Emotion Classification in a Serious Game for Training Communication Skills

Frederik Vaassen and Walter Daelemans

CLiPS Research Center, University of Antwerp

Abstract

We describe the natural language processing component of a new serious gaming project, *deLearyous*, which aims at developing an environment in which users can improve their communication skills by interacting with a virtual character in (Dutch) written natural language. The virtual characters' possible dialogue paths are defined by Leary's Rose, a framework for interpersonal communication. In order to apply this framework, input sentences must be classified into one of four possible "emotion" classes.

We tried to carry out this emotion classification task using several machine learning algorithms. More specifically, classification performance was measured using TiMBL –a memory-based learner–, a Naïve Bayes classifier, Support Vector Machines and Conditional Random Fields. Training was done on a relatively small dataset of manually tagged sentences. A large number of different features was extracted from the dataset, and a good feature subset was selected using a combination of a genetic algorithm and various filter metrics.

We achieved the best results using the memory-based learner TiMBL, using a combination of word unigrams, lemma trigrams and dependency structures. With this setup, 52.5% of the sentences were classified into the correct emotion quadrant, which is a significant improvement over the statistical baseline (25.15%) and over the scores achieved with a pure bag-of-words approach (41.6%).

1 Introduction

1.1 *deLearyous*

Interpersonal communication is rarely simple. Our conversation partners may hold different opinions from our own, they may not be in the mood to listen to what we have to say, or they may simply not like us very much. Yet, we would still like to get our point across, preferably without disrupting our relationship with whomever it is we are talking to.

Communication skills are especially important in a professional context. The outcome of negotiations, for instance, may have a big impact on the future of a company. Likewise, an employer can avoid many complications if he manages to tell his employees about potentially unpopular management changes in a diplomatic way. For this reason, many companies invest in communication training for their employees. This training is usually done by acting out short scenarios with specialized actors. Hiring these actors, however, is a significant expense, and the

Proceedings of the 20th Meeting of Computational Linguistics in the Netherlands
Edited by: Eline Westerhout, Thomas Markus, and Paola Monachesi.

time which employees can spend on training their communication skills is limited by the company's budget or the actor's schedule.

It is to counter these two downsides that the *deLearyous* project was devised. The goal of the project is to develop a serious video game that can be used to assist in the training of interpersonal communication skills. The video game will present the player with a virtual autonomous character with whom he can interact by typing in natural language sentences (in Dutch). The virtual character will then reply in a way that fits the user's input and the underlying communication framework. By conversing with the virtual character and figuring out what conversation tactics work best to achieve their goals, players will be able to improve their own communication skills without the constraints that working with real actors imposes.

The *deLearyous* project is a co-operation between the e-Media Research Lab at Groep T in Leuven, the CLiPS Research Center at the University of Antwerp and Opikanoba, a company specialized in e-learning. The e-Media Research Lab will develop the dialogue manager as well as the audio and video modules, while Opikanoba will write the scenario for the virtual character. We, at CLiPS, will focus entirely on the Natural Language Processing component of the project, which is also the focus of this paper.

1.2 Leary's Rose

Several frameworks have been developed to describe the dynamics involved in interpersonal communication. The framework used in the *deLearyous* project is the *Interpersonal Circumplex*, also known as *Leary's Rose* (Leary 1957).

Leary's Rose (Figure 1) has both a descriptive and a predictive function: it can be used to position any participant in a discussion according to his state of mind and behavior, but it can also be used to predict (to a certain extent) the conversation partners' reaction to the speaker's behavior.

