Automatic Emotion Classification for Interpersonal Communication

Frederik Vaassen
CLiPS - University of Antwerp
S.L.202, Lange Winkelstraat 40-42
B-2000 Antwerpen, Belgium
frederik.vaassen@ua.ac.be

Walter Daelemans
CLiPS - University of Antwerp
S.L. 203, Lange Winkelstraat 40-42
B-2000 Antwerpen, Belgium
walter.daelemans@ua.ac.be

Abstract

We introduce a new emotion classification task based on Leary's Rose, a framework for interpersonal communication. We present a small dataset of 740 Dutch sentences, outline the annotation process and evaluate annotator agreement. We then evaluate the performance of several automatic classification systems when classifying individual sentences according to the four quadrants and the eight octants of Leary's Rose. SVM-based classifiers achieve average F-scores of up to 51% for 4-way classification and 31% for 8-way classification, which is well above chance level. We conclude that emotion classification according to the Interpersonal Circumplex is a challenging task for both humans and machine learners. We expect classification performance to increase as context information becomes available in future versions of our dataset.

1 Introduction

While sentiment and opinion mining are popular research topics, automatic emotion classification of text is a relatively novel –and difficult– natural language processing task. Yet it immediately speaks to the imagination. Being able to automatically identify and classify user emotions would open up a whole range of interesting applications, from in-depth analysis of user reviews and comments to enriching social network environments according to the user's emotions.

Most experiments in emotion classification focus on a set of basic emotions such as "happiness", "sadness", "fear", "anger", "surprise" and "disgust". The interpretation of "emotion" we're adopting in this paper, however, is slightly more specific. We concentrate on the emotions that are at play in interpersonal communication, more specifically in the dynamics between participants in a conversation: is one of the participants taking on a dominant role? Are the speakers working towards a common goal, or are they competing? Being able to automatically identify these power dynamics in interpersonal communication with sufficient accuracy would open up interesting possibilities for practical applications. This technology would be especially useful in e-learning, where virtual agents that accept (and interpret) natural language input could be used by players to practice their interpersonal communication skills in a safe environment.

The emotion classification task we present in this paper involves classifying individual sentences into the quadrants and octants of Leary's Rose, a framework for interpersonal communication.

We give a brief overview of related work in section 2 and the framework is outlined in section 3. Section 4 introduces the dataset we used for classification. Section 5 outlines the methodology we applied, and the results of the different experiments are reported on in section 6. We discuss these results and draw conclusions in section 7. Finally, section 8 gives some pointers for future research.

2 Related Work

The techniques that have been used for 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 an instance's overall 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. Another possible approach –which we subscribe to– is to let a machine learner determine the appropriate 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, length-related features, semantic orientation features and features representing special symbols. Finally, Rentoumi et al. (2010) posit 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 classification system is improved.

Whereas emotion classification in general is a relatively active domain in the field of computational linguistics, little research has been done regarding the automatic classification of text according to frameworks for interpersonal communication. We have previously carried out a set of classification experiments using Leary's Rose on a smaller dataset (Vaassen and Daelemans, 2010), only taking the quadrants of the Rose into account. To our knowledge, this is currently the only other work concerning automatic text classification using any realization of the Interpersonal Circumplex. We expand on this work by using a larger dataset which we evaluate for reliability. We attempt 8-way classification into the octants of the Rose, and we also evaluate a broader selection of classifier setups, including one-vs-all and error-correcting systems.

3 Leary's Rose

Though several frameworks have been developed to describe the dynamics involved in interpersonal communication (Wiggins, 2003; Benjamin, 2006), we have chosen to use the Interpersonal Circumplex, better known as "Leary's Rose" (Leary, 1957).

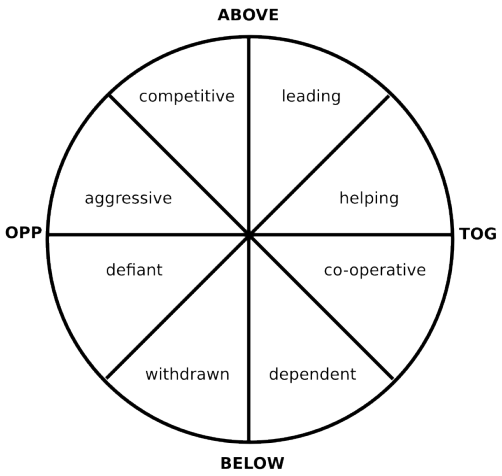

Figure 1: Leary's Rose

Leary's Rose (Figure 1) 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.

