Offensive Language Detection Using Multi-level Classification

Amir H. Razavi[1], Diana Inkpen[1], Sasha Uritsky[2], and Stan Matwin[1,3]

[1] School of Information Technology and Engineering (SITE), University of Ottawa, Ottawa, ON, Canada, K1N 6N5
[2] Natural Semantic Modules co. 5 Tangreen Court, Suite 510 Toronto, ON, M2M 4A7
[3]Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

{araza082, diana, stan}@site.uottawa.ca, sasha@nsemodules.com

Abstract. Text messaging through the Internet or cellular phones has become a major medium of personal and commercial communication. In the same time, flames (such as rants, taunts, and squalid phrases) are offensive/abusive phrases which might attack or offend the users for a variety of reasons. An automatic discriminative software with a sensitivity parameter for flame or abusive language detection would be a useful tool. Although a human could recognize these sorts of useless annoying texts among the useful ones, it is not an easy task for computer programs. In this paper, we describe an automatic flame detection method which extracts features at different conceptual levels and applies multi-level classification for flame detection. While the system is taking advantage of a variety of statistical models and rule-based patterns, there is an auxiliary weighted pattern repository which improves accuracy by matching the text to its graded entries.

Keywords: Flame Detection; Filtering; Information Extraction; Information Retrieval; Multi-level Classification; Offensive Language Detection.

1 Introduction

Recently, pattern recognition and machine learning algorithms are being used in a variety of Natural Language Processing applications. Everyday we have to deal with texts (emails or different types of messages) in which there are a variety of attacks and abusive phrases. An automatic intelligent software for detecting flames or other abusive language would be useful and could save its users time and energy.

Offensive phrases could mocks or insult somebody or a group of people (attacks such as aggression against some culture, subgroup of the society, race or ideology in a tirade). Here are several types of offensive language in this category:

Taunts: These phrases try to condemn or ridicule the reader in general.

References to handicaps: These phrases attack the reader using his\her shortcomings (i.e., “IQ challenged”).

Squalid language: These phrases target sexual fetishes or physical filth of the reader.

Slurs: These phrases try to attack a culture or ethnicity in some way.

Homophobia: These phrases are usually talking about homosexual sentiments.

Racism: These phrases intimidate race or ethnicity of individuals [10].

Extremism: These phrases target some religion or ideologies.

There are also some other kinds of flames, in which the flamer abuses or embarrasses the reader (not an attack) using some unusual words/phrases like:

Crude language: expressions that embarrass people, mostly because it refers to sexual matters or excrement.

Disguise: expressions for which the meaning or pronunciation is the same as another more offensive term.

Four-letter words: there are five or six words which consist of only four letters.

Provocative language: expressions that may cause anger or violence.

Taboos: expressions which are forbidden in a certain society/community. There are lots of expressions that are forbidden because of what they refer to, not necessarily there is some particular taboo words used in the expression.

Unrefined language: some expressions that lack polite manners and the speaker is harsh and rude [12].

Based on the above definitions, when we say flame detection, implicitly we are talking about every context that falls into one or more of the defined cases.

Sometime, internet users searching or browsing in some specific sites are frustrated as they encounter offensive, insulting or abusive messages. It occasionally happens even in frequently-used websites like Wikipedia.

Therefore an automatic system for discriminating between regular texts and flames would save time and energy during our browsing on the web or in our everyday emails or text messages. At this stage, when we take a look at the literature on attempts to discriminate between acceptable contexts and the flames, we observe considerable percentage of disagreement between human expert annotators having the same definition of flames [1,2,3]. Therefore, it becomes evident that we cannot provide a rigid product for flame detection for all purposes. Hence in this paper we will define a tolerance margin for abusive language, based on certain conditions or applications (different sites and usages), so that the user could have an acceptable interaction with the computer.

The literature on offensive language detection and specifically on natural language analysis describes flames as exhibiting extreme subjectivity [3], depending on the context. These kinds of subjectivity are either speculative or evaluative [2]. Speculative expressions include any doubtful phrases, whereas for evaluative expressions we are dealing with emotions (such as hate, anger), judgments or opinions [9]. Therefore, any sign of extremity in such subjectivities could be considered as an effective feature for evaluation and possibly, flame detection.

However, computer software does not have the ability of capturing the exact concept of a flame context; yet, there are some useful features that we could point out, such as:

- The frequency of phrases which fall into one of the graded (weighted) flaming patterns (for each grade/weight separately);

- The frequency of graded/weighted words or phrases with abusive/extremist load, in each grade;

- The highest grade (maximum weight) which occurs in a context;

- The normalized average of the graded/weighted words or phrases.