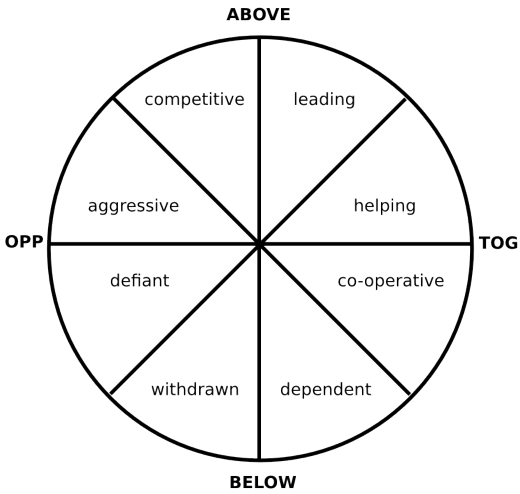

Figure 1: Leary's Rose

The Rose is defined by two axes: the *above-below* axis (vertical), which tells us whether the speaker is being dominant or submissive towards the listener; and the *together-opposed* axis (horizontal), which says something about the speaker's willingness to co-operate with the listener. The axes divide the Rose into four quadrants, and each quadrant can again be divided into two octants. Below are a few example sentences with their corresponding position on Leary's Rose.

- My name is John, how can I be of assistance? - **helping**
- How do you suggest we continue from here? - **dependent**
- That's not my fault, administration's not my responsibility! - **defiant**
- If you're going to be rude there's no use in continuing this conversation! - **aggressive**

Leary's Rose also allows us to predict what position the listener is most likely going to take in reaction to the way the speaker positions himself.

Two types of interactions are at play in Leary's Rose, one of complementarity and one of similarity:

- *above*-behavior triggers a response from the *below* zone and vice versa. When addressed by someone acting submissive, the listener's instinctive reaction will be to act more dominant and to take the lead, and vice versa.
- *together*-behavior triggers a response from the *together* zone, while *opposed*-behavior triggers a response from the *opposed* area of the Rose. When the speaker is being friendly towards the listener, the listener will be more likely to respond in kind, while unfriendly behavior is likely to elicit an unfriendly response.

The speaker can thus influence the listener's emotions (and consequently, his response) by consciously positioning himself in the quadrant that will likely trigger the desired reaction. Should the speaker wish to draw the listener out of the "aggressive" octant towards the "co-operative" octant, for instance, he would have to position himself in the *above-together* quadrant to gradually coax the listener to a more favorable disposition.

A first and crucial step in the development of *deLearyous* is making it possible to automatically detect the player's position on Leary's Rose. Since the player will interact with the virtual character by typing in natural language sentences, it is from these sentences that the "emotion" information is to be extracted. The emotion classification task will be the focus of the rest of this paper.

2 Related Work

To our knowledge, no work has yet been done specifically on the automatic classification of sentences based on Leary's Rose or on other incarnations of the Interpersonal Circumplex. There has however been quite a bit of research in the broader areas of automatic sentiment and emotion classification.

The techniques that have been used for sentiment and emotion classification can roughly be divided into pattern-based methods and machine-learning methods. An often-used technique in pattern-based approaches is to use pre-defined lists of keywords which help determine the instance's overall sentiment orientation or emotion contents. The AESOP system by Goyal et al. (2010), for instance, attempts to analyze the affective state of characters in fables by identifying affective verbs and by using a set of projection rules to calculate the verbs' influence on their patients. Balahur et al. (2010) evaluate several sentiment-annotated lexical resources (including MicroWNOp, WordNet Affect, SentiWordNet and their own JRC Tonality resource) on a set of newspaper quotes by computing sentiment in a window around the target of the quote.

Another possible approach is to let a machine learner determine the appropriate sentiment/emotion class. Mishne (2005) and Keshtkar and Inkpen (2009), for instance, attempt to classify LiveJournal posts according to their mood using Support Vector Machines trained with frequency features (word counts, POS-counts), length-related features (length of posts/sentences/...), semantic orientation features (using WordNet to calculate the distance of each word to a set of manually classified keywords) and special symbols (emoticons). Tsur et al. (2010) developed a system to recognize sarcasm in user opinions. They compiled feature vectors using punctuation-based features and patterns of high-frequency vs. content words, and applied a k-NN-like approach for classification.

Finally, Rentoumi et al. (2010) posited that combining the rule-based and machine learning approaches can have a positive effect on classification performance. By classifying strongly figurative examples using Hidden Markov Models while relying on a rule-based system to classify the mildly figurative ones, the overall performance of the classication system is improved.

3 Methodology

We have chosen to use a machine learning approach to try to automatically position Dutch sentences on Leary's Rose. Starting from a training set of sentences labeled with their position on the Rose, a machine learner should be able to pick up on cues that will allow the classification of new sentences into the correct emotion class.