What makes the Circumplex especially interesting for interpersonal communication training is that it also allows one to predict (to some extent) 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 (complementary) response from the *below* zone and vice versa, while *together*-behavior triggers a (similar) response from the *together* zone and *opposed*-behavior triggers a (similar) response from the *opposed* area of the Rose. 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.

4 Dataset

To evaluate how difficult it is to classify sentences –both manually and automatically– according to Leary's Rose, we used an expanded version of the dataset described in Vaassen and Daelemans (2010).

The dataset[1] contains a total of 740 Dutch sentences labeled according to their position on the Interpersonal Circumplex. The majority of the sentences were gathered from works specifically designed to teach the use of Leary's Rose (van Dijk, 2000; van Dijk and Moes, 2005). The remaining sentences were specifically written by colleagues at CLiPS and by e-learning company Opikanoba. 31 sentences that were labeled as being purely neutral were removed from the dataset for the purposes of this classification experiment, leaving a set of 709 Dutch sentences divided across the octants and quadrants of the Interpersonal Circumplex. Table 1 shows the class distribution within the dataset and also lists the statistical random baselines for both 8-class and 4-class classification tasks.

709 sentences	TOG_A: 165 sentences	leading: 109 sentences
		helping: 56 sentences
	TOG_B: 189 sentences	co-operative: 92 sentences
		dependent: 97 sentences
	OPP_B: 189 sentences	withdrawn: 73 sentences
		defiant: 116 sentences
	OPP_A: 166 sentences	aggressive: 71 sentences
		competitive: 95 sentences
Baseline	25.4%	13.1%

Table 1: Distribution of classes within the dataset[2]

Below are a few example sentences with their corresponding position on the Rose.

- Please have a seat and we'll go over the options together. - **helping (TOG_A)**
- So what do you think I should do now? - **dependent (TOG_B)**
- That's not my fault, administration's not my responsibility! - **defiant (OPP_B)**
- If you had done your job this would never have happened! - **aggressive (OPP_A)**

4.1 Agreement Scores

Placing sentences on Leary's Rose is no easy task, not even for human annotators. An added complication is that the sentences in the dataset lack any form of textual or situational context. We therefore expect agreement between annotators to be relatively low.

To measure the extent of inter-annotator disagreement, we had four annotators label the same random subset of 50 sentences. The annotators were given a short introduction to the workings of Leary's Rose, and were then instructed to label each of the sentences according to the octants of the Rose using the following set of questions:

- Is the current sentence task-oriented (*opposed*) or relationship-oriented (*together*)?
- Does the speaker position himself as the dominant partner in the conversation (*above*) or is the speaker submissive (*below*)?
- Which of the above two dimensions (affinity or dominance) is most strongly present?

Annotators were also given the option to label a sentence as being purely neutral should no emotional charge be present.

Table 2 shows Fleiss' kappa scores calculated for 4 and 8-class agreement.

# of classes	κ
4	0.37
8	0.29

Table 2: Inter-annotator agreement, 4 annotators

Though the interpretation of kappa scores is in itself subjective, scores between 0.20 and 0.40 are usually taken to indicate "fair agreement".

The full dataset was also annotated a second time by the initial rater six months after the first annotation run. This yielded the intra-annotator scores in Table 3. A score of 0.5 is said to indicate "moderate agreement".

# of classes	κ
4	0.50
8	0.37

Table 3: Intra-annotator agreement

The relatively low kappa scores indicate that the classification of isolated sentences into the quadrants or octants of Leary's Rose is a difficult task even for humans.

As an upper baseline for automatic classification, we take the average of the overlaps between the

[1] Dataset available on request.

[2] "TOG" and "OPP" stand for *together* and *opposed* respectively, while "A" and "B" stand for *above* and *below*.

main annotator and each of the other annotators on the random subset of 50 sentences. This gives us an upper baseline of 51.3% for 4-way classification and 36.0% for the 8-class task.

5 Methodology

Our approach falls within the domain of automatic text categorization (Sebastiani, 2002), which focuses on the classification of text into predefined categories. 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. 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.