These highlights led us to design and implement a fuzzy gauge of flame detection, and implement it in a software that could be modified regarding the acceptable tolerance margin, based on training data, manual adjustment, or even instant labeled contexts.

In section 2 of this paper we introduce some related works in this area, then we describe the flame-annotated data (section 3), the system features (sections 4), the methodology (section 5), the results (section 6), discussion (section 7), and conclusion and future work (section 8).

2 Related Work

Although there are few papers on computerized flame detection methods (which we review in this section), recently many researchers in Artificial Intelligence and Natural Language Processing have been working on different kinds of opinion extraction or sentiment analysis, e.g., Pang et al. [15], Turney and Littman [16], Gordon et al. [17], Yu and Hatzivassiloglou [18], Riloff and Wiebe [19], Yi et al. [20], Dave et al. [21], Riloff et al. [22] and Razavi and Matwin [23, 24]. In many cases detecting the level of intensity of moods or attitudes (Negative/Positive) could be an effective attribute of some specific opinion exploration for offensive language detection. Furthermore, subjective language recognition could also be useful in flame detection [1,9]. Hence, the subjective language detection is a task for which flame detection could be considered an offspring. In this area, we mention the work of Wiebe and her group: after tagging the contexts (as subjective or non subjective) using three expert judges, they applied machine learning algorithms for classifying texts based on some of their constituent words and expressions [13, 14]. This study led to similar, but more sophisticated work on evaluative and speculative language extraction [9]. Systematic subjectivity detection could be helpful in flame recognition or email classification as well [3, 5]

Swearing as a class of offensive language has been studied by Thelwall [25] which is mostly focused on the distribution by age considering their genders.

In addition to parts of speech, a corpus can be annotated with demographic features such as age, gender and social class, and textual features such as register, publication medium and domain. However some abusive languages may be related to religion (e.g. “Jesus”, “heaven”, “hell” and “damn”), sex (e.g. “fuck”), racism (e.g.

"nigger"), defecation (e.g. "shit"), homophobia (e.g. "queer") and other matters; [26, 27] try to examine only the pattern of uses of "fuck" and its morphological variants, because this is a typical swear-word that occurs frequently in the British National Corpus (BNC). Also McEnery et. Al. in this article try to build and expand upon the examination of "fuck" [28, 29] by examining the distribution pattern of "fuck" within and across spoken and written registers.

Specifically as flame detection systems, we should name *Smokey* [1] which probably is still being used by Microsoft in commercial applications. Smokey not only considers the insulting or abusing words, but also tries to recognize some structure of patterns through the flames. Smokey is equipped with a parser for syntactic analysis, which is a preliminary step for going through a semantic rule-base analysis process. Eventually, Smokey applies a C4.5 decision tree classifier for recognizing each context as a flame or not. The system, at the time of publication, used 720 message as its training set and 460 messages as testing set, and achieved 64% true-positive rate for the *flame* labeled messages and 98% true-positive rate for the *okay* labeled messages.

As another method for flame and insult detection, we can name Dependency Structure analysis which tries to detect any extreme subjectivity in texts [8].

Unfortunately, no flame detection software is freely available for trail or research purposes; therefore we cannot directly compare our results to results of other systems on our dataset.

3 Flame Annotated Data

In this study, we consider a message as a flame if either the main intention is *attack* (as we described above) or it contains *abusive* or *hostile* words, phrases or language, considering the desired tolerance margin.

We used two different sources of messages. The first set of data was provided by the NSM (Natural Semantic Module) company log files. This group of data contains 372 sentences in which the company's users ask for some kind of information, services, or fun activities, in an interactive manner. An example of offensive statement is: "Do you have plans for this smelly meeting that is supposed to take place today?"

The second set of data that we used consists of 1288 Usenet newsgroup messages which were already annotated and used for flame recognition task by Martin *et al.* [2]. This dataset is balanced among the alt, sci, comp, and rec categories from the Usenet hierarchy. An example message, annotated as "flame", is: "Feudalist has a new name. How many is that now? Feudalist. Quonster. Backto1913. That's four with BacktoTheStoneAge. I have never met anyone this insecure before. Actually, I think that BacktoTheStoneAge is intended as a parody. If not, he vastly miscalculated, because I have been laughing hysterically at these posts." Another example, also a "flame" is: "Do you find joy pouncing on strangers I have never found her doing this. Eric, have you?". After deleting the messages longer than 2500 characters and two messages in French, we obtained with 1153 usable messages. The first dataset is

composed mostly small of sentences using abusive language, and the second one contains rather long sentences full of sarcasms and ironic phrases; therefore we decided to combine them together in order to see the performance over a generic and typical offensive language detection task, rather a specific category.

