Multimed Tools Appl (2013) 62:319–332
DOI 10.1007/s11042-011-0909-8

Actor level emotion magnitude prediction in text and speech

Ricardo A. Calix · Gerald M. Knapp

Published online: 29 October 2011

Abstract The digital universe is expanding at very high rates. New ways of retrieving and enriching text and audio content are required. In this work, a methodology for actor level emotion magnitude prediction in text and speech is proposed. A model is trained to predict emotion magnitudes per actor at any point in a story using previous emotion magnitudes plus current text and speech features which act on the actor's emotional state. The methodology compares linear and non-linear regression techniques to determine the optimal model that fits the data. Results of the analysis show that non-linear regression models based on Support Vector Regression (SVR) using a Radial Basis Function (RBF) kernel provide the most accurate prediction model. An analysis of the contribution of the features for emotion magnitude prediction is performed.

Keywords Artificial intelligence · Natural language processing · Machine learning · Multimedia semantic analysis · Affect detection · Speech processing

1 Introduction

According to Gantz and Reinsel [13], between 2010 and 2020, the amount of digital information available on the web will grow to around 35 trillion gigabytes or more than 40 times larger than it is today. Gantz and Reinsel [13] note that while the growth in digital information will be by a factor of 44, the growth in the number of qualified IT professionals that can process this data will be by a factor of 1.4 during the same period. This rapid growth in the "digital universe" will create new challenges related to information search, understanding, and use. As a result, new systems that can address these challenges such as

R. A. Calix · G. M. Knapp (✉)
College of Engineering, Louisiana State University, Baton Rouge, LA, USA
e-mail: gknapp@lsu.edu

R. A. Calix
e-mail: rcalix1@lsu.edu

understanding the complicated semantics of language in text and speech, or than can understand human emotions will need to be developed. Finding ways of detecting emotion in speech and other media could be very useful to improve multimedia content management, generation, access, and distribution. In this work, a methodology for actor level emotion magnitude prediction from text and speech media is proposed and tested.

Predicted emotion magnitudes can be used for many applications related to content search and generation. In content search, estimated emotion magnitudes could be used for automatic tagging or automatic rating of audio narratives based on emotion content. In content generation, emotion magnitudes could be used for automatic emotional speech synthesis (from text) and emotional facial expression rendering in 3-D graphics (from text and speech). Emotion magnitudes can be used to adjust pitch, rate or volume parameters in XML based schema such as Speech Synthesis Markup Language (SSML). The intensity adjustments can be used to control the intonation and pitch used to synthesize speech. In computer graphics, emotion magnitudes could be used as weights to adjust the emotion expression of 3-D character renderings. This approach can be implemented with the morph targets technique [1] in which vertex level weighted emotion meshes are added to a neutral mesh. In this case, the weight, which determines the level of interpolation between the target and neutral mesh, can be determined by the emotion magnitude predicted by the model.

Much work has been done in emotion detection for document summarization, opinion mining, and sentiment polarity detection. However, many issues related to how to estimate emotion magnitudes, and how to map these estimates to actors through a story in a text or audio collection are still unresolved. This work addresses the following issues: (1) how should actors be considered in emotion prediction? (2) How do you determine the emotional state of each actor throughout a story? (3) How do you identify emotion triggers in a story and their contribution (or weight) to emotion prediction? (4) How do you estimate the magnitude of emotion an actor experiences? The methodology is trained to predict emotion magnitudes per actor at any point in a story using previous emotion magnitudes plus current text and speech features which act on the actor's emotional state. Linear and non-linear regression techniques are used to find the optimal model that can fit the data. The Affect Corpus 2.0 [10] was extended and is used to train and test this model. This methodology builds on work by Calix et al. [9] and Alm [4].

2 Literature review

2.1 Emotion analysis

Studies in affect analysis include [2, 4, 8, 9, 17, 19, 20, 22, 24]. In Tokuhisa et al. [24], the authors propose a model for detecting the emotional state of a user that interacts with a dialog system. The method uses corpus statistics and supervised learning to detect emotion in text. A two-step approach is used where coarse grained emotion detection is performed first followed by fine grained emotion detection. Alm et al. [2] and Alm [4] developed and analyzed a new affect corpus based on children's stories. They used supervised learning techniques to perform sentence level emotion detection for use in text-to-speech synthesis. Calix et al. [9] proposes a methodology to automatically extract emotion relevant words from annotated corpora. The emotion relevant words are used as features in sentence level

emotion classification with Support Vector Machines (SVM) using 5 emotion classes and a neutral class.

