Affect extraction using aural, visual and linguistic features from multimedia documents

Nikos Malandrakis

Department of Electronic and Computer Engineering

Technical University of Crete

A thesis submitted for the degree of

Master of Science

2012 February

1. Supervisor: Prof. Alexandros Potamianos

2. Reviewer: Prof. Vasileios Digalakis

3. Reviewer: Prof. Michail Zervakis

Day of the defense: 06/03/2012

Dedicated to my mother, Ioanna.

Acknowledgements

I want to express my appreciation to my supervisor Dr. Alexandros Potamianos for his guidance.

I am grateful to the members of the supervisory committee.

A big thank you to my friends, family and colleagues for their patience, love and support.

Finally, special thanks to M.Sc. candidate Maria Giannoudaki and Ph.D. candidate Iosif Elias, who both played major parts in the completion of the presented work.

Contents

List of Figures

List of Tables

Abstract

This work attempts to extract a viewer's emotion from the three modalities of a movie: audio, visual and text. A major obstacle for emotion research has been the lack of appropriately annotated databases, limiting the potential of supervised algorithms. To that end we develop and present a database of movie affect, annotated in continuous time, on a continuous valence-arousal scale. Supervised learning methods are proposed to model the continuous affective response using hidden Markov Models and low-level audio-visual features and classify each video frame into one of seven discrete categories (in each dimension); the discrete-valued curves are then converted to continuous values via spline interpolation. A variety of audio-visual features are investigated and an optimal feature set is selected. The potential of the method is verified on twelve 30-minute movie clips with good precision at a macroscopic level. This method proves not suitable to process subtitle information, so we explore the creation of a textual affective model, starting with a fully automated algorithm for expanding an affective lexicon with new entries. Continuous valence ratings are estimated for unseen words under the assumption that semantic similarity implies affective similarity. Starting from a set of manually annotated words, a linear model is trained using the least mean squares algorithm. The semantic similarity between the selected features and the unseen words is computed with various similarity metrics, and used to compute the valence of unseen words. The proposed algorithm performs very well on reproducing the valence ratings of the *Affective Norms for English Words (ANEW)* and *General Inquirer* datasets. We then use three simple fusion schemes to combine lexical valence scores into sentence-level scores, producing state-of-the-art results on the sentence rating task of the *SemEval 2007* corpus.

Chapter 1

Introduction

1.1 Movie modalities

This thesis details our work on emotion recognition aimed primarily at movies. The work can be split roughly into two parts. The first part focuses on extracting emotion from a movie's audio, with the video and subtitle modalities taking the back seat. To that end we developed a dataset containing clips from popular movies and annotated them in the appropriate way. We performed analysis and validation of this dataset and then performed supervised classification experiments on it. We selected the appropriate models and features and added video and subtitle features.

The inclusion of subtitle data proved more challenging and the second part of this work is dedicated to the extraction of affect from lexical units: words and sentences. Virtually all methods of text affect extraction work hierarchically, creating ratings for words, then merging them into sentences and beyond that into larger units. This is also the methodology we pursued: we devised a method of creating word ratings and combined those ratings into sentence ratings. These methods, though they did achieve very good results on every other task we applied them to, never worked well enough on subtitles to warrant the next logical step: exploration of fusion strategies.

1.2 Contribution

Contributions of this work:

- The first ever movie database with continuous time and scale affective annotations, created as part of a bigger project in collaboration with the CVSP lab at the National Technical University of Athens.
- The first ever experiments on supervised emotion tracking, using the above database.
- A new and completely general method of affective lexicon expansion.
  - No specialized resources (ontologies) or human intervention (seed word selection) are required: only a starting lexicon and a large corpus (web), making it infinitely generalizable and particularly well suited to application in languages other than English.
  - It can be adapted so as to be effective regardless of available resources, corpus nature, type of queries etc.
- An extremely large lexicon containing over 120,000 words, created with the above method.

1.3 Publications

This work has, so far, resulted in three published conference papers:

1. Nikos Malandrakis, Alexandros Potamianos, Giorgos Evangelopoulos, Athanasia Zlatintsi, *"A supervised approach to movie emotion tracking"*, Proceedings of ICASSP, May 2011
2. Nikos Malandrakis, Alexandros Potamianos, Elias Iosif, Shrikanth Narayanan, *"Kernel models for affective lexicon creation"*, Proceedings of Interspeech, August 2011
3. Nikos Malandrakis, Alexandros Potamianos, Elias Iosif, Shrikanth Narayanan, *"EmotiWord: Affective Lexicon Creation with Application to Interaction and Multimedia Data"*, MUSCLE International Workshop on Computational Intelligence for Multimedia Understanding, December 2011

We are also in the process of authoring a journal article.

1.4 Outline

The organization of this thesis a slightly unconventional: it is structured as two papers, mostly because it is actually composed of a union and expansion of our published and future papers.

Chapter 2 details the part of our work that targeted movies specifically, including all experiments using audio-visual information as well as the initial experiments in including subtitle information, that confirmed it would require significantly more effort.

Chapter 3 details the part of our work that targeted lexical units, starting from words, moving on to sentences and finally the subtitles of our movies.

Each chapter is self-contained, with it's separate sections on prior work, experimental procedure and analysis.

Chapter 4 brings together the conclusions from both parts of our work and looks at the future work required to proceed in each aspect individually, as well as merge the findings into a three-modality model for movie affect.

Finally, the appendices include results from experiments run along the way, including the analysis of movie affective annotations, an attempt to identify relations between cause and effect (event and affect) and some extra (failed) audio-visual experiments.