We used a total number of 1525 messages (1038 *(68%) Okay* and 487 *(32%)* Flame), from the two datasets together, from which 10% was used as a test set, and the rest was used as training set for our multi-level classifier.

4 Methodology

After data preprocessing[1], we run a three-level classification for flame detection. Considering the attributes of each level we tried most of the applicable machine learning algorithms implemented in Weka (the standard machine learning software developed at the University of Waikato) [11]. We considered factors like time efficiency and updatability for online applications that determined the choice of classifier used (e.g., for the first level we needed to use fast algorithms which could work with a large number of attributes in acceptable time). After determining which algorithms satisfy these requirements, we chose the one that achieved the highest level of performance among the varieties of simple and combined complex methods available in Weka. This process for classifier selection was applied for the other levels as well. The classifiers discussed in this paper provided the highest discriminative power, compared to the other classifiers that we tried. In the third level of classification we use our Insulting and Abusing Language Dictionary which contains some word, phrase, and expression patterns for corresponding pattern recognition.

4.1 Insulting and Abusing Language Dictionary

We have collected about 2700 words, phrases, and expressions, with different degrees of manifestation of flame varieties. All the entries of this dictionary have considerable load of either *abusing / insulting* impact or *extreme subjectivity* in some of the above listed categories. We initially assigned all the entries weights in the range of 1 to 5, based on the potential impact level of each entry on the classification of the containing context. The weights that accompany this data can be used for setting the tolerance margin on flame detection for different applications. Then, in several steps of adaptive learning (on training data), we performed modifications on the weights to

[1] In preprocessing, first all the different headers, internet addresses, email addresses and tags were filtered out. Then all the delimiters such as spaces, tabs or new line characters, in addition to the following characters: “\ \r : ( ) ` 1 2 3 4 5 6 7 8 9 0 \ ' , ; = \ [ ] ; / < > { } | ~ @ # $ \ % ^ & * _ + ” were removed from each message, whereas expressive characters (Punctuations) like: “ - . ‘ ’ ! ? ” were kept. Punctuations (including “ ”) could be useful for determining the scope of speaker’s messages. This step prevents the system from coming up with a lot of useless tokens as features for our first-level classifier.

address the task for a most generic purpose. (However the process of the adaptive leaning could be performed based on any targeted specific domain in the field of the flame detection.) We achieved stability for the weighs with the highest level of discrimination on flames/non-flames. The result is our Insulting or Abusive Language Dictionary (IALD), a fundamental resource for our system.

At the beginning, some of these phrases or expressions contained up to five words including some wild-cards like *Somebody* or *Something (i.e. "chew Somebody's ass out" Or "Ball Somebody or Something up")*. These entries are actually raw texts which in the next stage became patterns; they help the software to estimate the probability of being a flame for each context. At this level we make a pattern for each of the entries that match a variety of word sequences (Replacing Somebody or Something wild cards for the above example). In this way each pattern could be matched with any sequence of words in which we have a few (not more than three) tokens in place of wild cards. The patterns also could match series using different types of verbs (ending in *ing*, *ed*, *d*, *es*, *s*) or nouns (ending in *es*, *s*)[2]. Hence, the original patterns in the repository entries were generalized, achieving considerable flexibility; now they could match tens of thousands word sequences in everyday contexts.

At this level, after pattern matching for each message/sentence we could supply another resource for flame probability estimation for the main task, which is flame detection.

4.2 Multilevel Classification

As part of the machine learning core of our package, we run three-level classifications on training data, using the IAL Dictionary.

In the first level of classification, considering the high degree of feature sparsity, we use the Complement Naïve Bayes classifier [11] for selecting the most discriminative (~1700) features[3] as the new training feature space and pass them to the next level of classification. (The initial raw data resulted after tokenization contained 15636 features, after preliminary feature trimming, i.e., removal of stop-words and terms that occurred only once.)

In the second level, we chose the Multinomial Updatable Naïve Bayes classifier [11] in order to efficiently update its model (Model 2), based on new labeled sentences which could be added to the system after the initial training process in order to do adaptive learning. This classifier was run on the best feature space extracted from the previous level of classification. The outputs of this classification level are new aggregated features extracted from the previous level feature space, with the following attributes as the input for our last-level classification task, using IALD:

---

[2] In addition to matching the wild cards, any word, phrase or expression which has any special character (leading or tailing) in the message would be tested and matched with the corresponding IADL entry.