Our approach falls within the domain of text categorization (Sebastiani 2002), which focuses on the automatic classification of text into predefined categories. Most text-categorization systems first extract features (terms, n-grams, ...) from a set of pre-classified documents. From these features, they then select those that are most helpful in predicting the document's category. This new feature subset is used to train a machine learner, which will then be able to classify new documents into the correct class. Text categorization has been used successfully for a wide array of applications, including document filtering, categorization of web content and authorship attribution (Luyckx and Daelemans 2005).

An important advantage of the machine learning approach compared to pattern-based approaches is that machine learners are able to take advantage of complex interactions between features, interactions which are often impossible to capture using handcrafted patterns. Since there are no easily identifiable keywords or syntactic structures that are consistently used with a position on Leary's Rose, using a machine learning approach is a logical choice for this emotion classification task.

3.1 Data

We compiled a small dataset of 339 Dutch sentences labeled according to their position on Leary's Rose. A large part of these sentences were taken from *Beïnvloed anderen, begin bij jezelf* (van Dijk 2000), a book specifically describing the workings of the Rose. Other sentences originated from scenarios which were specifically written for this purpose by Opikanoba and colleagues at CLiPS. The resulting dataset is relatively well balanced across the four quadrants of the framework:

	OPP_B	OPP_A	TOG_B	TOG_A
# of instances	80	82	96	81

3.2 Feature Extraction

From this dataset we extracted a wide range of different features. The sentences were first parsed with Tadpole, a Dutch language parser (van den Bosch et al. 2007), which allowed us to extract linguistic information such as word tokens, lemmas, part-of-speech tags, syntactic functions and dependency structures. The actual feature vectors were then generated on the basis of this linguistic information by using a "bag of n-grams" approach, i.e. by constructing n-grams (unigrams, bigrams and trigrams) of each feature type (e.g. n-grams of word tokens, n-grams of part-of-speech tags...) and by counting for each n-gram in the training data how many times it occurs in the current instance. Additionally to these n-gram counts, we also included punctuation counts, average word length and average sentence length.

3.3 Feature Subset Selection, Parameter Optimization and Genetic Algorithms

We reduced the dimensionality of the resulting feature vectors by selecting subsets of informative features using a variety of filter metrics –specifically gainratio, infogain, the Gini coefficient and χ^2.

Since it is possible to vary the feature types included in the feature vectors as well as the filter metrics for subset selection and the number of features in the selected subset, testing out all possible combinations of these parameters would be a prohibitively time-consuming task[1] A solution to this problem is to use a genetic algorithm (we used Pyevolve[2], a genetic algorithm for Python) to try out different combinations of feature types, filter metrics and learner parameters for each individual learner, while maximizing the learner's classification accuracy[3].

3.4 Classification

The actual classification is done with one of four learners, TiMBL –a memory-based learner based on the k-NN algorithm– (Daelemans and van den Bosch 2005), a Naïve Bayes classifier as implemented in the Orange machine learning package for Python[4], a multiclass implementation of Support Vector Machines (SVM-multiclass[5]) and CRFsuite (Okazaki 2007), an implementation of Conditional Random Fields. Using a Naïve Bayes classifier and SVMs seemed like a natural choice for this type of classification problem, as these techniques have shown their worth in similar classification tasks in the past. TiMBL was co-developed by CLiPS and has turned out to be very useful for a wide range of NLP tasks, and it is also known to perform well on problems with small datasets. A CRF learner may seem like a less logical choice since the feature vectors we constructed no longer represent a sequence, but we have found CRFs to be surprisingly effective and have decided to include them in this overview.

---

[1] Exhaustively exploring subsets of up to three feature type combinations with TiMBL –without varying the learner parameters– would take approximately two weeks.

[2] *Pyevolve 0.6* by Christian S. Perone. - `http://pyevolve.sourceforge.net` (last visited on May 6th, 2010)

[3] The genetic algorithm was run for 20 generations using a crossover rate of 80% and a mutation rate of 1%. We used Pyevolve's default crossover method (one point crossover) and the "1D-list allele mutator". The parameters varied for TiMBL were the value of k, the weighting metric and the distance metric. For SVMs, we varied the tradeoff between margin and training error and the type of Kernel function. See sections 3.4 and 4.4 for a more detailed description of the classifiers and their parameters.