Chapter 2

Movie Affect[*]

2.1 Introduction

Emotion recognition has been a very active field in the past years, since emotional information is highly valuable in applications ranging from human-computer interaction to automated content delivery. Emotion is of particular interest to content delivery systems that provide personalized multimedia content, automatically extract highlights and create automatic summaries or skims. The motivation behind using such technology is simple; humans pick content (movies, music) based on its affective characteristics, therefore a system designed to deliver it should have access to such data. Furthermore, systems aimed at highlight extraction/summarization require detailed representations of emotion in a scalable domain, as well as, information of the temporal dynamics of emotion. The process of extracting such information is usually referred to as *emotion tracking* and it is, ideally, a continuous-time continuous-scale representation of the affective content of a movie. A suitable continuous-scale representation is the dimensional representation of valence-arousal, shown in Fig 2.1 This two-dimensional representation is becoming increasingly popular due to its flexibility and high descriptive power, but also because the representation of emotion in a Euclidean space allows for simpler general-purpose analysis and recognition algorithms. In addition to the two-dimensional valence-arousal model, the three-dimensional valence-arousal-dominance model (or valence-arousal-tension for music) is also popular. In the field of affective multimedia content analysis it has been shown that the two-dimensional model is ade-

[*] Parts of this chapter have appeared in [33]

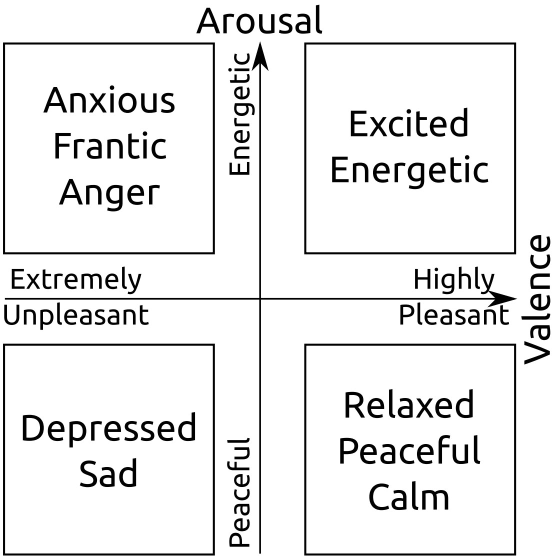


Figure 2.1: Illustration of the 2-D dimensional effect model.

quate to represent the range of emotions experienced by viewers/listeners [14]. Adding time as a third dimension, the affective content is represented as two continuous signals, the combination of which can yield an emotional state at any point within a multimedia stream.

There has been very little prior work towards emotion tracking in movies [21], with most researchers instead focusing on the more typical target of classifying large movie segments to a small number of distinct categories [27]. In all cases research has focused in narrow domains, such as specific movie genres [59]. To our knowledge, there has never been an attempt to apply supervised learning techniques to continuous-time emotion tracking in movies. A variety of models have been used to classify affective content, including Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) and Neural Networks (NNs). The features used are inspired by the ones used to characterize the modalities that make up a movie; timbre and rhythm to characterize music [4], color and motion to characterize video [60], energy, short-time spectral envelope and prosodic features to characterize speech [55].

One of the most important obstacles facing research in movie emotion and more particularly emotion tracking is the lack of movie databases annotated in an appropriate fashion, which probably explains the limited use of supervised techniques. As such, one of our targets was the creation of such a database, containing emotional responses annotated as continuous curves. Section 2.2 describes the creation of such a movie database of affect. In Section 2.3, we implement supervised learning techniques to train a classifier based on HMMs in order to perform emotion tracking, using a variety of audio-visual features. Experimental results are presented in Section 2.4 and conclusions in Section 2.5.

2.2 Database

Before describing the database, it is important to distinguish between three different "types" of movie emotion; *intended*, *expected* and *experienced* emotion. Intended emotion describes the emotional response that the movie attempts to evoke in its viewers, experienced emotion describes the emotion a user actually feels when watching the movie, while expected emotion is the expected value of experienced emotion in a population. Some prior research has assumed that intended and expected emotion match [21], however it is easy to see that a movie can be unsuccessful in conveying the desired effect. In fact the degree of effectiveness with which a movie creates the desired emotion in the viewer is a basic criterion humans use to assess movie quality. Our system attempts to predict intended emotion, however expected emotion is also desirable, since it can potentially be used as a basis for personalized predictions of experienced emotion [57]. This distinction is important for movie selection and annotating procedure definition.

2.2.1 The data

This emotional database was created as part of a larger project aiming at annotating movie data with affective, sensory and semantic cues. This is a joint project developed by the Technical University of Crete and the National Technical University of Athens, designed to be used by movie summarization systems such as that described in [17]. The database consists of contiguous thirty-minute video clips from twelve movies, featuring their visual, aural and textual data (subtitles). The movies selected are the ten winners of the Academy Award for best picture for the years 1998-2007 and two award winning animation films, namely; "Shakespeare in Love", "American Beauty", "Gladiator", "A Beautiful Mind", "Chicago", "The Lord of the Rings: The Return of the King", "Million Dollar Baby", "Crash", "The Departed", "No Country for Old

Men", "Ratatouille" and "Finding Nemo". Using the Academy Award winners list is one way of ensuring the high quality of the movies by a well-acknowledged criterion. One expected effect of this perceived quality is the higher correlation between intended and expected emotion; a high quality movie is expected to be successful in creating the desired emotional experience.

