Sentimatrix – Multilingual Sentiment Analysis Service

Alexandru-Lucian Gînscă[1], Emanuela Boroș[1], Adrian Iftene[1], Diana Trandabăț[1], Mihai Toader[2], Marius Corîci[2], Cenel-Augusto Perez[1], Dan Cristea[1, 3]
[1]"Al. I. Cuza" University, Faculty of Computer Science, Iasi, Romania
[2]Intelligentics, Cluj-Napoca, Romania
[3]Institute of Computer Science, Romanian Academy, Iasi, Romania
{lucian.ginsca, emanuela.boros, adiftene, dtrandabat, augusto.perez, dcristea}@info.uaic.ro, {mtoader, marius}@intelligentics.ro

Abstract

This paper describes the preliminary results of a system for extracting sentiments opinioned with regard with named entities. It also combines rule-based classification, statistics and machine learning in a new method. The accuracy and speed of extraction and classification are crucial. The service oriented architecture permits the end-user to work with a flexible interface in order to produce applications that range from aggregating consumer feedback on commercial products to measuring public opinion on political issues from blog and forums. The experiment has two versions available for testing, one with concrete extraction results and sentiment calculus and the other with internal metrics validation results.

1 Motivation

Nowadays, big companies and organizations spend time and money in order to find users' opinions about their products, the impact of their marketing decisions, or the overall feeling about their support and maintenance services. This analysis helps in the process of establishing new trends and policies and determines in which areas investments must be made. One of the focuses of our work is helping companies build such analysis in the context of users' sentiment identification. Therefore, the corpus we work on consists of articles of newspapers, blogs, various entries of forums, and posts in social networks.

Sentiment analysis, i.e. the analysis and classification of the opinion expressed by a text on its subject matter, is a form of information extraction from text, which recently focused a lot of research and growing commercial interest.

This paper describes Sentimatrix, a sentiment analysis service, doing sentiment extraction and associating these analyses with named entities, in different languages. We seek to explore how sentiment analysis methods perform across languages, especially Romanian. The main applications that this system experiments with are monitoring the Internet before, during and after a campaign/message release and obtaining consumer feedback on different topics/products.

In Section 2 we briefly discuss a state of the art in sentiment analysis, the system's architecture is described in Section 3 and in Section 4 we focus on identifying opinions on Romanian. Subsequently, we present the experiment results, analysis and discussion in Sections 5 and 6. Future work and conclusions are briefly described in Section 7.

2 Sentimatrix compared with state-of-the-art

A comprehensive state of the art in the field of sentiment analysis, together with potential applications of such opinion identification tools, is presented in (Pang and Lee, 2008).

Starting from the early 1990s, the research on sentiment-analysis and point of views generally assumed the existence of sub-systems for rather sophisticated NLP tasks, ranging from parsing to the resolution of pragmatic ambiguities (Hearst, 1992; Wiebe 1990 and 1994). In Sentimatrix, in order to identify the sentiment a user expresses about a specific product or company, the company name must be first identified in the text. Named

Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011, pages 189–195,
24 June, 2011, Portland, Oregon, USA

entity recognition (NER) systems typically use linguistic grammar-based techniques or statistical models (an overview is presented in (Nadeau and Satoshi Sekine. 2007)). Hand-crafted grammar-based systems typically obtain better precision, but at the cost of lower recall and months of work by experienced computational linguists. Besides, the task is hard to adapt to new domains. Various sentiment types and levels have been considered, starting from the "universal" six level of emotions considered in (Ovesdotter Alm, 2005; Liu et al., 2003; Subasic and Huettner, 2001): anger, disgust, fear, happiness, sadness, and surprise. For Sentimatrix, we adapted this approach to five levels of sentiments: strong positive, positive, neutral, negative and strong negative.

The first known systems relied on relatively shallow analysis based on manually built discriminative word lexicons (Tong 2001), used to classify a text unit by trigger terms or phrases contained in a lexicon. The lack of sufficient amounts of sentiment annotated corpora led the researchers to incorporate learning components into their sentiment analysis tools, usually supervised classification modules, (e.g., categorization according to affect), as initiated in (Wiebe and Bruce 1995).