[4] *Orange 2.0b*, Laboratory of Artificial Intelligence, Faculty of Computer and Information Science, University of Ljubljana - `http://www.ailab.si/orange/` (last visited on May 6th, 2010)

[5] *SVM-multiclass 2.20* by Thorsten Joachims - `http://svmlight.joachims.org/svm_multiclass.html` (last visited on May 6th, 2010)

The instances were classified into one of the four quadrants of Leary's Rose and results were calculated on the basis of 10-fold cross validation.

4 Results

4.1 Baseline

The statistical baseline for this 4-class classification problem, taking into account the slight imbalances in the class distribution, is 25.15%. An additional and more useful baseline is the performance of each learner when using only token unigrams without any kind of feature subset selection. These baseline results are illustrated in Table 10.1.

	TiMBL	Naïve Bayes	SVM	CRF
statistical baseline	25.15%			
baseline accuracy	41.6%	41.6%	26.8%	44.2%

Table 10.1: Baseline scores using only token unigrams

4.2 Performance

Table 10.2 gives an overview of the top scores that we managed to achieve with each of the four learners, i.e. using the combination of features and learner parameters that was determined to give the best accuracy by the genetic algorithm. The "features" column indicates the types of features that have been used:

- w - word tokens
- l - lemmas
- c - characters
- d - dependency groups
- wlc - average word length based on the number of characters in a word
- wls - average word length based on the number of syllables in a word

The numbers appended to the feature types indicate the size of the n-grams used (1 - unigrams, 2 - bigrams, 3 - trigrams).

The parameters for each learner were determined using the genetic algorithm. TiMBL used a k-value of 1, gainratio for weighting and the dot-product metric as the distance metric. SVMs were used with standard parameters, except for the trade-off between training error and margin, which was set to 0.6[6].

All learners outperform the statistical baseline by a large margin, but the results of the memory-based learner TiMBL are especially interesting as they also signif-

[6] SVMs appear to be extremely sensitive to parameter changes, and there is a lot of interaction between learner parameters and the feature set used for training. We believe that the SVM scores can still be improved given a more stable dataset and a more thorough search through the parameter/feature space.

	features	# of feats	filter metric	accuracy	stdev.	F-score (macro avg.)
TiMBL	w1, l3, d	1000	Gini	52.5%	10.5%	52.2%
Naïve Bayes	l1, w3, wls	all	n/a	46.6%	8.8%	46.8%
SVM	c1, l3, wlc	250	infogain	35.7%	11.3%	35.7%
CRF	w1, wlc	500	infogain	47.8%	11.0%	47.7%

Table 10.2: Top results per learner

icantly improve on the second baseline[7], which uses token unigrams only. 52.5% of the sentences are assigned the correct quadrant in Leary's Rose.

Table 10.3 shows the detailed class scores for the TiMBL experiment, while Table 10.4 shows the confusion matrix.

	precision	recall	F-score
TOG_A	0.53	0.49	0.51
OPP_A	0.55	0.48	0.51
OPP_B	0.55	0.53	0.54
TOG_B	0.48	0.61	0.53

Table 10.3: Class scores for TiMBL

	true TOG_A	true OPP_A	true OPP_B	true TOG_B
TOG_A	43	17	8	19
OPP_A	17	45	18	13
OPP_B	9	12	44	18
TOG_B	12	8	10	46

Table 10.4: Confusion matrix for TiMBL

Performance is relatively uniform over all classes and there seem to be no significant trends in the classification errors. This also makes the analysis and correction of errors more difficult, as there are no clear problems that can be resolved easily by identifying a cue the learner does not pick up on. The current scores are most likely limited by the small size (339 instances) and limited coverage of the dataset, and we expect to see improvements once more data has been gathered.