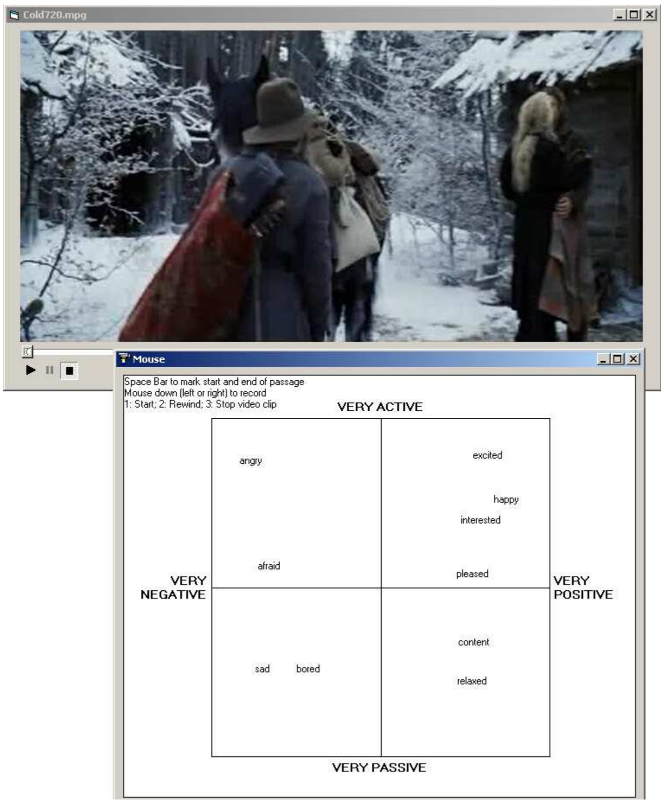
2.2.2 Annotating Procedure

Annotation was performed on two levels; intended emotion was annotated by experts, while volunteers annotated their individual experienced emotion, from which we derive the expected emotion. The annotations were performed using the FEELTRACE [13] emotion annotation tool. The participants track the annotated emotional response by moving the mouse pointer on a square two-dimensional area representing the valence-arousal emotional space, in real-time as they were watching the movie. The user interface is shown in Fig 2.2. So far seven volunteers, 20-30 years old, two female and five male have performed the annotation of experienced emotion. All annotators evaluated all clips, with five (out of seven) performing the entire process twice for intra-annotator agreement validation. Furthermore, annotators were presented with their results (curves) and their interpretation in textual terms in order to validate them and filled a questionnaire (shown in Fig 2.3) containing questions regarding their prior knowledge of the movies, their opinion of the movies and clips in regards to informativeness and enjoyability, their own annotating performance and their own perception of some suspected phenomena. Expected emotion is derived from the individual experienced emotion annotations using a correlation-based rejection scheme similar to that in [20] with particularly uncorrelated annotations being rejected as outliers. An example of the iterative rejection process is shown in Fig 2.4.

Validation of the database was done via analyzing the disagreement between users as well as between the users and the intended emotion against the factors suspected of leading to such disagreement from their answers to our questionnaires.

2.2.3 Annotation Results

The result of each annotation is a pair of curves, one curve for arousal and one for valence. These curves have values in the range $[-1, 1]$ for each dimension and are down-sampled to match the video rate of 25 fps. Overall, including duplicates, 144



Space Bar to mark start and end of passage
Mouse down (left or right) to record
1: Start; 2: Rewind; 3: Stop video clip

Figure 2.2: FeelTrace interface.

- before the annotation
  - had you watched the movie? *(binary)*
  - if yes
    - how many times? *(number)*
    - was the last time relatively recently? *(binary)*
    - did you recall a lot of details? (such as the order of scenes) *(binary)*
    - did you recall at least basic elements? (such as the identity of the hero) *(binary)*
    - did you enjoy it? *(binary)*
    - are you bored of it? *(binary)*
  - if not
    - do you believe the clip familiarized you regarding basic plot elements? *(binary)*
- first annotation
  - did you experience any difficulty in following the plot? *(binary)*
  - was there significant time delay until you understood what was happening so you could enter it in feeltrace? *(binary)*
- second annotation
  - how many days passed between the first and second annotations? *(number)*
  - was there a significant change in your opinion of the clip's contents, now having a more complete understanding than the first time? *(binary)*
  - did you observe your opinion of some scene to be affected by the knowledge of what will happen next? *(binary)*
- overall
  - did you enjoy the clip? *(binary)*
  - were you bored by the clip? *(binary)*

Figure 2.3: The questions answered by our participants

STEP 1

	$user_1$	$user_2$	$user_3$	$user_4$	$user_5$
$user_1$	1	0.11	0.63	0.43	-0.28
$user_2$	0.11	1	0.01	0.21	0.36
$user_3$	0.63	0.01	1	0.5	-0.39
$user_4$	0.43	0.21	0.5	1	0.11
$user_5$	-0.28	0.36	-0.39	0.11	1
average	0.22	0.17	0.19	0.31	-0.05

$\Longrightarrow$ $user_5$ rejected

STEP 2

	$user_1$	$user_2$	$user_3$	$user_4$
$user_1$	1	0.11	0.63	0.43
$user_2$	0.11	1	0.01	0.21
$user_3$	0.63	0.01	1	0.5
$user_4$	0.43	0.21	0.5	1
average	0.39	0.11	0.38	0.38

$\Longrightarrow$ $user_2$ rejected

STEP 3

	$user_1$	$user_3$	$user_4$
$user_1$	1	0.63	0.43
$user_3$	0.63	1	0.5
$user_4$	0.43	0.5	1
average	0.53	0.57	0.47

$\Longrightarrow$ process finished

Figure 2.4: Rejection process example. The tables show pair-wise correlation coefficients and their averages, which are used to select which will be rejected.

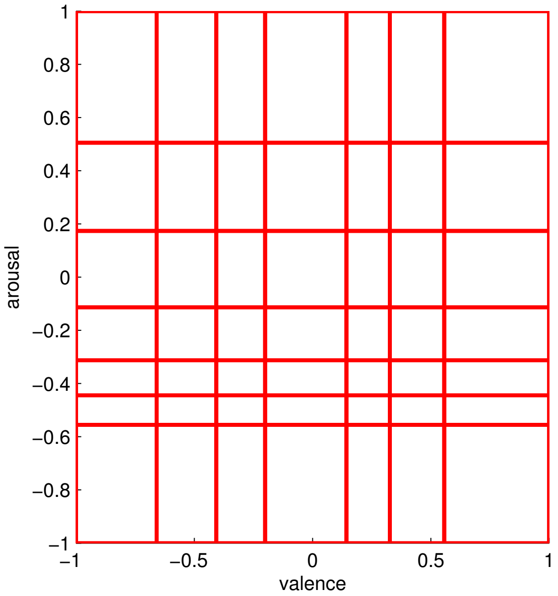


Figure 2.5: Quantization boundaries.

annotations of the experienced emotion and 36 annotations of intended emotion were produced, from which twelve annotations of expected emotion and twelve annotations of intended emotion, one of each per movie clip, were created. Fig 2.6 shows two-dimensional histograms of our annotations for intended and expected emotion. The "V" shape is very similar to that shown in [14] and [21] regarding the response to emotional media, which is reasonable given the similar context. Fig 2.7 shows some sample frames taken from the extremes of the two emotional dimensions. Table 2.1 shows agreement statistics in the annotations of experienced emotion. The low agreement is expected, since the participants annotate their own, very subjective, affective response. It is worth comparing these statistics between the two dimensions; distance metrics score higher for valence, while correlation is higher for arousal. That means that agreement in rough terms ("positive", "exciting") is higher for valence than arousal, yet perception of the dynamics ("more", "less") is more uniform for arousal. Factors expressing the viewer's opinion alter agreement as expected; for example, users that like a particular movie agree more with each other and with the intended emotion. Expected and intended emotion end up being highly similar, with correlation coefficients of 0.74 for arousal and 0.70 for valence. Before using for classification, the expected and intended emotion