In Moilanen and Pulman [19], the authors argue that emotion content in a sentence is based on the emotional content of its constituent parts. Therefore, the authors propose a series of sentiment composition rules which can be used to infer the emotional polarity of a sentence. Neviarouskaya et al. [20] also uses a semantic composition and a rules-based approach to detect emotion in text. Pang and Lee et al. [22] provides a survey of the current methodologies and issues related to opinion mining and sentiment analysis. In Busso et al. [8] the authors study the implications of speech features alone in emotion detection in audio databases. Their results indicate that utterance or sentence level pitch statistics are more accurate in emotion detection than pitch statistics for shorter speech segments such as words. Luengo et al. [17] perform an analysis of the speech features that are more important for emotion detection in audio files. They conclude that spectral level features outperform prosody features in emotion detection.

In Lu et al. [16] the authors consider the issue of who experiences an emotion by using a semantic role labeling tool which for each verb in a sentence identifies constituents with a semantic role such as patient or agent. This helps to find possible subjects and objects in a sentence. Once the subjects and objects in a sentence are identified, the system uses a web mining engine to find definitions for the subjects and objects which can be used to detect emotional content. Emotion magnitude estimation has been addressed by Grimm et al. [14]. In their work, the researchers analyzed the estimation of continuous values for 3 emotion class related properties or dimensions: valence, activation, and dominance. Valence represents the positive vs. negative relation of an emotion. Activation describes the level of excitation (from calm to excited) and dominance describes the influence of the person (from weak to dominant). Their study [14] focused on speech only features at the utterance level to estimate the emotion levels using several regression approaches. In general, the literature uses the big six emotion labels or 2–4 dimensional primitives (e.g. power, valence, activation, expectation) to annotate emotion. To be consistent with the initial UIUC annotations (Alm 2008), the big six emotion labels were used in this work.

Studies that have addressed the issue of state flow over time include [7, 12, 18]. In El-Nasr et al. [12], the authors developed PETEEI, a pet with evolving emotional intelligence. This system combined a fuzzy logic model with heuristic rules to simulate emotion in agents which depends on action sequences and rewards. The system can adapt emotional state and intensity based on probabilities of actions in a sequence (sequence mining) and user feedback. In Burns et al. [7] the authors propose the use of hierarchical Bayesian networks for adaptive reasoning over time. Mao et al. [18] conducted a study on the prediction of sentiment flow in documents using conditional random fields. The objective of their study was to predict a sequence of sentiments in a document based on a sequence of sentences. Results of their study indicate that sequential models are better than non-sequential models at predicting and describing sentiment.

Although addressed in some of the previous studies, several issues related to emotion analysis still require further research. These issues include: (1) how can emotion intensity or magnitude be predicted? (2) May the addition of acoustic features improve text-based detection of emotion intensity? And (3) how should sentiment flow per actor be accumulated through a story? These are all important issues that need to be addressed. The work presented in this paper proposes a methodology that addresses and integrates these issues into a model for actor level emotion magnitude prediction.

2.2 Features

Emotion detection in speech collections requires the use of both text and speech features. Text from the audio recording can be obtained by using an Automatic Speech Recognition (ASR) system which can convert the signal into a text transcript. Additional speech features such as pitch and MFCCs (Mel Frequency Cepstral Coefficients) can be obtained from the signal to obtain additional information about the intonation of what was said [15]. The state of the art in ASR technology has progressed to the point where good results in speech-to-text translation can be obtained. The results are usually measured in word error rate (WER) for certain domains. The commercially available Nuance Dragon software [21] is currently a leader in speech-to-text synthesis and achieves over 90% accuracy for one speaker speech transcriptions. These ASR systems rely on huge vocabularies to produce good transcriptions. The Dragon software can use vocabularies with more than 300,000 words.

Many works such as [8, 14, 17] argue that pitch related speech features such as pitch average, MFCCs, and formants can help to capture aspects of emotion in speech. The Affect Corpus 2.0 [10] will be used to obtain the text and speech features to train and test the model. This corpus contains audio and transcribed versions for 89 children's stories.

2.3 Machine learning

Common learning methodologies to address magnitude estimation issues include linear and non-linear regression models such Linear Regression, Artificial Neural Networks, and Support Vector Regression (SVR). For this work, Support Vector Regression [23], based on Support Vector Machines, was used. According to Smola et al. [23], the SVR methodology tries to obtain a regression function that is as flat as possible and with low prediction error. The difference between the predicted values and the actual values from the training set (predicted error) should be no more than a certain specified value (E). The "flatness" property of the regression function is achieved by minimizing the norm of the weight vector (w). A kernel trick may be used to map non-linear data to linear space. The soft margin optimization problem can be formulated as follows:

$$\text{Minimize} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \tag{1}$$

Subject to:

$$Y_i - w\cdot\phi(X_i) - b \le E + \xi_i \tag{2}$$

$$w\cdot\phi(X_i) + b - Y_i \le E + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0 \quad \text{for i} = 1 \ldots \text{n}$$

where C is the cost (tradeoff between w and prediction error), "i" is the sample, w are the weights, $\phi(\cdot)$is the high dimensional feature space function, $\xi_i$and $\xi_i^*$are the slack variables that allow for errors to occur during training, with E the tolerated error level. This optimization problem with constraints to find the weight vector (w) and the

bias (b) can be solved using Lagrange multipliers. The formulation is quadratic and can provide a single minimum solution using quadratic programming optimization. The fast LibSVM implementation [11] in conjunction with WEKA was used to train and test the SVR model.

3 Methodology

The methodology for predicting evolving emotional state of actors consisted of three main faces:

- Pre-processing of the input files.
- Extracting text and speech elements which include actors, environmental text elements, emotion triggers (i.e. emotion words, actions, environments), emotion signal features, and speech features in the training corpus.
- Training and testing of a regression model to predict emotion magnitudes per actor.

Changes in emotional states of actors are assumed to be caused by triggers (the occurrence of actions or new environmental elements that cause emotion changes). The magnitude of the change depends on the current emotion triggers and previous emotional state. For the purposes of this work, actors are defined as sentient entities (people and creatures) which demonstrate sentient behavior such as speaking, feeling, and thinking within the context of a conversation or story. In general, such entities can be found in noun phrases. The Affect Corpus 2.0 [3, 10] has been annotated with actors and their corresponding emotion magnitudes.

Since annotations were performed at the actor level [10], an instance in this work is associated with an actor that is present in a sentence. Therefore, an instance is a per actor feature vector. A window approach is used to extract the features where the window is the sentence that contains the actor. The actor's previous emotion magnitude for the given class is one feature, another feature determines subject/object relationship, 33 features are extracted from the speech segment that matches the sentence that contains the actor, 36 text based features (Table 1) are extracted from the sentence, and the rest are emotion words (1214), environment words (825), action words (1007).

3.1 Corpus

The Affect Corpus 2.0 [3, 10] is used as a base for this work. This corpus consists of 89 fairy tales by three authors which are the Brothers Grimm, H. C. Andersen, and B. Potter. Each of the 89 annotated stories in the corpus includes an audio recording and a text transcript. The audio versions are stored in MP3 format. These audio files were manually annotated and are used to extract speech features at the sentence level. The MP3 files for the stories provided in the Affect Corpus 2.0 were read by many speakers to avoid issues of dependency of emotions across speakers. In general, not more than 4 stories were recorded by one speaker and the number of females to males was almost evenly distributed.

The text files are stored as text files and are used to extract the semantic content from each story. The corpus includes annotations of the actors in each story and their presence in a given sentence. The annotated actors for each story are stored in a text file. The corpus also includes actor level emotion magnitude annotations to perform the training and testing of the prediction model. Therefore, the emotional states and magnitudes can be determined

Table 1 Text features for multimodal emotion classification

Number of features	Names	Type	Category	Heuristic rule for calculation
1	Contains NNP	Text	NNP binary (Syntactic)	If NNP in sentence: NNP=1
5	Counts for happy, sad, angry, afraid, surprised	Text	Number of words per emotion class	If word from sentence in happy words list: counthap+= 1 Note: same for other 4 emotion classes
1	Number of words in sentence	Text	Number of words in sentence	Number of words in sentence list
2	Subject actor/environment	Text	Metric for subjects	If VP not in BranchStack: IntSubj+= 1
2	Object actor/environment	Text	Metric for objects	If NP not in BranchStack: IntObj+= 1
1	Sentiment composition current sentence	Text	Sentiment composition	If word from sentence in happy or surprised words list: SentCompCurr+= 1 Negative_list=[sad, angry, afraid] If word from sentence in Negative_list: SentCompCurr -=1
5	Sentiment composition accumulated whole story for happy, sad, angry, surprised, and afraid	Text	Sentiment flow	SentCompAccumHappy= SentCompAccumHappy+counthap Note: same for other 4 emotion classes
1	Sentiment composition change previous current accumulated	Text	Sentiment flow (delta)	SentCompChanPrevCurrAccum= SentCompAccumCombined - SentCompAccumPrevSent
4	Sentiment composition accumulated whole story for POS, NEG, POS1, NEG1	Text	Sentiment flow	CompAccumPos=CompAccumPos+ w * counthap CompAccumNeg= CompAccumNeg+w * (countsad+ countang+countafr) Note: w parameter changed manually
1	Sentiment composition accumulated whole story for the previous sentence	Text	Sentiment flow	SentCompAccumPrevSent (s)= SentCompAccumCombined (s-1) Note: s stands for sentence
1	Sentiment composition accumulated whole story combined	Text	Sentiment flow	SentCompAccumCombined= SentCompAccumCombined+ counthap – countsad – countang – countafr+countsup
5	Number of happy, sad, angry, afraid, and surprised phrases	Text	Emotion phrase count	If phrase from sentence in happy phrases list: PhraseCountHap+= 1 Note: same for other 4 emotion classes
2	Intensify emotion, reduce emotion	Text	Intensifier or reducer features (e.g. very)	If word in sentence is in list of intensifiers: IntensifyEmt+= 1 Note: same approach for reducer
5	Changes in happy, sad, angry, surprised, and afraid	Text	Change in sentiment flow (delta)	ChangeHap=counthap – PreviousHap PreviousHap=counthap Note: Same for 4 other classes