5.1 Feature Extraction

The sentences in our dataset were first syntactically parsed using the Frog parser for Dutch (Van den Bosch et al., 2007). From the parsed output, we extracted token, lemma, part-of-speech, syntactic and dependency features using a "bag-of-ngrams" approach, meaning that for each n-gram (up to trigrams) of one of the aforementioned feature types present in the training data, we counted how many times it occurred in the current instance. We also introduced some extra features, including average word and sentence length, features for specific punctuation marks (exclamation points, question marks...) and features relating to (patterns of) function and content words.

Due to efficiency and memory considerations, we did not use all of the above feature types in the same experiment. Instead, we ran several experiments using combinations of up to three feature types.

5.2 Feature Subset Selection

Whereas some machine learners (e.g. Support Vector Machines) deal relatively well with large numbers of features, others (e.g. memory-based learners) struggle to achieve good classification accuracy when too many uninformative features are present. For these learners, we go through an extra feature selection step where the most informative features are identified using a filter metric (see also Vaassen and Daelemans (2010)), and where only the top n features are selected to be included in the feature vectors.

5.3 Classification

We compared the performance of different classifier setups on both the 4-way and 8-way classification tasks. We evaluated a set of native multiclass classifiers: the memory-based learner TiMBL (Daelemans and van den Bosch, 2005), a Naïve Bayes classifier and SVM Multiclass (Tsochantaridis et al., 2005), a multiclass implementation of Support Vector Machines. Further experiments were run using SVM light classifiers (Joachims, 1999) in a one-vs-all setup and in an Error-Correcting Output Code setup (ECOCs are introduced in more detail in section 5.3.1). Parameters for SVM Multiclass and SVM light were determined using Paramsearch's two-fold pseudo-exhaustive search (Van den Bosch, 2004) on vectors containing only token unigrams. The parameters for TiMBL were determined using a genetic algorithm designed to search through the parameter space[3].

5.3.1 Error-Correcting Output Codes

There are several ways of decomposing multiclass problems into binary classification problems. Error-Correcting Output Codes (ECOCs) (Dietterich and Bakiri, 1995) are one of these techniques. Inspired by *distributed output coding* in signal processing (Sejnowski and Rosenberg, 1987), ECOCs assign a distributed output code –or "codeword"– to each class in the multiclass problem. These codewords, when taken together, form a code matrix (Table 4).

Class 1	0	1	0	1	0	1	0
Class 2	0	0	0	0	1	1	1
Class 3	1	1	1	1	1	1	1
Class 4	0	0	1	1	0	0	1

Table 4: Example code matrix

Each column of this code matrix defines a binary classification task, with a 0 indicating that the instances with the corresponding class label should be part of a larger negative class, and a 1 indicat-

[3]The fitness factor driving evolution was the classification accuracy of the classifier given a set of parameters, using token unigram features in a 10-fold cross-validation experiment.

ing the positive class. A binary classifier (or "dichotomizer") is trained for each column. When a new instance is to be classified, it is first classified by each of these dichotomizers, which each return their predicted class (1 or 0). The combined output from each dichotomizer forms a new codeword. The final class is determined by choosing the codeword in the code matrix that has the smallest distance (according to some distance metric) to the predicted codeword.

This method offers one important advantage compared to other, simpler ensemble methods: because the final class label is determined by calculating the distance between the predicted codeword and the class codewords, it is possible to correct a certain number of bits in the predicted codeword if the distance between the class codewords is large enough.

Formally, a set of ECOCs can correct $\lfloor \frac{d-1}{2} \rfloor$ bits, where d is the minimum Hamming distance (the number of differing bits) between codewords in the code matrix. The error-correcting capacity of an ECOC setup is thus entirely dependent on the code matrix used, and a great deal of attention has been devoted to the different ways of constructing such code matrices (Ghani, 2000; Zhang et al., 2003; Álvarez et al., 2007).

In our ECOC classification setup, we used code matrices artificially constructed to maximize their error-correcting ability while keeping the number of classifiers within reasonable bounds. For 4-class classification, we constructed 7-bit codewords using the exhaustive code construction technique described in Dietterich and Bakiri (1995). For the 8-class classification problem, we used a Hadamard matrix of order 8 (Zhang et al., 2003), which has optimal row (and column) separation for the given number of columns. Both matrices have an error-correcting capacity of 1 bit.