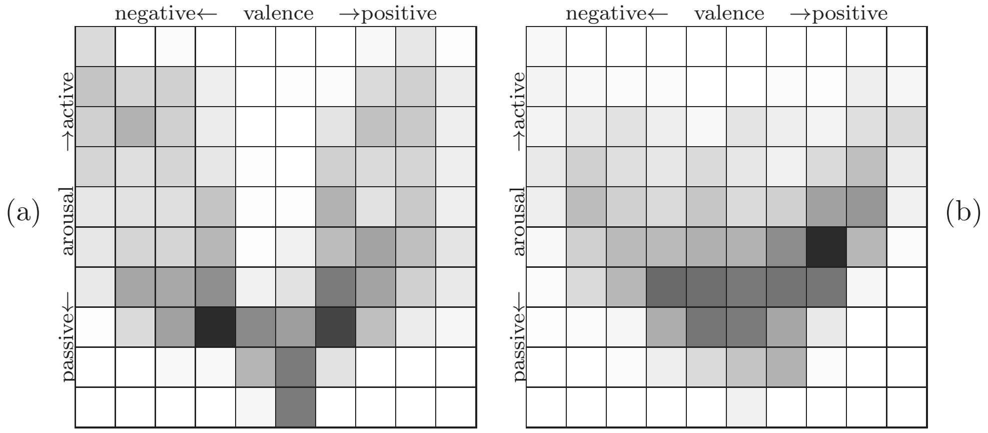


Figure 2.6: Joint valence-arousal histograms for (a) intended and (b) expected emotion (darker signifies higher value).

Table 2.1: Inter-annotator agreement.

metric	valence	arousal
correlation	0.293	0.409
difference of means	0.288	0.411
mean abs. difference	0.445	0.513
Krippendoff's α ordinal (7 levels)	0.308	0.152
Cohen's k (7 levels)	0.035	0.029

curves are quantized into seven equi-probable bins, using the cumulative distribution function estimated via Parzen windows. The category boundaries are shown in Fig 2.5. While the 7 levels per dimension are equi-probable, the $7 \times 7 = 49$ areas are not.

2.3 System Design

Emotion is a dynamic process that evolves rapidly through time. In order to capture the dynamic nature of emotion, we choose to use hidden Markov models that are popular in time series modeling and have been shown to work to model emotion [27]. The next important modeling issue is how to handle the two affective dimensions. As shown in Fig 2.2, arousal and valence are correlated. A way to exploit this relation would be either to model arousal and valence jointly, e.g., using 2-D HMMs, or to use a series of classifiers, e.g., the output of the arousal classifier being (one of) the input(s) of the

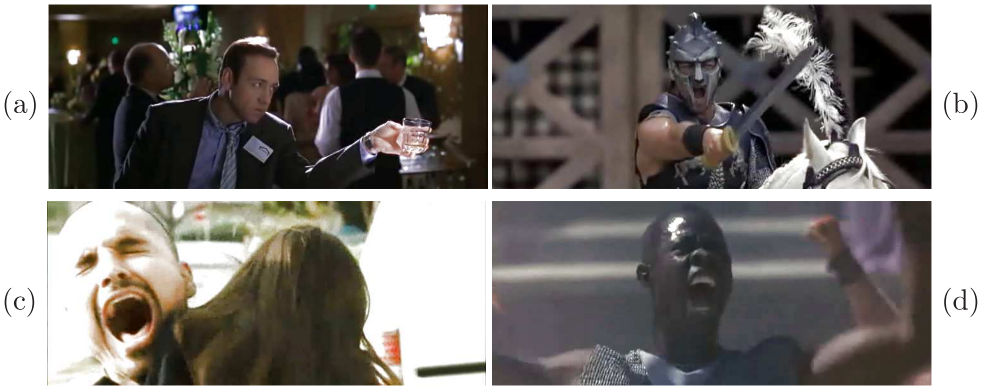


Figure 2.7: Sample frames for: (a) Low arousal, (b) High arousal, (c) Very negative valence, (d) Very positive valence.

valence classifier. In this paper, we choose to use independent classifiers, one for each dimension, which are also evaluated separately.

HMMs using various numbers of states and Gaussian components were evaluated. We found that increasing the number of states is more beneficial than increasing the number of Gaussian components, particularly when using short-time spectral envelope audio features, e.g. Mel Frequency Cepstral Coefficients (MFCCs), presumably because longer models better capture complex temporal interactions between low level features and emotion. Results are presented next for recognizers that model each affective category with a left-to-right HMM with 32 hidden states and a single Gaussian distribution per state. Inter-category transitions are modeled with a bigram language model that only allows transitions between adjacent categories. Humans don't change affective levels very fast and the language model probabilities are assigned a large exponential weight (40) compared to the acoustic-visual features (1). This weighting results also in smoother curves. The models are trained using the *Baum-Welch* algorithm and classification is achieved via the *Viterbi* algorithm (using the HTK speech recognition package).

2.3.1 Audio features

A variety of features have been investigated broadly separated into three categories (modalities): audio, music and visual features. The low level audio features tested were: fundamental frequency (F0), intensity, log energy, signal zero crossings rate,

Table 2.2: The effect of number of states and number of Gaussian components on the performance of arousal and valence prediction. The metrics are: accuracy, accuracy±1 and correlation of the discrete curve and Mean square error and correlation of the continuous curve.