at the actor level. Magnitudes were annotated for 5 emotion classes: happy, sad, angry, afraid, and surprised. Neutral is represented when the emotion magnitude from all emotion classes is set to zero. The rating scale consisted of values from 0 to 100 where 0 represent no emotion and 100 is the maximum amount of the given emotion [10]. A sample of 19 stories from the corpus were double annotated and measured for inter-annotator agreement. The results of this analysis are discussed in [10].

3.2 Emotion triggers for magnitude prediction

A regression approach to predict actor level emotion magnitudes requires combining emotion trigger features in a linear or non-linear model. The following 5 sets of triggers are used as regression factors:

(1) A set of Emotion Tokens (emo) which consists of emotion strong nouns, adjectives, and adverbs (for example, happy, happiness, etc.). These emotion words were collected manually and expanded using WordNet and ConceptNet. These words were manually classified into 5 emotion subsets. A total of 1214 stemmed emotion words were used as Emotion Token features.
(2) A set of Environment tokens (env) which consists of 825 stemmed environment words. These words consist of mainly nouns of places and other settings were an actor might be located. The list of environments was manually collected using noun phrases from the corpus and external sources from the internet. The initial words list was extended using ConceptNet's at location relation property. Examples of environment tokens include road, forest, farm, land, storm, etc.
(3) A set of Action Tokens (act_s, act_o) which consist of 1007 stemmed verbs. For each actor in a sentence, the sentence dependency parse was used to identify if the actor was the subject or object of the verb. Action token examples include words such as kill, attack, stuck, etc.
(4) A set of 36 text based emotion signal features (EmSin). These features consist of counts of the number of emotion words that appear in a sentence per emotion class or polarity. These counts can be traced through a story to see how emotion intensity varies between positive, negative and between emotion classes. These features use sentiment composition principles to measure the changes and distribution of emotion in sentences. A subset of the 36 features is used to accumulate the emotion intensity from sentence to sentence through a story and to compare the difference in sentiment flow (e.g. the difference in the counts between the current and previous sentence).
(5) A set of 33 sentence level speech features (Speech) which include max and average F0, max and average intensity, formants (F1, F2, F3, F4, F5), and the mean and standard deviation for 12 Mel Frequency Cepstral Coefficients (MFCCs). F0 is the pitch or the "mental sensation of fundamental frequency" [15]. Formants are amplitude peaks in a given frequency range that can be correlated to given sounds [15]. MFFCs are a representation of the signal after taking two Fourier transforms and after mapping the original signal to the Mel scale of the human auditory signal. Since humans cannot perceive all sound frequencies, this mapping helps to better capture speech signal differences [15].

The text features are extracted using python scripts (Table 1), the Stanford parser, and NLTK [5]. All word tokens were stemmed using the Porter Stemmer to reduce dimensionality of the token sets. Speech features are extracted using Praat scripts [6]. All features are combined into feature vectors per actor.

Table 1 shows the features that help to capture sentiment flow and how they are computed. Additionally, since the approach uses previous magnitudes, this feature also helps to capture sentiment flow.

3.3 Prediction model

The dependent variables for the model are the 5 emotional magnitudes for happy, sad, angry, surprised, and afraid. Each of the 5 emotion magnitudes is allowed to range from 0 (neutral) to 1 (maximum). Once the model is trained, the emotion magnitude for each actor can be calculated as the previous sentence's emotion magnitude plus a linear or non-linear weighted sum of the trigger tokens present in the current sentence and associated with the actor. Formally, the model is presented in Eq. 3.

$$E_{d[s],a} = E_{d[s-1],a} + A_{d[s],a} * Q \tag{3}$$

$$Q = [P^{emo}_{d[s]} \cdot \alpha + P^{env}_{d[s]} \cdot \beta + P^{act\_s}_{d[s],a} \cdot \gamma + P^{act\_o}_{d[s],a} \cdot \theta + P^{sp}_{d[s]} \cdot \upsilon + P^{EmSin}_{d[s]} \cdot \psi] \tag{4}$$