[3] We used Wrapper Supervised Feature Selection algorithm with "RankSearch" method as our search method in Weka [11].

- Frequency of IALD word/phrase/expression patterns which are matched in the current instance, in each weight level (five attributes);

- Maximum weight of IALD entries that have been matched in the current message;

- Normalized average weight of IALD entries which have been matched in the current message;

- The probability that the current instance is *Okay,* based on the previous level classification applying Model 2;

- The probability that the current instance is a *Flame,* based on the previous level classification applying Model 2;

- The prediction of the previous level classification on the current instance, applying Model 2 (*Okay or Flame*);

In the last level, we run a rule-based classifier named DTNB (Decision Table/Naive Bayes hybrid classifier [6]) on the output of the second level (the features described above and label assigned in the previous level), which makes the final decision upon the current instance (*Okay or Flame*).[4]

5 Results

After preprocessing and before performing the feature selection, we ran the Complement Naïve Bayes classifier on the whole feature space (15,636); applying 10-fold cross-validation on the above described data we got the results depicted in the first row of Tables 1 and 2.

At this level, the accuracy was about 16% better than the baseline. The baseline that we use for comparison always chooses the most frequent class (it reflects the class distribution) and has an accuracy of 68%. As shown in Table 1, there were 936 Okay texts classified correctly classified as Okay, and 349 Flames corrected classified as Flames. The others are classification errors: 102 Flames classified as Okay, and 138 Okay texts classified as Flames.

Since the 10-fold cross-validation works on features selected from the entire dataset, this is different from the operation of a deployed package where the test instances will not participate in the feature selection process. To evaluate the performance in such more realistic situation, we have trained separately, then tested on a held-out (10%) randomly selected test file for system stability verification: at the same level we applied the method on 10% test set (same baseline) and trained the method based on the rest of the data, and we achieved the results shown in the second row in Tables 1 and 2.

[4] As most parts of the computation are run prior to the final detection, the system could be applied easily in online interactive applications.

Table 1. Flattened confusion matrices for all 6 classification results – True Pos. shows the number of texts which correctly classified as Okay; False Pos. shows the number of texts which falsely classified as Okay; True Neg. shows the number of texts which correctly classified as Flame and the False Neg. shows the number of texts which falsely classified as Flame.

True Pos.	False Pos.	True. Neg.	False Neg.	Classification#
936	102	349	138	1
89	16	36	11	2
999	39	385	102	3
84	3	27	8	4
1022	16	454	33	5
86	0	32	4	6

At the second classification level, we used the most expressive selected features (~1700 features selected by classification); the results of the Naïve Bayes Multinomial Updateable Classifier, applied with 10-folds cross-validation are shown in the third row of the Tables 1 and 2. This results show that the second level of classification increased the software performance about 7%.

As above, we applied the method on 10% test set (same baseline) and trained the system based on the rest and we achieved the results shown in the fourth row in Tables 1 and 2.

At this stage, raising the system's discriminative power and going beyond the previous-level accuracy (~91%) was pretty tough task. The software needed lots of consideration and going through the structural details of IALD entries in order to increase the detection power beyond 91%. Hence, we applied the DTNB (Decision Table/Naive Bayes hybrid classifier) rule-based classifier based on extra added information extracted from IALD and its built-in semantic rules (pattern matching modules).

The third level results, using 10-fold cross-validation are in row 5 of the Tables 1 and 2. This result shows that performing the last level improves the accuracy by a valuable extra 6%.

T**able 2.** Performance comparison along the three levels of classifications, for cross-validation (C.V.) on the training data, and on the test set.

Results → / Experiments ↓		Correctly Classified	Incorrectly Classified	Okay Precision	Flame Precision	Row No
First level Classification	10 old C.V.	84.26%	15.73%	87.2%	77.4%	1
	10% Test Size	81.37%	18.62%	86.0%	56.3%	2
Second level Classification	10 Fold C.V.	90.75%	9.24%	90.7%	90.8%	3
	10% Test Size	90.98%	9.01%	9.13%	90.0%	4
Third level Classification	10 Fold C.V.	96.78%	3.21%	96.9%	96.6%	5
	10% Test Size	96.72%	3.27%	95.6%	100%	6

As with the previous levels, we tried to verify the stability of the achieved results, so we applied the method on a test set with size of 10% of the data, and obtained the results shown in row 6 of Tables 1 and 2.