arousal						
states	Gaussians	ACC	ACC±1	D.CORR	MSQE	C.CORR
3	3	0.22	0.53	0.34	0.20	0.46
1	1	0.19	0.42	0.18	0.41	0.28
1	2	0.23	0.50	0.33	0.27	0.45
1	4	0.22	0.50	0.32	0.25	0.44
1	8	0.22	0.51	0.32	0.24	0.44
1	16	0.22	0.52	0.33	0.23	0.45
1	1	0.19	0.42	0.18	0.41	0.28
2	1	0.19	0.49	0.23	0.24	0.36
4	1	0.23	0.52	0.32	0.23	0.43
8	1	0.22	0.54	0.38	0.21	0.48
16	1	0.23	0.55	0.40	0.21	0.49
32	1	**0.24**	**0.57**	0.43	0.20	0.51
64	1	0.23	**0.57**	0.43	0.21	0.50
8	2	0.23	0.56	0.40	**0.19**	0.51
8	3	0.21	0.55	0.39	**0.19**	0.51
16	2	0.22	0.55	0.43	0.20	**0.53**
16	3	0.22	0.56	**0.44**	**0.19**	**0.53**

valence						
states	Gaussians	ACC	ACC±1	D.CORR	MSQE	C.CORR
3	3	0.20	0.51	0.10	0.30	0.16
1	1	0.16	0.43	-0.01	0.40	0.03
1	2	0.18	0.45	0.03	0.40	0.04
1	4	0.18	0.48	0.06	0.32	0.11
1	8	0.20	0.50	0.09	0.29	0.16
1	16	0.20	0.50	0.07	0.31	0.13
1	1	0.16	0.43	-0.01	0.40	0.03
2	1	0.17	0.46	0.06	0.35	0.10
4	1	0.17	0.49	0.08	0.33	0.11
8	1	0.21	0.55	0.13	0.29	0.17
16	1	0.21	0.56	**0.18**	**0.28**	**0.22**
32	1	**0.22**	0.57	0.16	0.30	0.19
64	1	**0.22**	**0.58**	0.17	0.33	0.19
8	2	0.20	0.53	0.10	0.31	0.12
8	3	0.21	0.52	0.08	0.32	0.11
16	2	0.21	0.53	0.12	0.31	0.16
16	3	0.21	0.55	0.15	0.30	0.18

Table 2.3: The effect of language model class (0: zerogram, 1: unigram, 2:bigram) and complexity on the performance of arousal and valence prediction. The metrics are: accuracy, accuracy±1 and correlation of the discrete curve and Mean square error and correlation of the continuous curve.

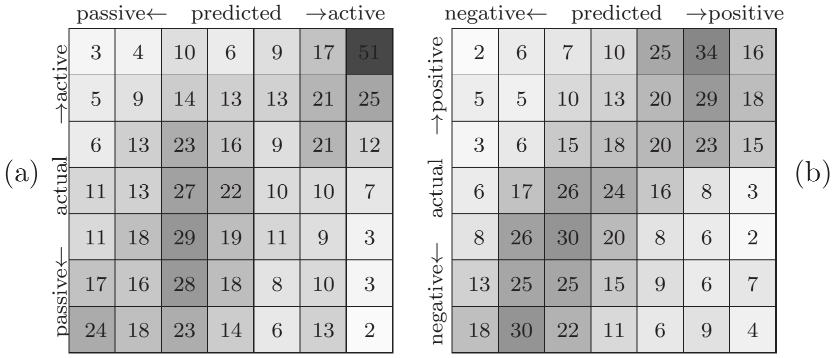
arousal						
lm class	weight	ACC	ACC±1	D.CORR	MSQE	C.CORR
2	1	0.22	0.53	0.34	**0.20**	0.46
0	1	0.21	0.52	0.31	**0.20**	0.45
1	1	0.22	0.52	0.32	**0.20**	0.45
1	20	0.22	0.54	0.38	**0.20**	0.48
1	50	0.22	0.54	0.39	0.21	0.46
1	100	0.22	0.53	0.39	0.22	0.44
2	20	0.23	0.56	0.41	**0.20**	0.48
2	30	0.23	0.57	0.43	**0.20**	0.49
2	40	**0.24**	0.57	0.44	**0.20**	0.50
2	50	**0.24**	**0.58**	**0.45**	**0.20**	**0.51**
2	100	**0.24**	**0.58**	0.43	0.21	0.49
2	200	**0.24**	0.56	0.40	0.22	0.45
2	500	**0.24**	0.56	0.38	0.24	0.43

valence						
lm class	weight	ACC	ACC±1	D.CORR	MSQE	C.CORR
2	1	0.20	0.51	0.10	**0.30**	0.16
0	1	0.19	0.49	0.09	**0.30**	0.16
1	1	0.19	0.50	0.09	**0.30**	0.15
1	20	0.20	0.51	0.11	0.33	0.14
1	50	0.21	0.52	0.10	0.35	0.12
1	100	0.21	0.52	0.09	0.38	0.10
2	20	0.21	0.54	0.14	0.32	0.16
2	30	0.21	0.54	0.14	0.32	0.16
2	40	0.21	0.55	0.16	0.32	0.18
2	50	0.22	0.55	0.16	0.32	0.18
2	100	**0.23**	**0.56**	**0.17**	0.32	**0.19**
2	200	0.22	**0.56**	0.15	0.31	0.17
2	500	0.22	0.54	0.13	0.33	0.15

Table 2.4: List of features used for emotion recognition.

Valence	audio	12 MFCCs and C0, plus derivatives
	video	maximum color value
	video	maximum color intensity
Arousal	audio	12 MFCCs and C0, plus derivatives

spectral centroid, spectral flux, spectral roll-off, line spectral pairs, chroma coefficients, MFCCs and Perceptual Linear Prediction (PLP) coefficients. Audio features were extracted via OpenSMILE [18] using a 200ms window and 40ms update. We also created a more extensive feature set by extracting the aforementioned low level features using a 40ms window, 10ms update, then calculating the statistics of these samples (moments, derivatives, extrema) within a 200ms window (and using the statistics as features). High level music features were extracted using the MIR Toolbox [29], namely: tempo, pulse clarity, event density, spectral flatness, rhythm irregularity and inharmonicity. These features must be computed using a larger window in order to be meaningful, so we used a window of 1sec, updated every 40ms. The video features used were the statistics of color, intensity and motion, extracted, per video frame (40ms), through the algorithms described in [42]. All features were evaluated using three models of increasing complexity (states, Gaussian components). The selected feature set was created by hierarchically merging the best performing features. The rejected features did not necessarily perform inadequately, some were simply highly correlated with "more successful" features and therefore provided no additional benefit. Energy and all energy-related features (e.g., 0th order MFCC) performed very well, as expected, for detecting arousal and for separating neutral from non-neutral valence (but were not able to distinguish between positive and negative valence). F0 and rhythm-based features performed poorly; this was perhaps due to the complexity of the audio stimulus containing speech, music, silence and various audio sounds. Visual motion and (musical) tempo performed well individually but failed to provide any additional improvement if the feature set contained energy-based features. MFCCs, PLPs and Chroma coefficients performed similarly in isolation. Color-based video features proved valuable in valence classification. All in all, the selected parsimonious features set that provided the best emotion recognition results can be seen in Table 2.4.