4.3 The Importance of Feature Subset Selection

A brief look at the top performing feature sets in Table 10.5 (repeated from Table 10.2) tells us that there isn't simply one set of features that is best for all the

[7] The statistical significance of the difference between TiMBL's results and the bag-of-words baseline was tested using a paired t-test which yielded a p-value of 0.0257, which is considered statistically significant.

	features	# of feats	filter metric	accuracy
TiMBL	w1, l3, d	1000	Gini	52.5%
Naïve Bayes	l1, w3, wls	all	n/a	46.6%
SVM	c1, l3, wlc	250	infogain	35.7%
CRF	w1, wlc	500	infogain	47.8%

Table 10.5: Top results per learner

different learners, as each learner performs best with different types of features. Classifiers also seem to react differently to feature subset selection, with Naïve Bayes performing best when provided with all features while other learners benefit from using a reduced set of features. To illustrate the importance of finding the right feature set for the right learner, Table 10.6 shows how Naïve Bayes, SVM and CRF fare when we train them using the top feature set for TiMBL. There is a clear and significant drop in performance when using a feature set that isn't adapted for the learner in use.

	Naïve Bayes	SVM	CRF
top accuracy	46.6%	35.7%	47.8%
using TiMBL features	30.4%	26.5%	36.9%
difference	-16.2%	-9.2%	-10.9%

Table 10.6: Importance of adapted feature subsets per learner

While different feature subsets are needed to get the best out of each learner, there are some elements that stay constant. Word tokens and lemmas, for instance, are consistently present in the top feature combinations for every learner. Table 10.7 illustrates this by comparing the top feature combination for each learner to the first combination *not* using word tokens or lemmas. There is only a small performance drop for Naïve Bayes, SVMs and CRF, but TiMBL results without words or lemmas are significantly lower.

	TiMBL	Naïve Bayes	SVM	CRF
top accuracy	52.5%	46.6%	35.7%	47.8%
without tokens or lemmas	38.4%	41.9%	33.9%	46.6%
difference	-14.1%	-4.7%	-2.2%	-1.2%

Table 10.7: Importance of word tokens and lemmas as features

When we examine the specific words and lemmas that the filter metrics propose as the most relevant features, we see that most of them seem instinctively plausible as important cues related to Leary's Rose. Question marks and exclamation marks, for instance, are amongst the 10 most relevant features, and their relevance is easily illustrated by examining the following sentences from the *deLearyous* dataset:

- Wat vindt u zelf van dit voorstel**?** (*What do you think of this suggestion?*) - **TOG_B**
- Zoek het zelf maar uit**!** (*Just figure it out for yourself!*) - **OPP_A**

In the first sentence, the speaker positions himself in an inferior, expectant position towards the listener. The listener is given full control over the situation as he can still reject the speaker's suggestion. The fact that the sentence is a question contributes to the speaker's submissive, cautious position. The question mark is therefore associated with the "below" half of Leary's Rose.

In the second sentence, the exclamation mark accentuates the dominant position of the speaker. The speaker is annoyed or angry at the listener and authoritatively tells him what to do. The exclamation mark is thus associated with the "above" half of the Rose, since it usually points to dominant behavior.

It should be noted that exclamation marks needn't always express dominance, and question marks don't always point to submission. These features on their own are far from enough to classify sentences into the quadrants of Leary's Rose. It's only in combination with many other features that they turn out to be especially useful.

Another interesting feature pair that shows up in the top 10 most relevant features is the distinction between the personal pronouns "u" and "je". "U" is a Dutch pronoun that marks politeness, while "je" is the more general unmarked second person pronoun. Using one or the other tells us a lot about the power dynamics in a conversation:

- Ik begrijp dat **u** kwaad bent, mevrouw... (*I understand that you're angry, Mrs...*) - **TOG_B**
- Ik verwacht dat **je** naar me luistert! (*I expect you to listen to me!*) - **OPP_A**

Finally, there's "we", the first person plural pronoun, which instinctively guides the interpretation of a sentence to the "together" half of Leary's Rose:

- Gaat u even zitten, dan zoeken **we** samen naar een oplossing. (*"Please sit down, we'll find an answer together."*) - **TOG_A**

4.4 The Importance of Parameter Optimization

For those learners that allow different parameters, determining the most efficient parameters may be just as important as choosing the appropriate feature types. Indeed, a classifier that has been trained with suboptimal features but uses good parameters might not be far off, performance-wise, from a learner trained with the best features but using the "wrong" parameters (see also Daelemans et al. (2003)). Table 10.8 illustrates this problem: it compares the results of a classification task with a suboptimal set of features (but with "good" learner parameters) with the results of a classification task where the learner parameters have not been optimized (but the features are the "correct" ones).