Where:

$E_{d[s],\ a}$	is a row vector of emotion magnitudes (1 column for each emotion) for actor "a" in sentence s of document d (d[s-1] indicates previous sentence in same document).
$A_{d[s],\ a}$	is a scalar value=1 if actor "a" is physically present in sentence s of document d, and 0 otherwise. α is a matrix of emotion token weights (5 columns, one for each emotion, by N_{emo} emotion rows where N_{emo} is the total number of emotion tokens in the corpus).
β	is a matrix of environmental token weights (5 columns by N_{env} rows).
γ	is a matrix of subject action token weights (5 columns by N_{act_s} rows).
θ	is a matrix of object action token weights (5 columns by N_{act_o} rows).
υ	is a matrix of the speech feature weights.
ψ	is a matrix of the emotion signal (EmSin) weights.
$P^{emo}_{d[s]}$ and $P^{env}_{d[s]}$	are 0/1 row vectors whose elements indicate whether the corresponding token (emotion and environment, respectively) is present (1) or not (0) in sentence s of document d.
$P^{act\_s}_{d[s],a}$ and $P^{act\_o}_{d[s],a}$	are 0/1 row vectors whose elements indicate whether the corresponding action token (with the actor "a" as subject or object, respectively) is present or not in sentence s of document d.
$P^{Sp}_{d[s]}$ and $P^{EmSin}_{d[s]}$	are vectors for speech features and text emotion signal measurements (EmSin), respectively.

For the first sentence of a document, $E_{d[s-1]}=0$, the zero vector (vector of zero values), corresponding to a neutral emotion state. The weights correspond to emotion magnitude changes (deltas) to be applied for each emotion if the trigger is present in a sentence.

In conclusion, the predicting model is a regression equation where the goal is to find the weights for each of the variables. The equation includes variables for speech features, text based features (Table 1), tokens (e.g. the presence of emotion words, environment words and action words), and 1 variable for that actor's previous emotion magnitude for the given class magnitude being predicted. Each of these variables is combined in a linear sum to find the predictive equation. Therefore, the given weights for each of the variables must be

found using the training data from the corpus. The following section details the methodology for learning these weights.

3.4 Training

In this work, a regression/optimization approach is used to learn the weights to assign to each trigger/emotion pairing. Training is performed using linear and non-linear regression (e.g. SVR) approaches.

3.4.1 Linear regression approach

The linear regression is performed over all stories in the corpus with the goal of minimizing the sum of squared error terms between the corpus emotion annotation and the calculated emotions across all actors in all sentences of all corpus stories. The weights are then applied during testing/application to calculate the emotional magnitude of each actor at each sentence.

Decision Variables: Token weight matrices α, β, γ, θ, ψ, υ
Objective Function: Min $z = \Sigma_{d[s],a \in c} \varepsilon^2_{d[s],a}$
where $\varepsilon_{d[s],a}$= error term for actor a, sentence s in document d; and c=corpus
subject to constraints:

$$E_{d[s],a} = E_{d[s-1],a} + A_{d[s],a} * Q \quad \forall \mathrm{d[s]}, \mathrm{a} \in \mathrm{c} \tag{5}$$

$$\varepsilon_{d[s],a} = E_{d[s],a} - E'_{d[s],a} \quad \forall \mathrm{d[s]}, \mathrm{a} \in \mathrm{c} \tag{6}$$

where $E'_{d[s],a}$is the annotated emotion level from corpus c for actor "a" in sentence s of document d. Additionally, each weight element of the matrices α, β, γ, θ, υ, ψ is constrained to be between −1 and +1.

3.4.2 Non-Linear approach: Support Vector Regression

The optimization model formulated with Eqs. 1 and 2 produces the weight vector (w) and bias (b) and the regression equation becomes:

$$g(x)_i = \sum_{i=1}^{nv} (\lambda_i - \lambda_i^*) K(x_i, x) + b \tag{7}$$

where $\mathrm{K}(\mathrm{x_i}, \mathrm{x}) = \phi^T(x_i) \cdot \phi(x)$is the kernel function that maps the input vector to higher dimensional space, $\lambda_i$and $\lambda_i^*$are the LaGrange multipliers, and nv is the number of support vectors.