Much of the literature on sentiment analysis has focused on text written in English. Sentimatrix is designed to be, as much as possible, language independent, the resources used being easily adaptable for any language.

Some of the most known tools available nowadays for NER and Opinion Mining are: Clarabridge (www.clarabridge.com), RavenPack (ravenpack.com), Lexalytics (www.lexalytics.com) OpenAmplify (openamplify.com), Radian6 (www.radian6.com), Limbix (lymbix.com), but companies like Google, Microsoft, Oracle, SAS, are also deeply involved in this task.

3 System components

In Figure 1, the architecture and the main modules of our system are presented: preprocessing, named entity extraction and opinion identification (sentiment extraction per fragment).

The final production system is based on service oriented architecture in order to allow users flexible customization and to enable an easier way for marketing technology. Each module of the system (Segmenter, Tokenizer, Language Detector, Entity Extractor, and Sentiment Extractor) can be exposed in a user-friendly interface.

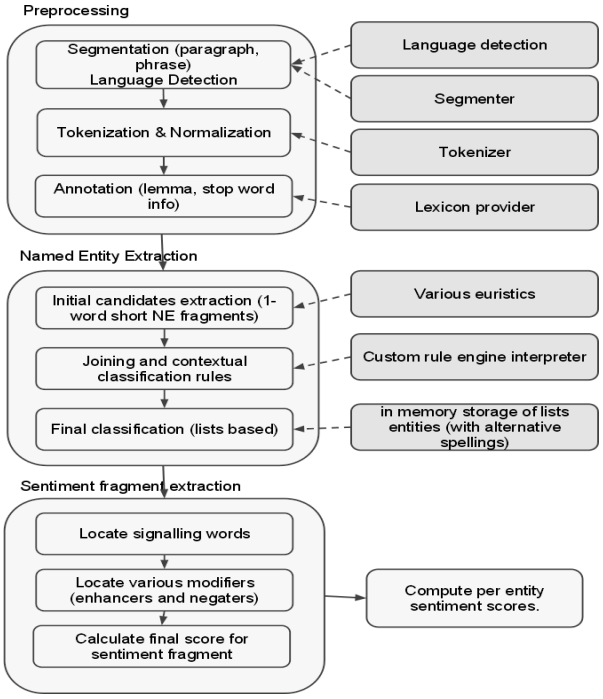

Figure 1. System architecture

3.1 Preprocessing

The preprocessing phase is made out of a text segmentator and a tokenizer. Given a text, we divide it into paragraphs, every paragraph is split into sentences, and every phrase is tokenized. Each token is annotated with two pieces of information: its lemma (for Romanian it is obtained from our resource with 76,760 word lemmas corresponding to 633,444 derived forms) and the normalized form (translated into the proper diacritics[1]).

3.2 Language Detection

Language detection is a preprocessing step problem of classifying a sample of characters based on its features (language-specific models). Currently, the system supports English, Romanian and Romanian without Diacritics. This step is needed in order to correctly identify a sentiment or a sentiment modifier, as the named entity detection depends on this. We combined three methods for

[1] In Romanian online texts, two diacritics are commonly used, but only one is accepted by the official grammar.

identifying the language: *N-grams detection*, strictly *3-grams detection* and *lemma correction*.

The 3-grams classification method uses corpus from Apache Tika for several languages. The Romanian 3-gram profile for this method was developed from scratch, using our articles archive. The language detection in this case performs simple distance measurement between every language profile that we have and the test document profile. The N-grams classification method implies, along with computing frequencies, a posterior Naive Bayes implementation. The third method solves the problematic issue of short phrases language detection and it implies looking through the lemmas of several words to obtain the specificity of the test document.

3.3 Named Entity Recognition

The Named Entity Recognition component for Romanian language is created using linguistic grammar-based techniques and a set of resources. Our component is based on two modules, the named entity identification module and the named entity classification module. After the named entity candidates are marked for each input text, each candidate is classified into one of the considered categories, such as Person, Organization, Place, Country, etc.

