changes, since it is not based on specific dictionary or knowledge basis. The adamptation of MORE to languages different from English is a work in progress. Moreover, one plans to lift the methods to other classification task, focusing on different lexical classes. A natural application setting of our investigation, as a medium time goal, will be sentiment analisys, focusing in particular on social media.

References

[1] S.A. Abdulhak, W. Riviera, N. Zeni, M. Cristani, R. Ferrario, and M. Cristani. Semantic-analysis object recognition: Automatic training set generation using textual tags. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8926:309–322, 2015.

[2] G. Boleda, S. Schulte Im Walde, and T. Badia. An analysis of human judgements on semantic classification of catalan adjectives. *Research on Language and Computation*, 6(3-4):247–271, 2008.

[3] E. Burato and M. Cristani. Learning as meaning negotiation: A model based on english auction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5559 LNAI:60–69, 2009.

[4] E. Burato and M. Cristani. The process of reaching agreement in meaning negotiation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7270 LNCS:1–42, 2012.

[5] E. Burato, M. Cristani, and L. Viganò. A deduction system for meaning negotiation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6619 LNAI:78–95, 2011.

[6] J. Butcher, C. Drake, and M. Leach. *Copy-Editing: The Cambridge Handbook for Editors, Authors and Publishers*. Cambridge University Press, 2 edition, 2006.

[7] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44:65–77, 2018.

[8] P. Chesley, B. Vincent, L. Xu, and R. Srihari. Using verbs and adjectives to automatically classify blog sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 27–29, 2006.

[9] M. Cristani, A. Bertolaso, Scannapieco S., and C. Tomazzoli. Future paradigms of automated processing of business documents. *International Journal of Information Management*, 40:67 – 75, 2018.

[10] M. Cristani and E. Burato. Approximate solutions of moral dilemmas in multiple agent system. *Knowledge and Information Systems*, 18(2):157–181, 2009.

[11] M. Cristani and E. Burato. A complete classification of ethical attitudes in multiple agent systems. volume 2, pages 1122–1123. IFAAMAS, 2009.

[12] M. Cristani, M. Cristani, A. Pesarin, C. Tomazzoli, and M. Zorzi. Making sentiment analysis algorithms scalable. In *Proceedings of the 4th International Workshop on Knowledge Discovery on the WEB, KDWeb 2018, Cáceres, Spain, June 5, 2018*. CEUR-WS.org, 2016.

[13] M. Cristani, D. Fogoroasi, and C. Tomazzoli. Measuring homophily. volume 1748 of *CEUR Workshop Proceedings 1748*. CEUR-WS.org, 2016.

[14] M. Cristani, C. Tomazzoli, and F. Olivieri. Semantic social network analysis foresees message flows. volume 1, pages 296–303, 2016.

[15] M. Dye, P. Milin, R. Futrell, and M. Ramscar. Alternative solutions to a language design problem: The role of adjectives and gender marking in efficient communication. *Topics in Cognitive Science*, 10(1):209–224, 2018.

[16] M. Hartung and A. Frank. A semi-supervised type-based classification of adjectives: Distinguishing properties and relations. In Nicoletta Calzolari, editor, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[17] K. Hengeveld, J. Rijkhoff, and A. Siewierska. Parts-of-speech systems and word order. *Journal of Linguistics*, 40(3):527–570, 2004.

[18] F. Hristea and M. Popescu. Adjective sense disambiguation at the border between unsupervised and knowledge-based techniques. *Fundamenta Informaticae*, 91(3-4):547–562, 2009.

[19] C. Kennedy. Vagueness and grammar: the semantics of relative and absolute gradable adjectives. *Linguistics and Philosophy*, 30(1):1–45, Feb 2007.

[20] E. Klein. A semantics for positive and comparative adjectives. *Linguistics and Philosophy*, 4(1):1–45, Mar 1980.

[21] C. D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[22] M Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[23] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March 2002.

[24] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June 2011.

[25] G. B. Torrent. Automatic acquisition of semantic classes for adjectives, ph.d thesis, automatic acquisition of semantic classes for adjectives ph. d. thesis, university of barcelona.

[26] Y. Tsvetkov, N. Schneider, D. Hovy, A. Bhatia, M. Faruqui, and C. Dyer. Augmenting english adjective senses with supersenses. In Nicoletta Calzolari, editor, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[27] P. D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK, 2001. Springer-Verlag.

[28] S. Walter, C. Unger, and P. Cimiano. Automatic acquisition of adjective lexicalizations of restriction classes: a machine learning approach. *Journal on Data Semantics*, 6(3):113–123, 2017.

[29] H. Wang and J. Ma. Automatic collection polarity adjectives in sentiment analysis. *Journal of Computational Information Systems*, 8(16):6569–6577, 2012.

[30] M. Zorzi, C. Combi, R. Lora, M. Pagliarini, and U. Moretti. Automagically encoding adverse drug reactions in meddra. In *Proceedings of 2015 International Conference on Healthcare Informatics, ICHI 2015, Dallas, TX, USA, October 21-23, 2015*, pages 90–99, 2015.

[31] M. Zorzi, C. Combi, G. Pozzani, E. Arzenton, and U. Moretti. A co-occurrence based meddra terminology generation: Some preliminary results. In *Proceedings of Artificial Intelligence in Medicine - 16th Conference on Artificial Intelligence in Medicine, AIME 2017, Vienna, Austria, June 21-24, 2017, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10259 LNAI,*, pages 215–220, 2017.

[32] M. Zorzi, C. Combi, G. Pozzani, and U. Moretti. Mapping free text into meddra by natural language processing: A modular approach in designing and evaluating software extensions. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB 2017, Boston, MA, USA, August 20-23, 2017*, pages 27–35. Association for Computing Machinery, Inc, 2017.