When we considered the above results and the results of other numerous experiments that we run, we clearly observed that the stability of the system after each level rose, and at the last level, the results on cross-validation and on the test set were quite similar.

If we consider the pair-wise agreement of judges, on the data from the previous annotation project [2] (which was part of our data), we see that the pair-wise agreement between human judges (based on the same definition of a flame message) on average is 92%, whereas if we take a look at other survey results (on similar but different data), we can see that although the agreement rate is 98% for non-flammatory messages, this rate diminished to 64% consensus for flame messages [1]. One important issue for human annotation which should be taken into account is that the distribution of the data (balanced/unbalanced) does not have any considerable influence on human judgments, unlike for the machine learning classifiers.

Hence, our higher percentage of agreement with the labels shows that the current software has a high level of adaptivity, based on the training dataset, and the IALD patterns and weights. Therefore, we can conclude that our method has a high capacity of being customized for any specific application.

The reasons for discrepancies between human judges (with the same problem definition) could be their different sensitivity, mood, background and some other subjective conditions. Human judgment is subjective and it is not necessarily the same among different people. It is thus helpful to have a standard detection system that can pass judgments based on some constant predefinitions, patterns and rules.

Unfortunately, no flame detection software is freely available for trial or research purposes, therefore we cannot directly compare our results to results of other systems on our dataset.

6 Discussion

Many of our IALD entries are applied as semantic classification rules. In the third level of classification, we attempt to match each of the corresponding patterns that have been built regarding the entry's *wild cards* or some additional prefix, suffix or special characters (leading or trailing), which help to distinguish whether the containing instance is a *Flame* or an *Okay* instance.

The advantages of the method could be listed as:

- The software can be used for message level or sentence level classification application in real-time applications (a fraction of a second for each new context).
- Our system benefits from both statistical models and rule-based patterns, in addition to specific semantic patterns inside the IALD, and does not rely on only one of them.

- Our software is not very sensitive to punctuation and grammatical mistakes.
- The method could be adapted in time, based on user feedback.

Among the limitations of our system is the fact that it does not consider the syntactical structure of the messages explicitly and could be equipped with some modules designed for subjectivity detection based on their lexicons (in this case we have to take into account that the length of each message would be a limitation for the method).

As we apply some patterns from IALD, as well as classifier models for flame detection, it is important to prevent training the classifiers based on some instances in which the assigned labels are opposed to some of IALD built-in weighted patterns and vice versa. Otherwise, the system will suffer from a considerable level of noise in the data.

7 Conclusion and Future Work

We designed and implemented novel and very efficient flame detection software. It applies models from multi-level classifiers, boosted by an Insulting and Abusing Language Dictionary. We built two rule-based auxiliary systems; one of them is the last level of our classifiers and the other is used for building patterns out of the IALD repository. The software performs with a high level of accuracy for both normal text and for flames.

Our flame detection method can be modified based on any accumulative training data and applied on any collaborative writing web site in which people can add or modify content, in the style of Wikipedia. It could also be handy for some web-logs or some specialist forums. The software could also be adapted for some kinds of spam detection for any type of text messaging services, suc as cellular phone SMS. It also could be useful over text chat services, as well as any comment acceptance posts in social networking sites like *Orkut* and *Facebook*.

In future work, we could apply second order co-occurrence features (Pedersen et. al. [30]) in order to extract more semantic information by processing surrounding terms and contexts of each preliminarily detected flame. We could add a synchronized adaptive weight modifier module to the IALD accessory, based on further provided training data.

Acknowledgements. We gratefully acknowledge the contribution of Dr. Melanie J. Martin for sharing the 1288 annotated messages; this data was crucial for the progress of this research. Natural Sciences and Engineering Research Council of Canada supports the research of the second and fourth author.