Named Entity Extraction: After the pre-processing step, every token written with a capital letter is considered to be a named entity candidate. For tokens with capital letters which are the first tokens in phrases, we consider two situations:

1. *this first token of a phrase is in our stop word list* (in this case we eliminate it from the named entities candidate list),
2. *the first token of a phrase is in our common word list*. In the second situation there are considered two cases:
   a. *this common word is followed by lowercase words* (then we check if the common word can be found in the list of trigger words, like *university*, *city*, *doctor*, etc.),
   b. *this common word is followed by uppercase words* (in this case the first word of the sentence is kept in the NEs candidate list, and in a further step it will be decided if it will be combined with the following word in order to create a composed named entity).

Named Entities Classification: In the classification process we use some of rules utilized in the unification of NEs candidates along with the resource of NEs and several rules specifically tailored for classification. Thus, after all NEs in the input text are identified and, if possible, compound NEs have been created, we apply the following classification rules: *contextual rules* (using contextual information, we are able to classify candidate NEs in one of the categories Organization, Company, Person, City and Country by considering a mix between regular expressions and trigger words) and *resource-based rules* (if no triggers were found to indicate what type of entity we have, we start searching our databases for the candidate entity).

Evaluation: The system's Upper Bound and its performance in real context are evaluated for each of the two modules (identification and classification) and for each named entity type. The first part of the evaluation shows an upper bound of 95.76% for F-measure at named entity extraction and 95.71% for named entity classification. In real context the evaluation shows a value of 90.72% for F-measure at named entity extraction and a value of 66.73% for named entity classification. The results are very promising, and they are being comparable with the existing systems for Romanian, and even better for Person recognition.

4 Identify users opinions on Romanian

4.1 Resources

In such a task as sentiment identification, linguistic resources play a very important role. The core resource is a manually built list of words and groups of words that semantically signal a positive or a negative sentiment. From now on, we will refer to such a word or group of words as "sentiment trigger". Certain weights have been assigned to these words after multiple revisions. The weights vary from -3, meaning strong negative to +3, which translates to a strong positive. There are a total of 3,741 sentiment triggers distributed to weight groups as can be observed in Figure 2. The triggers are lemmas, so the real number of words that can be identified as having a sentiment value is much higher.

This list is not closed and it suffers modifications, especially by adding new triggers, but in certain cases, if a bad behavior is observed, the weights may also be altered.

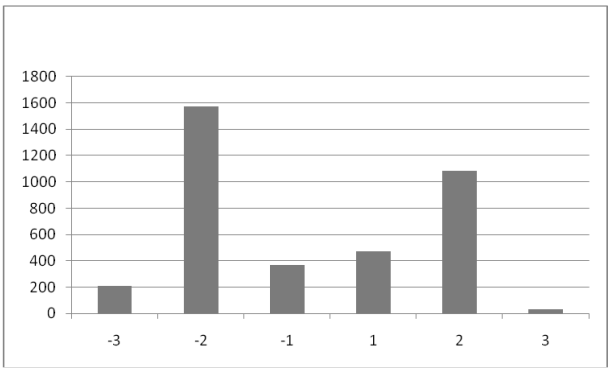

Figure 2. Number of sentiment words by weight groups

We define a modifier as a word or a group of words that can increase or diminish the intensity of a sentiment trigger. We have a manually built list of modifiers. We consider negation words a special case of modifiers that usually have a greater impact on sentiment triggers. So, we also built a small list of negation words.

4.2 Formalism

General definitions: We define a sentiment segment as follows:

$$s_{SG} = ([negation], [modifier], sentimentTrigger)$$

s_{SG} is a tuple in which the first two elements are optional.