6 Results

All results in this section are based on 10-fold cross-validation experiments. Table 5 shows accuracy scores and average F-scores for both 4-way and 8-way classification using classifiers trained on token unigrams only, using optimal learner parameters. For TiMBL, the number of token unigrams was limited to the 1000 most predictive according to the Gini coëfficient[4]. All other learners used the full range of token unigram features. The Naïve Bayes approach performed badly on the 8-way classification task, wrongly classifying all instances of some classes, making it impossible to calculate an F-score.

	4-class		8-class	
	accuracy	F-score	accuracy	F-score
SVM Multiclass	47.3%	46.8%	31.6%	28.3%
Naïve Bayes	42.6%	40.1%	26.1%	NaN
TiMBL	41.3%	41.3%	23.6%	22.9%
SVM / one-vs-all	46.0%	45.4%	29.3%	27.2%
SVM / ECOCs	48.1%	47.8%	31.3%	26.3%
Random baseline	25.4%		13.1%	
Upper baseline	51.3%		36.0%	

Table 5: Accuracy and average F-scores - token unigrams

All classifiers performed better than the random baseline (25.4% for 4-class classification, 13.1% for classification into octants) to a very significant degree. We therefore take these token unigram scores as a practical baseline.

	feature types	accuracy	avg. F-score
SVM Multiclass	w1, l3, awl	49.4%	49.4%
TiMBL	w1, w2, l1	42.0%	42.0%
SVM / one-vs-all	l2, fw3, c3	51.1%	51.0%
SVM / ECOCs	l2, c3	52.1%	51.2%

Table 6: Best feature type combinations - quadrants[5]

	feature types	accuracy	avg. F-score
SVM / one-vs-all	w1, l1, c1	34.0%	30.9%
SVM / ECOCs	w2, fw3, c3	34.8%	30.2%

Table 7: Best feature type combinations - octants

We managed to improve the performance of some of the classifier systems by including more and different features types. Tables 6 and 7 show performance for 4-way and 8-way classification respectively, this time using the best possible combination

[4]The filter metric and number of retained features was determined by testing the different options using 10-fold CV and by retaining the best-scoring combination (Vaassen and Daelemans, 2010).

[5]The "feature types" column indicates the types of features that were used, represented as a letter followed by an integer indicating the size of the n-gram: w: word tokens, l: lemmas, fw: function words, c: characters, awl: average word length (based on the number of characters)

of up to three feature types[6] for every classifier setup where an improvement was noted.

We used McNemar's test (Dietterich, 1998) to compare the token unigram scores with the best feature combination scores for each of the above classifiers. For both 4-way and 8-way classification, the one-vs-all and ECOC approaches produced significantly different results[7]. The improvement is less significant for TiMBL and SVM Multiclass in the 4-way classification experiments.

Note that for classification into quadrants, the performance of the SVM-based classifiers is very close to the upper baseline of 50.3% we defined earlier. It is unlikely that performance on this task will improve much more unless we add context information to our interpersonal communication dataset. The 8-way classification results also show promise, with scores up to 30%, but there is still room for improvement before we reach the upper baseline of 36%.

In terms of classifiers, the SVM-based systems perform better than their competitors. Naïve Bayes especially seems to be struggling, performing significantly worse for the 4-class classification task and making grave classification errors in the 8-way classification task. The memory-based learner TiMBL fares slightly better on the 8-class task, but isn't able to keep up with the SVM-based approaches.

When we examine the specific features that are identified as being the most informative, we see that most of them seem instinctively plausible as important cues related to positions on Leary's Rose. Question marks and exclamation marks, for instance, are amongst the 10 most relevant features. So too are the Dutch personal pronouns "u", "je" and "we" – "u" being a second person pronoun marking politeness, while "je" is the unmarked form, and "we" being the first person plural pronoun. Of course, none of these features on their own are strong enough to accurately classify the sentences in our dataset. It is only through complex interactions between many features that the learners are able to identify the correct class for each sentence.

[6]The best feature type combination for each setup was determined experimentally by running a 10-fold cross-validation test for each of the possible combinations.

[7]4-class SVM one-vs-all: P=0.0014, 4-class SVM ECOCs: P=0.0170, 8-class SVM one-vs-all: P=0.0045, 8-class SVM ECOCs: P=0.0092

7 Conclusions

We have introduced a new emotion classification task based on the Interpersonal Circumplex or "Leary's Rose", a framework for interpersonal communication. The goal of the classification task is to classify individual sentences (outside of their textual or situational context), into one of the four quadrants or eight octants of Leary's Rose. We have outlined the annotation process of a small corpus of 740 Dutch sentences, and have shown the classification task to be relatively difficult, even for human annotators. We evaluated several classifier systems in a text classification approach, and reached the best results using SVM-based systems. The SVM learners achieved F-scores around 51% on the 4-way classification task, which is close to the upper baseline (based on inter-annotator agreement), and performance on 8-class classification reached F-scores of almost 31%.