4 Analysis and results

To determine the contribution of semantic information in emotion prediction, the model is trained and analyzed using 2 different approaches. The first approach (Table 2) uses all text and speech features described in section 3.2. The second approach (Table 3) uses speech features only to train the prediction model. Therefore, a total of 3134 features are used for

Table 2 Regression Modeling Results (all features & previous magnitude)

Regression Modeling – Speech and Text (Train: 80%; Test: 20%)									
	SVR (Linear)		SVR (Polynomial)		SVR (RBF)		Linear Regression		Total
	Corr. Coef.	RMSE	Corr. Coef.	RMSE	Corr. Coef.	RMSE	Corr. Coef.	RMSE	
Happy	0.66	17.74	0.43	23.83	0.76 g=0.15 c=120	15.29	0.61	20.13	2640
Sad	0.73	18.33	0.65	27.22	0.77 g=0.07 c=20	17.32	0.59	25.10	1590
Angry	0.78	15.79	0.55	25.65	0.80 g=0.05 c=60	15.44	0.00	25.72	1192
Surp.	0.64	16.91	0.34	22.09	0.69 g=0.1 c=90	16.05	0.52	20.366	3551
Afraid	0.73	18.04	0.60	26.71	0.79 g=0.1 c=200	16.34	0.66	21.07	2693

Legend: Corr. Coef. = Correlation Coefficient RMSE: Root Mean Squared Error

the "all features" approach and 33 features are used for the "speech only" approach. Both approaches include 1 additional feature for the actors' previous emotion magnitude in the story as indicated in the model (Eq. 3). Training and testing are performed on each subset using an 80%/20% split. The prediction model is trained and tested using linear regression, multilayer perceptron (MLP), and Support Vector Regression (SVR) techniques.

The SVR is trained using the linear, polynomial, and RBF (Radial Basis Function) kernels. Results of the model accuracy are evaluated and compared using Root Mean Square Error (RMSE) and the correlation coefficient. In total, 5 prediction equations are trained and tested (one for each of the 5 emotion classes: happy, sad, angry, afraid, and

Table 3 Regression Modeling Results (speech features only & previous magnitude)

Regression Modeling—Speech (Train: 80%; Test: 20%)									
	SVR (Linear)		SVR (Polynomial)		SVR (RBF)		Linear Regression		Total
	Corr. Coef.	RMSE	Corr. Coef.	RMSE	Corr. Coef.	RMSE	Corr. Coef.	RMSE	
Happy	0.46	21.34	0.43	23.69	0.66 g=0.3 c=12000	19.06	0.48	20.50	2640
Sad	0.63	21.98	0.62	21.37	0.69 g=0.07 c=10000	20.11	0.63	21.25	1590
Angry	0.72	18.4	0.74	17.45	0.78 g=0.1 c=11000	16.40	0.73	17.42	1192
Surp.	0.61	18.06	0.65	16.92	0.68 g=0.1 c=6000	16.34	0.61	17.00	3551
Afraid	0.67	19.89	0.65	26.37	0.76 g=0.8 c=240	17.51	0.68	19.23	2693

Legend: Corr. Coef. = Correlation Coefficient RMSE: Root Mean Squared Error

Table 4 Significance testing comparison between "all features" and "speech only"

	SVR-RBF (All features & previous magnitude)	SVR-RBF (Speech features & previous magnitude)	SVR-RBF (Text features & prev. magnitude)
Happy (Corr. Coef.)	0.77	0.66*	0.77
Happy (RMSE)	14.83	18.75*	14.93
Sad (Corr. Coef.)	0.78	0.71*	0.78*
Sad (RMSE)	17.38	20.15*	17.67*
Angry (Corr. Coef.)	0.78	0.73*	0.78
Angry (RMSE)	16.60	18.64*	16.82*
Surp. (Corr. Coef.)	0.71	0.69	0.70
Surp. (RMSE)	16.14	16.69	16.23
Afraid (Corr. Coef.)	0.79	0.76*	0.78*
Afraid (RMSE)	15.94	17.11*	16.33*

Legend: v=statistically better *=statistically worst

surprised). Of the 3 models used, the SVR with an RBF kernel performed the best for both accuracy and speed. This suggests that the data is not linear and that a non-linear approach is required. The model with the worst performance was the multilayer perceptron because of the time required to train the model and high RMSE. The root mean squared errors (RMSE) for the MLP-PCA or MLP-SO (speech only) fell between 21.71 and 36.61. The SVR-RBF model performed the best overall. The SVR correlation coefficients using an RBF kernel fell between a range of 0.69 and 0.80 (Table 2). This result indicates that the model is able to get a good correlation between the features and the predicted magnitudes. The RMSE using the SVR with RBF kernel fell between 15.29 and 17.32. These error results are much lower than the ones obtained with the MLP and the best overall.