Let N_L be the set of negation words that we use, M_L the set of modifiers and T_L the set of sentiment triggers. We define two partially ordered sets:

$$P_+ = (S_+, \leq_+), \quad where\ S_+ \subseteq N_L\ X\ M_L\ X\ T_L \text{ and}$$
$$P_- = (S_-, \leq_-), \quad where\ S_- \subseteq N_L\ X\ M_L\ X\ T_L$$

We consider $\leq_+$ and $\leq_-$ are two binary relations that order sentiment segments based on their weights. The weights give a numeric representation of how strong or weak is the sentiment expressed by the sentiment segment. For instance, if we have s_{SG1}, s_{SG2}, s_{SG3} with the weights 1, 2, 3 and s_{SG4}, s_{SG5}, s_{SG6} with the weights 4, 5, 6, then $s_{SG1} \leq_+ s_{SG2} \leq_+ s_{SG3}$ and $s_{SG4} \leq_- s_{SG5} \leq_- s_{SG6}$.

We define a weight function, *weightS:* $S \rightarrow R$, over the set of sentiment segments that returns a real number representing the global weight that takes into consideration the effect of the negation words and modifiers on the sentiment trigger.

Global sentiment computation: In this section, we will describe how the cumulative value of a sentiment segment, expressed by the *weightS,* is computed.

At the base of a sentiment segment stands the given weight of the sentiment trigger that is part of the general segment. Besides that, modifiers and negation words have a big impact. For example, consider the following three sentences.

1. *John is a good person.*
2. *John is a very good person.*
3. *John is the best.*

In the first one, a positive sentiment is expressed towards *John*. In the second one, we also have a positive sentiment, but it has a bigger power and in the third one the sentiment has the strongest intensity.

We distinguish two separate cases in which negation appears. The first one is when the negation word is associated with a sentiment trigger and it changes a positive one into a negative trigger and vice versa; and the second one refers to the case in which the negation affects a trigger accompanied by a modifier. We illustrate these situations in the following examples.

A1. *John is a good person.*
A2. *John is not a good person.*
B1. *John is the best.*
B2. *John is not the best.*

If we assign the weight +2 to *good* in the A1 sentence, it is safe to say that in A2, *not good* will have the weight -2. From a semantic perspective, we have the antonym relation*: good* $\neq$ ⅂ *good* and the synonym relation ⅂ *good* = *bad).*

On the other hand, in the B2 example, *not the best* is not the same as *the worst*, the antonym of *the best*. In this case, we consider *not the best* to be somewhere between *good* and *the best*. We give a more detailed description of this kind of ordering in the formalisms section.

Entity sentiment computation: Let *E* denote a named entity and *Sent* a sentence. We define the sentiment value, *sv,* of an entity *E* in a sentence

Sent as the general sentiment expressed towards *E* in *Sent*. This value is a real number and is the cumulative effect of all the sentiment segment's weights in that sentence.

Let S_{Sent} be the set of all sentiment segments in the sentence *Sent* and *distance(E, s_{SG})* the number of tokens between *E* and s_{SG}. The expression for computing the sentiment value of an entity in a sentence is given below:

$$\mathrm{sv(E, Sent)} = \frac{\sum_{\mathrm{s_{SG}} \in \mathrm{S_{Sent}}} \frac{\mathrm{weightS(s_{SG})}}{\ln[1 + \mathrm{distance(E, s_{SG})}]}}{|\mathrm{S_{Sent}}|}$$

The *sv* for an entity *E* in a larger text will be the sum of the sentiment values for *E* in every sentence of the text.

4.3 Evaluation

For testing our system, we were interested in two aspects: how well does it recognize sentiment segments and how accurate is the semantic meaning given by the system compared to the one attributed by a person. More than that, we dissected the sentiment segment and analyzed the system's performance on finding sentiment triggers and modifiers.

Evaluation resources: Finding or developing clean resources is the most difficult part of the evaluation task. We used 100 complex sentences selected from news articles that were manually annotated as a gold standard. Despite the small number of sentences, they were specially thought to capture a large number of situations.

Evaluation methods: We used precision, a widely known information retrieval metric and other measures that we developed for this task, such as a relaxed precision and deviation mean. We provide below a more detailed description of these metrics.