8 Future Research

The initial results of the emotion classification tasks described in this paper are promising, but there is a clear sense that without some contextual information, it is simply too difficult to correctly classify sentences according to their interpersonal emotional charge. For this reason, we are currently developing a new version of the dataset, which will no longer contain isolated sentences, but which will instead consist of full conversations. We expect that having the sentences in their textual context will make the classification task easier for both human annotators and machine learners. It will be interesting to see if and how the classification performance improves on this new dataset.

Acknowledgments

This study was made possible through financial support from the IWT (the Belgian government agency for Innovation by Science and Technology, TETRA-project deLearyous). Many thanks go out to our colleagues at the e-Media Lab (Groep T, Leuven, Belgium) and Opikanoba, partners in the deLearyous project. We would also like to thank the WASSA 2.011 reviewers for their helpful feedback.

References

Victor Álvarez, Jose A. Armario, Maria D. Frau, Elena Martin and Amparo Osuna. 2007. Error Correcting Codes from Quasi-Hadamard Matrices. *Lecture Notes in Computer Science*, volume 4547/2007.

Lorna S. Benjamin, Jeffrey C. Rothweiler and Kenneth L. Critchfield. 2006. The Use of Structural Analysis of Social Behavior (SASB) as an Assessment Tool. *Annual Review of Clinical Psychology*, Vol. 2, No. 1.

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

Thomas G. Dietterich and Ghulum Bakiri. 1995. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*.

Thomas G. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computing*, volume 10.

Rayid Ghani. 2000. Using Error-Correcting Codes for Text Classification. *Proceedings of the Seventeenth International Conference on Machine Learning*.

Amit Goyal, Ellen Riloff, Hal Daume III and Nathan Gilbert. 2010. Toward Plot Units: Automatic Affect State Analysis. *Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*.

Thorston Joachims. 1999. *Making large-scale support vector machine learning practical.* MIT Press, Cambridge, MA.

Fazel Keshtkar and Diana Inkpen. 2009. Using Sentiment Orientation Features for Mood Classification in Blogs. *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2009)*.

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

Kim Luyckx. 2011. The Effect of Author Set Size and Data Size in Authorship Attribution. *Literary and Linguistic Computing*, volume 26/1.

Francesco Masulli and Giorgio Valentini. 2004. An Experimental Analysis of the Dependence Among Codeword Bit Errors in ECOC Learning Machines. *Neurocomputing*, volume 57.

Gilad Mishne. 2005. Experiments with Mood Classification in Blog Posts. *Proceedings of the 1st Workshop on Stylistic Analysis of Text for Information Access*.

Vassiliki Rentoumi, 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. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Fabrizio Sebastiani. 2002. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, volume 34/1.

Terrence J. Sejnowski and Charles R. Rosenberg. 1987. Parallel Networks that Learn to Pronounce English Text. *Complex Systems*.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453-1484 (2005).

Frederik Vaassen and Walter Daelemans. 2010. Emotion Classification in a Serious Game for Training Communication Skills. *Computational Linguistics in the Netherlands 2010: selected papers from the twentieth CLIN meeting*.

Antal van den Bosch. 2004. Wrapped Progressive Sampling Search for Optimizing Learning Algorithm Parameters. *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC2004)*.

Antal van den Bosch, Bertjan Busser, Walter Daelemans and Sander Canisius. 2007. An Efficient Memory-based Morphosyntactic Tagger and Parser for Dutch. *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting (CLIN17)*.

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

Bert van Dijk and Fenno Moes. 2005. *Het grote beïnvloedingsspel*. Thema.

Jerry S. Wiggins. 2003. *Paradigms of Personality Assessment*. Guilford Press.

Aijun Zhang, Zhi-Li Wu, Chun-Hung Li and Kai-Tai Fang. 2003. On Hadamard-Type Output Coding in Multiclass Learning. *Lecture Notes in Computer Science*, volume 2690/2003. Springer Berlin / Heidelberg.