References

1. Spertus, E. Smokey: Automatic recognition of hostile messages. In Proceedings of the Eighth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), pp. 1058-1065 (1997)
2. Martin, M.J.: Annotating flames in Usenet newsgroups: a corpus study. For NSF Minority Institution Infrastructure Grant Site Visit to NMSU CS department (2002)
3. Wiebe, J., Wilson, T., Bruce, R. Bell, M. and Martin, M.: Learning Subjective Language. Computational Linguistics, 30, (3):277-308 (2004)
4. Gyamfi, y., Wiebe, J., Mihalcea, R. and Akkaya, C.: Integrating Knowledge for Subjectivity Sense Labeling. Joint Conference of the North American Chapter of the Association for Computational Linguistics and the Human Language Technologies Conference (NAACL-HLT 2009).
5. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation, 39 (2-3): 165-210 (2005)
6. Hall, M., Frank, E.: Combining Naive Bayes and Decision Tables. FLAIRS Conference: 318-319 (2008)
7. Wiebe, J., Wilson, T., Bell, B.: Identifying Collocations for Recognizing Opinions. Proc. ACL 01 Workshop on Collocation. Toulouse, France, (2001)
8. Mahmud, A., Ahmed, K.Z., Khan, M.: Detecting flames and insults in text, Proc. of 6th International Conference on Natural Language Processing (ICON-2008), CDAC Pune, India, December 20 - 22 (2008)
9. Wiebe, J., Bruce, R., Bell, M., Martin, M., Wilson, T.: A Corpus Study of Evaluative and Speculative Language. Proceedings of 2nd ACL SIGdial Workshop on Discourse and Dialogue. Aalborg, Denmark (2001)
10. Kaufer, D.: Flaming: A White Paper. (2000)
11. Witten, I., Frank, E., Gray, J.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, ISBN13: 9781558605527, (2008)
12. Richard A. Spears. Forbidden American English, ISBN: 9780844251493; (1991)
13. Bruce, R.F., Wiebe, J.: Recognizing subjectivity: a case study in manual tagging. Natural Language Engineering 5 (2), (1999)
14. Wiebe, J., Bruce, R.F., O'Hara, T.: Development and use of a gold standard data set for subjectivity classifications. In Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL-99), pp. 246-253 (1999)
15. Pang, B., Lee, L., Vaithyanathan, SH.: Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
16. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), 21(4):315–346, (2003)
17. Gordon, A., Kazemzadeh, A., Nair, A., Petrova, M.: Recognizing expressions of commonsense psychology in English text. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), pp. 208–215 (2003)
18. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proceedings of the

Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 129–136 (2003)

19. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2003), pp. 105–112 (2003)
20. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM-2003) (2003)
21. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of produce reviews. In Proceedings of the 12th International World Wide Web Conference (2003)
22. Riloff, E., Wiebe, J.,Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In Proceedings of the 7th Conference on Natural Language Learning, pp. 25–32. (CoNLL), (2003)
23. Razavi, A.H., Amini, R., Sabourin, C., Sayyad Shirabad, J., Nadeau, D., Matwin, S., De Koninck, J.: Classification of emotional tone of dreams using machine learning and text analyses. (Paper presented at the Meeting of the Associated Professional Sleep Society in Baltimore. Sleep, 31, A380-381 (2008)
24. Razavi, A.H., Amini, R., Sabourin, C., Sayyad Shirabad, J., Nadeau, D., Matwin,S., De Koninck, D.: Evaluation and Time Course Representation of the Emotional Tone of dreams Using Machine Learning and Automatic Text Analyses. - 19th Congress of European Sleep Research Society; ESRS-Glasgow Journal of Sleep Research, (In press). (2008)
25. Thelwall, M.: Fk yea I swear: Cursing and gender in a corpus of MySpace pages, Corpora, 3(1), 83-107 (2008)
26. McEnery, A.M.: Swearing in English: Bad Language, Purity and Power from 1586 to the Present, Routledge, London. (in press) ISBN 978-0415258371(2005)
27. McEnery, A.M., Xiao, Z.: 'Swearing in modern British English: the case of fuck in the BNC', Language and Literature, Volume 13, Issue 3, pp 235-268. ISSN 0963-9470- (2004)
28. McEnery, A.M., Baker, J.P.,Hardie, A.: 'Swearing and abuse in modern British English', in B. Lewandowska-Tomaszczyk and P.J. Melia (eds) Practical Applications of Language Corpora, Peter Lang, Hamburg, pp 37-48. (2000)
29. McEnery, A.M., Baker, J.P.,Hardie, J.: 'Assessing claims about language use with corpus data – swearing and abuse', in J. Kirk (ed) Corpora Galore, Rodopi, Amsterdam, pp 45-55. (2000)
30. T. Pedersen, A. K. Kulkarni, R. Angheluta, Z. Kozareva and Th. Solorio. An Unsupervised Language Independent Method of Name Discrimination Using Second Order Co-occurrence Features - The Seventh International Conference on Intelligent Text Processing and Computational Linguistics, Volume 3878 of Lecture Notes in Computer Science, Springer, , Mexico City, Mexico. February 19-25, 2006.