(a)

	passive←			predicted			→active
→active	3	4	10	6	9	17	51
	5	9	14	13	13	21	25
	6	13	23	16	9	21	12
actual	11	13	27	22	10	10	7
	11	18	29	19	11	9	3
	17	16	28	18	8	10	3
passive←	24	18	23	14	6	13	2

(b)

	negative←			predicted			→positive
→positive	2	6	7	10	25	34	16
	5	5	10	13	20	29	18
	3	6	15	18	20	23	15
actual	6	17	26	24	16	8	3
	8	26	30	20	8	6	2
	13	25	25	15	9	6	7
negative←	18	30	22	11	6	9	4

Figure 2.8: Misclassification matrices normalized by row (%) for (a) arousal and (b) valence.

2.3.2 Text features

To add textual information to the model we extract a number of audio features and simply append them to the audio-visual feature vector. The features are word frequencies within a window, similarly to [41]. First we assign a rating to each subtitle utterance, equal to the average value of the affective curve for the duration of the utterance. Words are assigned ratings equal to the sentence ratings in which each word appears. These ratings are used to calculate the Mutual Information for each word-category pair and the 50 words with the highest ratings are selected as features. Finally, in order to create the feature vectors we use a very large temporal window (60 - 200 seconds) and count the number of appearances of each word-feature within that window.

2.4 Experimental Results

The output of our system is a –usually very noisy– time series of seven categories. The signal is initially filtered with a low pass filter and then passed through a Savitzky-Golay filter [44] that interpolates the affective signal into a continuous-valued curve. An example of discrete and continuous output is shown in Fig 2.10. To evaluate our system we compare the (seven-level) discrete output of the HMM system with the discretized affective curves. The interpolated continuous output curves are also compared with the reference continuous affective curves. Thus separate results are provided for the discrete-valued and continuous-valued curves.

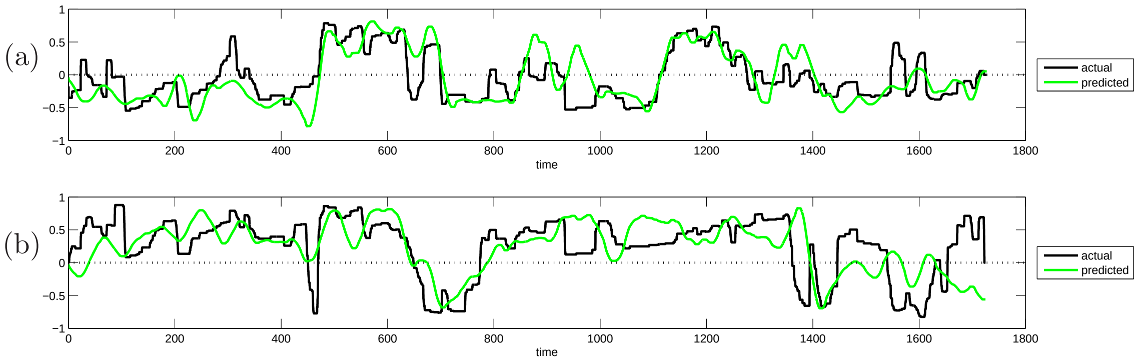


Figure 2.9: Continuous intended emotion recognition vs. annotated curves for the “Ratatouille” clip: (a) arousal and (b) valence.

Table 2.5: Result evaluation metrics.

	metric	arousal	valence	2-D
Discrete (7 levels)	Accuracy	0.24	0.24	0.06
	Accuracy±1	0.57	0.62	0.37
	Mean abs. error	0.52	0.47	0.82
	Mean sq. error	0.48	0.43	0.92
	Correlation	0.43	0.22	-
Continuous	Mean abs. error	0.32	0.37	0.55
	Mean sq. error	0.17	0.24	0.41
	Correlation	0.54	0.23	-

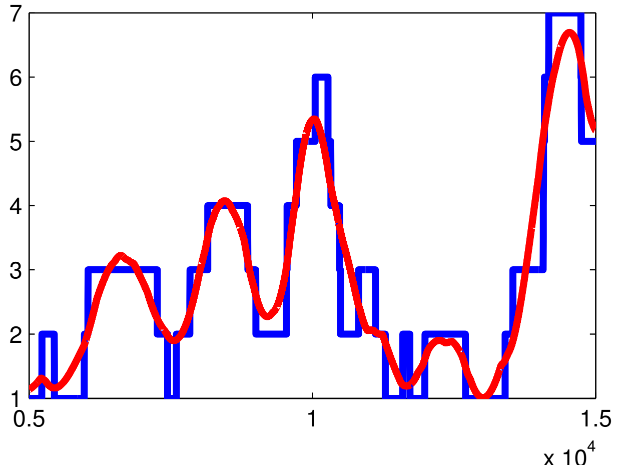


Figure 2.10: An example of a discrete curve and it's continuous form obtained via our method.

Experiments are conducted using a "leave one (movie) out" cross-validation scheme. Results are presented as averages across all clips. The following evaluation metrics are shown: classification accuracy, classification accuracy ±1 (which considers a miss by 1 category as a hit), mean absolute error (MAE), mean square error (MSE) and correlation coefficient. MAE and MSE are calculated after rescaling the curves to a $[-1, 1]$ range.

Table 2.2 shows the effect that altering the number of HMM states and Gaussian components has on performance, when using MFCCs as features. As noted previously, the number of states seems far more important than the complexity of observation distributions. Table 2.3 shows performance with different classes (zerogram, unigram and bigram) and weights of language models used to represent inter-category transition probabilities. There is a definite benefit to using a bigram model, while the weight has a sweet spot around 50. The final model uses a bigram model with a weight of 40 and HMMs with 32 states and single Gaussian components.