For the purpose of this experiment, the "wrong" feature types used were character bigrams, part-of-speech trigrams and unigrams of syntactic functions. The "bad" parameters were the default parameters (for TiMBL: k = 1, weighting = gainratio, distance metric = weighted overlap; for SVMs: training error/margin trade-off = 0.01).

	TiMBL	SVM
top accuracy ("good" features and parameters)	52.5%	35.7%
using "bad" features	**25.0%**	26.3%
using "bad" parameters	30.1%	**25.1%**

Table 10.8: Performance when using "bad" learner parameters or feature subsets

For the TiMBL experiments, the results of the classification task using "bad" parameteres are almost as low as for the experiment using the "wrong" feature types. For SVMs, the comparison is even more striking, as the optimized learner trained on the "bad" features even outperforms (though not significantly so) the learner trained on the "good" ones without parameter optimization.

5 Conclusion

We have outlined the natural language component of a new project, project *deLearyous*, which aims to create a serious video game that will allow players to interact with a virtual character using Dutch natural language sentences. Through this interaction with the virtual character, the players should be able to improve their communication skills by learning to use Leary's Rose, a framework for interpersonal communication. The natural language processing component is of primordial importance to *deLearyous*, and we described how we attempted to identify the position of the player in Leary's Rose on the basis of their textual input.

We chose to use machine learning techniques to perform this classification task, and we used a small dataset of 339 Dutch sentences to test four different learner algorithms: a memory-based learner (TiMBL), a Naïve Bayes classifier, a multi-class implementation of Support Vector Machines and Conditional Random Fields. So far, we managed to achieve an accuracy of 52.5% using TiMBL and a combination of word token unigrams, lemma trigrams and average word length as features.

We have determined that word tokens and lemmas are important feature types for all learners, and have looked at some of these top features in more detail to determine how they relate to Leary's Rose. We have also shown the influence that the use different feature subsets has on each learner, and noted that finding the right set of learner parameters is at least as important as finding a good feature subset.

6 Future Research

At the moment, we can only classify one out of two sentences into the correct quadrant of Leary's Rose, which is insufficient if *deLearyous* is to be a useful tool for communication training. An additional problem is that the final product will not restrict itself to the four quadrants on the Rose, but it will incorporate dynamics relating to the eight octants, which complicates the classification task significantly.

There are several elements that will help balance out these difficulties, however. Right now, the learners have been trained on a very small dataset that only sparsely covers a wide array of different possible communication scenarios. As the *deLearyous* project moves forward, one specific scenario will be chosen, which will allow us to expand the dataset with relevant instances. Additionally, each sentence is now looked at in isolation, i.e. the machine learner has no idea of what has happened in the conversation prior to the current sentence. Once the scenario has been established, however, it will be possible to also integrate information about the context, hopefully improving classification accuracy in the progress.

Until now, we have not used any features based on emotion or sentiment lexicons. Most existing resources (SentiWordNet, WordNet Affect, etc.) have been compiled for the English language only, though Jijkoun and Hofmann (2009) have sucessfully constructed a Dutch subjectivity lexicon based on the English lexicon of OpinionFinder (Wilson et al. 2005) and Cornetto (Vossen et al. 2007) –an extension of the Dutch WordNet. Since integrating sentiment orientation features into the machine-learning approach has been successful for English (Mishne (2005), Keshtkar and Inkpen (2009), Inkpen et al. (2009)), there is reason to believe that this technique will also prove to be useful for Dutch.

A final aspect that needs to be researched is the problem of reliability. Until now, all data was manually labeled by one annotator, but identifying emotions in written text is not straightforward, even for humans. It is therefore important to have several annotators label the data and to check inter- and intra-annotator agreement scores. On the basis of these scores, we can then determine a ceiling beyond which we can not realistically expect the machine learners to perform.