We computed the precision for sentiment segments, sentiment triggers and modifiers as follows:

$$P_{entity=} \frac{\#\ correct\ found\ entities}{\#\ total\ found\ entities},$$
$$where\ entity\ \in \{\ sentiment\ segment,\ sentiment\ trigger, modifier\ \}$$

For the weight associated with the sentiment segment, we use two types of precision: an exact match precision, P_{weight} in which we considered a found weight to be correct if it is equal to the weight given in the gold corpus and a relaxed precision, RP_{weight}. We computed these metrics only on the correctly identified segments. Let CS be the set of correctly identified segments, w_F the weight of the sentiment segment returned by our system and w_G the weight of the sentiment segment from the gold corpus.

$$RP_{weight} = \frac{\sum_{s_{SG} \in CS} partialMatch(s_{SG})}{|CS|},$$

$$where\ partialMatch(s_{SG}) = \begin{cases} 1, & |w_F - w_G| < 1.5 \\ 0, & otherwise \end{cases}$$

The RP_{weight} measure is important because the weights given to a sentiment segment can differ from one person to another and, by using this metric, we allow our system to make small mistakes.

Besides the sentiment segments, we also tested the sentiment values of entities. For this task, we used four metrics. The first one is a relaxed precision measure for the sentiment values computed for the entities. Let S_{SV} be the set of the sentiment values returned by the system, sv_F the sentiment value found by the system and sv_G the sentiment value specified in the gold corpus.

$$RP_{sv} = \frac{\sum_{sv_F \in S_{sv}} partialMatch(sv_F)}{|S_{sv}|},$$

$$where\ partialMatch(sv_F) = \begin{cases} 1, & |sv_F - sv_G| \leq 0.5 \\ 0, & otherwise \end{cases}$$

The last three metrics address the problem of how far the sentiment values are returned by the system from those considered correct by a human annotator. We called these measures *sv positive deviation,* D_{sv+}, which takes into account only positive sentiment values, *sv negative deviation,* D_{sv-}, which takes into account only negative sentiment values and *sv general deviation,* D_{sv+-}, an average of the first two.

$$D_{sv+} = \frac{\sum_{sv_F \in S_{sv+}} |sv_F - sv_G|}{|S_{sv+}|}$$

S_{SV+} is the set of positive sentiment values found by the system. D_{sv-} is calculated in a similar manner as D_{sv+}.

5 Results

The results were obtained using the manually annotated sentences presented in the Evaluation resources section. Out of those sentences, 58% contain entities and 42% contain only sentiment segments. The entity-related metrics could be applied only on the first type of sentences. The results can be observed in Figure 3.

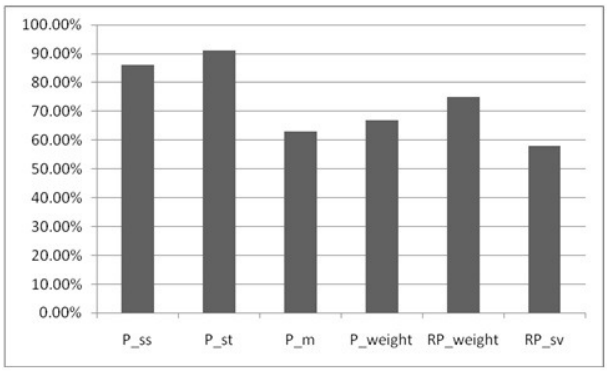

Figure 3. Precision metrics results

In Figure 3, $P\_ss = P_{sentiment\ segment}$, $P\_st = P_{sentiment\ trigger}$, $P\_m = P_{modifier}$ and the rest of the metrics have the same meaning as defined in the evaluation methods section.

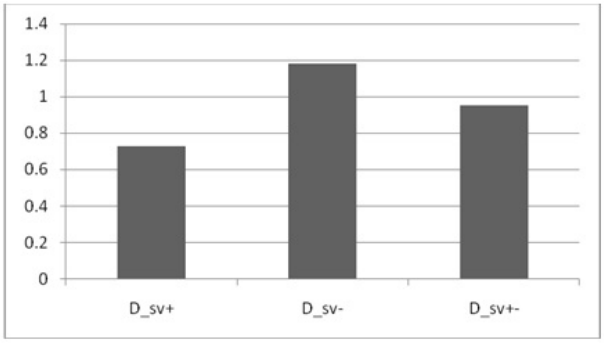

Figure 4. Deviation metrics results

In Figure 4, we show the results of the metrics that follow the sentiment value deviation.