Results from the final model are shown in Table 2.5. Classification accuracy for seven classes is, as expected, rather low at 25%. Accuracy±1 (equivalent to using fewer categories) is fairly high at 60%; given the variety of movies in our database and the difficulty of the task this is a promising result. Note the very low correlation for valence that is further investigated next. Smoothing the discrete-valued curves further

Table 2.6: Performance with the addition of Text features. The metrics are: accuracy, accuracy±1 and of the discrete curve and correlation of the continuous curve.

	Arousal	Valence
Accuracy	0.1471	0.1708
Accuracy±1	0.4225	0.4839
Correlation	0.0311	0.1185

improves our results, as can be seen from the significantly improved MSE and MAE continuous results, especially for arousal.

Fig 2.8 shows the misclassification matrices for arousal (a) and valence (b) normalized by the sum of each row, i.e, each cell (i, j) indicates the percentage of samples that belong to category i (actual) and are classified in category j (predicted). Best emotion recognition results are obtained for high arousal values, over 50% of the high activity frames are classified correctly (level 7). Note that frames are rarely misclassified to very distant categories, while neighboring categories are highly confusable.

Adding subtitle information did not work. No performance benefit was observed throughout our experiments. Adding more features, such as morphology (like punctuation) did not improve results. Table 2.5 shows the performance achieved when adding subtitle features to our audiovisual model. Correlation in particular suffers, showing how bad this approach really is.

Overall, the classifiers on both dimensions perform very well in classifying the mood of large segments, with the arousal classifier also performing well in describing detailed dynamics. The valence classifier fails at describing the continuous curve in detail, as revealed by the low correlation coefficient. Interestingly, this observation, as well as the overall relative performance of the classifiers in the two dimensions (prior to interpolation) also holds true for the performance of human annotators when evaluating their own experience (see Section 2.2). Note that a typical error in valence recognition is the misclassification of a contiguous area to entirely wrong valence categories, very positive scenes being identified as very negative and vice versa. This seems to happen in scenes where there is a conflict of modalities (e.g., “joyous” music, but “angry” video) or a conflict of sensory and semantic information. Our system lacks such semantic information, so it can not understand that a dark and gloomy battle will be perceived as positive if the viewers know that the hero is going to win. An example actual vs. predicted annotation for a 30 minute movie clip is shown in Fig 2.9.

2.5 Conclusions

We have briefly presented an annotated database of affect and our experiments in tracking the affective contents of the movies using HMMs. Evaluation of a large number of audio-visual features yielded somewhat surprising results, with many popular features being rejected before selecting the "optimal" feature set. Two independent HMM recognizers were used for arousal and valence, each utilizing a small number of low level features and a large number of states. The recognizers work well at a macroscopic level, capturing the general mood of the vast majority of scenes across movies. On the arousal dimension, the model also does well in capturing fine detail, subtle transitions, as revealed by the average correlation coefficient of 0.54. On the valence dimension, the model is successful at capturing the mood but sometimes fails at capturing the valence sign and transitions. Overall this is a first step towards continuous emotion recognition in movies. Further research in feature extraction, high-level semantic analysis, modeling and modality fusion is required to improve these results.

Chapter 3

Text Affect*

3.1 Introduction

Affective text analysis, the analysis of the emotional content of lexical information is an open research problem that is very relevant for numerous natural language processing (NLP), web and multimodal dialogue applications. One very popular such application is *sentiment analysis/opinion mining*, which aims to identify the emotion expressed in news stories [32], blogs and public forums [5] or product reviews [24, 58]. Generally opinion mining is restricted to separating positive and negative views or positive, negative and neutral views and focused on writer-perspective emotion, the emotion expressed by the writer, rather than the emotion experienced by the reader. Emotion recognition from multimedia streams (audio, video, text) and emotion recognition of users of interactive applications is another area where the affective analysis of text plays an important, yet still limited role [3, 30, 31]. Another application is the creation of affective text/speech, to be used in *Human Computer Interaction (HCI)* [6]: realistic expression of emotion by any artificial representation of a human is integral to their believability and effectiveness. Such systems should also be capable of responding appropriately to expressions of emotion by the humans using them. Again, this application requires writer-perspective emotion. Other applications may focus on the reader/media consumer perspective, such as multimedia content analysis through subtitles [41] or news headlines analysis [47]. The requirements of different applications lead to the definition of sub-tasks, such as emotional category labeling (assigning text

*Parts of this chapter have appeared in [34] and [35] a label, such as “sad”), polarity recognition (classifying into positive or negative) and subjectivity identification (separating subjective from objective statements). Furthermore the task is affected heavily by the emotional representation used (such as basic emotions or valence) and the scope of analysis (word, sentence, documents characterization).

Given the wide range of important applications, affective text analysis has been a popular topic of research in the NLP community in recent years. However the wide range of application scenarios and the different ways to define affective tasks have also lead to a fragmentation of research effort. The first step towards a general task-independent solution to affective text analysis is the creation of an appropriate affective lexicon; a resource mapping each word (or term) to a set of affective ratings. A number of affective lexicons for English have been manually created, such as the General Inquirer [46], and Affective norms for English Words (ANEW) [8]. However, they fail to provide the required vocabulary coverage; the negative and positive classes of the General Inquirer contain just 3600 words, while ANEW provides ratings for just 1034 words. Therefore computational methods are necessary to create or expand an already existing lexicon. Well-known lexica resulting from such methods are SentiWordNet [16] and WORDNET AFFECT [48]. However, such efforts still suffer from limited coverage.

Our aim is to create an affective lexicon containing fine-grained/pseudo-continuous valence ratings, ranging from very negative to very positive. This lexicon can be readily expanded to cover unseen words with no need to consult ontologies or other linguistic resources. The work builds on [53]. The proposed method only requires a small number (a few hundred) labeled seed words and a web search engine to estimate similarity between the seed and unseen words. Further, to improve the quality of the affective lexicon we propose a machine learning approach to training a linear valence estimator. The affective lexicon created is evaluated against manually labeled corpora both at the word and the sentence level, achieving state-of-the-art results despite the lack of underlying syntactic or pragmatic information in our model.