The linear regression model performed between the SVR and the MLP-PCA. Feature analysis using WEKA's ReliefF feature ranking [25] indicated that the feature with the highest contribution to emotion prediction accuracy was the actors' previous emotion magnitude. This is an important result which suggests that overtime the accuracy of the model can improve and that previous information about the emotional state of the actor is very important. From the comparison between the "all features" approach (Table 2) and the "speech only" approach (Table 3) it can be seen that a multimodal approach using both text and speech features helps to improve prediction accuracy. This result shows the importance of using higher level semantic approaches.

Table 5 Regression analysis with PCA (95%) and SVR-RBF

	Regression Modeling—All features (Train: 80%; Test: 20%)		
	Corr. Coef.	RMSE	Number of PCAs
Happy	0.67	17.6	627
Sad	0.54	24.2	530
Angry	0.55	21.7	472
Surprised	0.57	18.2	796
Afraid	0.66	20.1	669

Legend: Corr. Coef. = Correlation Coefficient RMSE: Root Mean Squared Error

To address the issue of long processing time caused by the high dimensionality of the feature set, the data dimensionality was reduced using Principal Component Analysis (PCA). PCA dimensionality reduction was performed and evaluated on all 3 models (linear regression, multilayer perceptron, and SVR). The prediction results after PCA for all approaches were worse than the results using all features without dimensionality reduction. The dimensionality reduction for each of the 5 emotion classes was as follows: happy (628), sad (531), angry (473), surprised (797), and afraid (670).

To measure the significance of the difference between the approaches, the two best methods using "all features" and "speech features only" were tested using WEKA's experimenter module. Testing was done with the Paired *T*-Test [25] with a significance level of 0.05 (two tailed). The experiment type used 10 fold cross-validation with 10 repetitions. Both correlation coefficient and RMSE were tested for each of the five emotion classes. The results are presented in Table 4.

All analysis performed in this work was done on a computer with 6 GB of memory, and an Intel i3 core. The operating system environment is Windows 7. The scripts for Praat and python can be used in both Linux and Windows environments. Finally, since the variables could be correlated, the regression analysis using PCA and SVR-RBF is performed on the "all features" dataset as well. The principal components obtained from PCA are a set of transformed features which will be un-correlated. The results are presented in Table 5.

5 Conclusion and future work

In this work, a model was proposed and tested to predict emotion magnitudes for actors in a story. Text and speech features were used and compared. Overall, the model was able to learn and achieved good prediction results. Of the 3 machine learning algorithms used to train the model, the support vector regression technique with an RBF kernel performed the best overall. The SVR correlation coefficients using an RBF kernel fell between a range of 0.69 and 0.80. This result indicates that the model is able to get a good correlation between the features and the predicted magnitudes. The best RMSE results fell in the range of 15.29 to 17.32 and were obtained using the SVR with RBF kernel. These error results are much lower than the ones obtained with the MLP and the best overall.

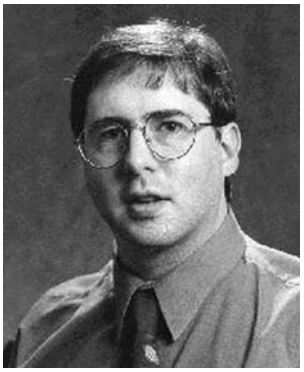
Feature analysis indicated that the feature with the highest contribution to emotion prediction accuracy was the actors' previous emotion magnitude. This is an important result which suggests that overtime the accuracy of the model can improve and that previous information about the emotional state of the actor is very important. From the comparison between the "all features" approach and the "speech only" approach it can be seen that a multimodal approach using both text and speech features helps to improve prediction accuracy. This result shows the importance of using higher level semantic approaches. The prediction model combined with the morph targets technique [1] can provide a powerful tool to automatically render emotional facial expressions and gestures from emotion detection in text and speech.

Future work will focus on applying the technique in human computer interaction settings and in healthcare communication dialogue systems. Predicted emotion magnitudes in healthcare systems can be used for embodied conversational agents. The predicted magnitudes can be mapped to facial expression parameters using mesh morphing so that a virtual agent's face is adjusted based on the predicted emotion magnitudes. Additional detail in areas such as sentient actor detection, anaphora resolution, and speech feature analysis will be explored. Finally, emotion expression renderings produced by the predicted emotion magnitudes will be evaluated in user rating studies.