6 Discussion

The main problem encountered is the contexts in which the opinions that we identify appear. It is possible that the same trigger has a positive meaning in a context, and in another context to be negative. For example, “scade TVA” (En: “reduce VAT”) which is positive, compared to "scad salariile” (En: “reduce salaries”) which is negative. In these cases the trigger “scade” (En: reduce) can lead to opposing opinions. As for “inchide fabrica” (En: “close the plant”), that has a negative context compared to “inchide infractorul” (En: “close the offender”) which is positive.

Another problem in quantifying the sentiments and opinions is related to numerical values that we identify in the text. For example “15 profesori protesteaza” (En: “15 teachers protest”) compared to “2.000.000 de profesori protesteaza” (En: “2,000,000 teachers protest”). In both cases we have negative sentiments, but it is clear that the second case has even a stronger sense due to the large number of people who participate in the protest. If in the first case it seems to be a local issue, at the school, in the second case, it seems to be a general problem that is seen nationwide.

7 Conclusion and Future Work

This paper introduces the Sentimatrix system. The main components of the system are dedicated to identifying named entities, opinions and sentiments. Preliminary evaluation show promising results.

Future work includes completing the resources lists with entities, sentiment triggers and modifiers. As we have seen in the tests, rapid improvements can be achieved by taking into consideration modifiers such as “daca”, “posibil”, “ar putea” (En: “if”, “possible”, “could”) which have the effect of lowering the intensity of opinions and sentiments. Also, we intend to build a bigger gold corpus to evaluate sentiments by using a semi-automatic approach (at first the system generates annotation, which is later to be validates and completed by a human annotator).

Acknowledgments

The research presented in this paper is partially funded by the Sectoral Operational Program for Human Resources Development through the project “Development of the innovation capacity and increasing of the research impact through post-doctoral programs” POSDRU/89/1.5/S/49944.

References

Bo Pang, Lillian Lee. 2008. *Opinion Mining and Sentiment Analysis*, Found. Trends Inf. Retr., Vol. 2, No. 1–2. (January 2008), pp. 1-135

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

David Nadeau and Satoshi Sekine. 2007. *A survey of named entity recognition and classification*, Linguisticae Investigationes 30, no. 1, 3{26, Publisher: John Benjamin's Publishing Company

David Nadeau. 2007. *Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision*, PhD Thesis.

Ellen Riloff, Janyce Wiebe, and William Phillips. 2005. Exploiting subjectivity classification to improve information extraction. In *Proceedings of AAAI*, pages 1106–1111.

Hugo Liu, Henry Lieberman, and Ted Selker. 2003.A model of textual affect sensing using real-world knowledge. In *Proceedings of Intelligent User Interfaces (IUI)*, pages 125–132.

Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. 2006. Overview of the TREC-2006 Blog Track. In *Proceedings of the 15th Text REtrieval Conference (TREC 2006)*.

Janyce M. Wiebe 1990. Identifying subjective characters in narrative. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 401–408.

Janyce M. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Janyce Wiebe and Rebecca Bruce. 1995. Probabilistic classifiers for tracking point of view. In *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 181–187.

Marti Hearst. 1992. Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, *Text-Based Intelligent Systems*, pages 257–274. Lawrence Erlbaum Associates.

Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

Pero Subasic and Alison Huettner. 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*, 9(4):483–496.

Richard M. Tong. 2001. An operational system for detecting and tracking opinions in on-line discussion. In *Proceedings of the Workshop on Operational Text Classification (OTC)*.

Scurtu V., Stepanov E., Mehdad, Y. 2009. *Italian named entity recognizer participation in NER task@evalita 09*, 2009.