7 Acknowledgements

This study was made possible through financial support from the IWT (the government agency for Innovation by Science and Technology, TETRA-project *deLearyous*). We would also like to thank the CLIN reviewers for their helpful feedback.

References

Balahur, Alexandra, Ralf Steinberger, Mijail Alexandrov Kabadjov, Vanni Zavarella, Erik Van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva (2010), Sentiment analysis in the news, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10).*

Crammer, Koby and Yoram Singer (2002), On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of Machine Learning Research* pp. 265–292, MIT Press.

Daelemans, Walter and Antal van den Bosch (2005), *Memory-Based Language Processing*, Cambridge, UK: Cambridge University Press.

Daelemans, Walter, Véronique Hoste, Fien De Meulder, and Bart Naudts (2003), Combined optimization of feature selection and algorithm parameter interaction in machine learning of language, *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, Berlin: Springer, pp. 84–95.

Goyal, Amit, Ellen Riloff, Hal Daumé III, and Nathan Gilbert (2010), Toward plot units: Automatic affect state analysis, *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*.

Inkpen, Diana, Fazel Keshtkar, and Diman Ghazi (2009), Analysis and generation of emotion in texts, *in* Clujeana, Presa Universitara, editor, *KEPT 2009 Knowledge Engineering-Principles and Techniques, Selected Papers*, Militon Frentiu and Horia F. Pop, pp. 3–13.

Jijkoun, Valentin and Katja Hofmann (2009), Generating a non-english subjectivity lexicon: Relations that matter., *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, The Association for Computer Linguistics, pp. 398–405.

Keshtkar, Fazel and Diana Inkpen (2009), Using sentiment orientation features for mood classification in blogs, *IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2009), Dalian, China*.

Leary, T. (1957), *Interpersonal Diagnosis of Personality: Functional Theory and Methodology for Personality Evaluation*, New York: Ronald Press.

Luyckx, Kim and Walter Daelemans (2005), Shallow text analysis and machine learning for authorship attribution, *Computational Linguistics in the Netherlands 2004: Selected papers from the Fifteenth CLIN Meeting*, pp. 149–160.

Mishne, Gilad (2005), Experiments with mood classification in blog posts, *Proceedings of the 1st Workshop on Stylistic Analysis Of Text For Information Access*.

Okazaki, Naoaki (2007), CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Rentoumi, Vassiliki, Stefanos Petrakis, Manfred Klenner, George A. Vouros, and Vangelis Karkaletsis (2010), United we stand: Improving sentiment analysis by joining machine learning and rule based methods, *in* Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, European Language Resources Association (ELRA), Valletta, Malta.

Sebastiani, Fabrizio (2002), Machine learning in automated text categorization, *ACM Comput. Surv.* **34** (1), pp. 1–47, ACM, New York, NY, USA.

Tsur, Oren, Dmitry Davidov, and Ari Rappoport (2010), ICWSM - a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews, *in* Hearst, Marti, William Cohen, and Samuel Gosling, editors, *Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM-2010)*, The AAAI Press, Menlo Park, California.

van den Bosch, Antal, Bertjan Busser, Sander Canisius, and Walter Daelemans (2007), An efficient memory-based morphosyntactic tagger and parser for dutch, *in* Eynde, F. Van, P. Dirix, I. Schuurman, and V. Vandeghinste, editors, *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting*, Leuven, Belgium, pp. 99–114. http://ilk.uvt.nl/tadpole/.

van Dijk, Bert (2000), *Beïnvloed anderen, begin bij jezelf. Over gedrag en de Roos van Leary*, 4th ed., Thema.

Vossen, Piek, Katja Hofmann, Maarten de Rijke, Erik Tjong Kim Sang, and Koen Deschacht (2007), The Cornetto database: Architecture and user-scenarios, *in* Moens, M.-F., T. Tuytelaars, and A.P. de Vries, editors, *Proceedings DIR 2007*, pp. 89–96.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann (2005), Recognizing contextual polarity in phrase-level sentiment analysis, *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 347–354.