References

1. Akenine-Moller T, Haines E, Hoffman N (2008) Real-Time Rendering. A K Peters, Wellesley, Massachusetts
2. Alm CO, Roth D, Sproat R (2005) Emotions from text: machine learning for text-based emotion prediction. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 579–586
3. Alm CO (2011) Affect data. http://lrc.cornell.edu/swedish/dataset/affectdata/index.html. Accessed 30 March 2011
4. Alm CO (2008) Affect in Text and Speech. Dissertation, University of Illinois at Urbana-Champaign
5. Bird S, Klein E, Loper E (2009) Natural Language Processing with Python. 1st ed., O'Reilly Media
6. Boersma P, Weenink D (2011) Praat: doing phonetics by computer. Version 5.2.21. http://www.praat.org/. Accessed 30 March 2011
7. Burns B, Morrison C (2003) Temporal Abstraction in Bayesian Networks. In Working Notes of Association for the Advancement of Artificial Intelligence (AAAI), Spring Symposium Workshop: Foundation and Applications of Spatio-Temporal Reasoning, AAAI,Technical Report SS-03-03, 2003
8. Busso C, Lee S, Narayanan S (2009) Analysis of emotionally salient aspects of fundamental frequency for emotion detection. IEEE Transactions on Audio, Speech, and Language Processing Vol. 17, No. 4
9. Calix RA, Mallepudi S, Chen B, Knapp GM (2010) Emotion recognition in text for 3D facial expression rendering. IEEE Transactions on Multimedia, Special Issue on Multimodal Affective Interaction 12 (6):544–551
10. Calix RA, Knapp GM (2011) Affect Corpus 2.0: An extension of a corpus for actor level emotion magnitude detection. In Proceedings of the 2nd ACM Multimedia Systems (MMSys) conference, Feb. 2011, San Jose, California, U.S.A.
11. Chang CC, Lin C (2001) LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm. Accessed 30 March 2011
12. El-Nasr M, Loerger T, Yen J (1999) PETEEI: A pet with evolving emotional intelligence. Proceedings of the third annual conference on autonomous agents, Seattle, Washington, USA, pp. 9–15
13. Gantz J, Reinsel D (2010) The digital universe decade—Are you ready? IDC Report. http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm. Accessed 30 March 2011
14. Grimm M, Kroschel K, Narayanan S (2007) Support Vector Regression for automatic recognition of spontaneous emotions in speech. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), Vol. 4, pp. IV-1085-IV-1088
15. Jurafsky D, Martin J (2008) Speech and Language Processing, 2nd edn. Prentice Hall, New Jersey
16. Lu CY, Hong J, Cruz-Lara S (2006) Emotion detection in textual information by semantic role labeling and web mining techniques. Third Taiwanese-French Conference on Information Technology—TFIT
17. Luengo I, Navas E, Hernaez I, (2010) Feature analysis and evaluation for automatic emotion identification in speech. IEEE Transactions on Multimedia, Vol. 12, No. 6
18. Mao Y, Lebanon G (2006) Sequential models for sentiment prediction. In Proceedings of the International Conference on Machine Learning (ICML), Workshop on Learning in Structured Output Spaces, Pittsburg, PA
19. Moilanen K, Pulman S (2007) Sentiment Composition. In Proceedings of Recent Advances in Natural Language Processing (RANLP 2007), September 27–29, Borovets, Bulgaria, pp. 378–382
20. Neviarouskaya A, Prendinger H, Ishizuka M (2009) Semantically distinct verb classes involved in sentiment analysis. In Proceedings IADIS international conference on applied computing, AC 1:27–35
21. Nuance (2011) Naturally speaking software. http://www.nuance.com/dragon/index.htm. Accessed 30 March 2011
22. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1–2):1–135
23. Smola A, Scholkopf B (2004) A tutorial on Support Vector Regression. Stat Comput 14:199–222
24. Tokuhisa R, Inui K, Matsumoto Y (2008) Emotion classification using massive examples extracted from the web. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, pp. 881–888
25. Witten I, Frank E (2005) Data Mining: Practical Machine Learning Tools and Techniques, 2 dth edn. Morgan Kaufmann Publishers Inc., San Francisco

Ricardo A. Calix received the B.S. degree in industrial and systems engineering from the Universidad Tecnologica Centroamericana, Tegucigalpa, Honduras, in 2001 and an M.B.A. from Louisiana State University (LSU), Baton Rouge, in 2006. He received the M.S. and Ph.D. degrees in engineering science with concentration in information technology and engineering from Louisiana State University in 2010 and 2011, respectively. His research interests include human computer interaction, semantic analysis and natural language processing, and machine learning.



Gerald M. Knapp received the B.S. and M.S. degrees in industrial engineering from the State University of New York, Buffalo, in 1987 and 1989, respectively, and the Ph.D. degree in industrial engineering from the University of Iowa, Iowa City, in 1992. He is Fred B. & Ruth B. Zigler Associate Professor of Engineering at Louisiana State University, Baton Rouge. His research interests include semantic analysis and natural language processing, information systems, and human computer interaction.