3.2 Prior Work

The task of assigning affective ratings, such as binary “positive - negative” labels, also known as semantic orientation [23], has received much attention and a wide variety of methods have been proposed. The underlying assumption at the core of these methods is that *semantic similarity can be translated to affective similarity.* Therefore given some metric of the similarity between two words one may derive the similarity between their affective ratings. This approach, pioneered in [53], is using a set of words with known affective ratings, then using the similarities between these words and every new word to define this new word's ratings. These reference words are usually referred to as *seed words.* There is significant variety on the nature of the seed words; they may be the lexical labels of affective categories ("anger","happiness"), small sets of words with un-ambiguous meaning or even all words in a large lexicon. Having a set of seed words and an appropriate similarity measure, the next step is devising a method of combining these to create the final rating. In most cases the desired rating is some form of binary label like "fear" - "not fear", in which case a classification scheme, like *nearest neighbour* may be used to provide the final result. Alternatively, continuous/pseudo-continuous ratings may be acquired via some numerical combination of similarities and known ratings [49].

In [53] and [54] the method used (which is very similar to our own) utilizes conjunctive "NEAR" queries to get the co-occurrence of words in web documents, from which semantic similarity is extracted through point-wise mutual information. The estimated valence $\hat{v}(w_j)$ of each new word w_j is expressed as a linear combination of the valence ratings $v(w_i)$ of the seeds w_i and the semantic similarities between the new word and each seed $d(w_i, w_j)$ as;

$$\hat{v}(w_j) = \sum_{i=1}^{N} v(w_i) \cdot d(w_i, w_j), \tag{3.1}$$

The seeds used are 14 adjectives (7 pairs of antonyms) shown in Table 3.1 and their known ratings are binary (-1 or 1). The method is shown to work very well in terms of binary (positive/negative) classification, achieving an 82.8% accuracy in the general inquirer dataset. The method is capable of creating continuous ratings - though it's performance in that task is not explored. The major weakness of this method is it's dependency on the, now defunct, altavista NEAR queries. While AND queries, which are available through all search engines, return all documents that contain both terms, altavista NEAR queries returned all documents where both terms existed *with a distance of 10 words.* As shown in [54] and [50] the method performs much worse using AND queries.

Table 3.1: The 14 seeds used in the experiments by Turney and Littman.

positive	negative
good	bad
superior	inferior
positive	negative
correct	wrong
fortunate	unfortunate
nice	nasty
excellent	poor

In [52], one of the most imaginative approaches, the "spin model", is proposed as a method of binary word classification. A network of words, representing word relatedness, is constructed using gloss definitions, thesaurus, and co-occurrence statistics. Each word is regarded as en electron and has a "spin" with either an "up" or "down" orientation. Neighboring spins tend to have the same orientation from an energetic point of view, which is similar to neighboring words having similar valence. The problem is handled as an optimization problem and solved through the use of the mean field method. This method is very complex, requires a lot of different resources and it can not be used in more complex problems, it is strictly a binary classification method. This of course is problematic due to the fact that most words have a low, insignificant in the binary case, valence.

WordNet based methods such as those in [15], [1] and [48] start with a small set of annotated words, usually with binary ratings. These sets are then expanded by exploiting synonymy, hypernymy and hyponymy relations along with simple rules. After that there are different approaches to calculating the similarity between new words and the seed words, but a common approach is using contextual similarity based on glosses.

In [22], a random walk model is used to perform binary classification of words into positive or negative. First a word network is created, based on Wordnet relations (synonymy, hypernymy), on which related words are connected. In order to classify a word of unknown valence, multiple random walks are initiated from the unknown word. Each random walk stops when it hits a word of known valence, a seed. Then the unknown word is classified based on the average number of steps required to hit a seed of the selected category. The method is computationally expensive, yet fails to overcome the results produced by the method in [54].

The next step is the combination of these word ratings to create ratings for larger lexical units, phrases or sentences. Initially the affect-bearing words need to be selected, depending on their part-of-speech tags [11], affective rating and/or the sentence's structure [2]. Then their individual ratings are combined, typically in a simple fashion, such as a numeric average. More complex approaches involve taking into account sentence structure, word/phrase level interactions such as valence shifters [40] and large sets of manually created rules [2, 11]. In [37] a supervised method is used to train the parameters of multiple hand-selected rules of composition.

As discussed, most of the aforementioned work is based on the assumption that semantic similarity implies affective similarity. Thus it is important to also review the literature on computational methods for estimating semantic similarity between words or terms. Semantic similarity metrics can be roughly categorized into: (i) ontology-based similarity measures, e.g., [9], where similarity features are extracted from ontologies (usually WordNet), (ii) context-based similarity measures [38], where similarity of context is used to estimate semantic similarity between words or terms, (iii) hit-based similarity metrics where the frequency of co-occurrence of terms in (web) documents is the main feature used for estimating semantic similarity [50, 53], and (iv) combinations of the aforementioned methods [7]. Recently corpus-based methods (especially context-based metrics) where shown to perform almost at a par with ontology-based metrics. For details see [25].

3.3 A supervised approach to Affective Lexicon creation

Just as in [53] we start from an existing, manually annotated lexicon. Then we automatically select a subset of it to be used as seed words. The rating (in our case valence) for an unseen word is estimated as the linear combination of the ratings of seed words weighted by the semantic similarity between the unseen and seed words. In addition, a linear weight is used that regulates the contribution of each seed word in the valence computation. The weight of each seed word is selected to minimize the mean square estimation error on all words in the training set.

The motivation behind introducing a trainable weight for each seed word has to do with the fact that semantic similarity does not fully capture the relevance of a seed word for valence computation. For instance, consider an unseen word and a lexicon that consists of two seed words that are equally (semantically) similar to the unseen word. Based on the assumption that semantic similarity implies affective similarity both seed words should be assigned the same feature weight. However, there is a wide range of factors affecting the relevance of each seed word, e.g., words that have high affective variance (many affective senses) might prove to be worse features that affectively unambiguous words. Other factors might include the mean valence of seed words and the degree of centrality (whether they are indicative samples of their affective area). Instead of evaluating the effect of each factor separately, we choose to use machine learning to estimate a single weight per seed word using *Least Mean Squares estimation (LMS)*.

3.3.1 Word Level Tagging

We aim at characterizing the affective content of words in a continuous valence range of $[-1, 1]$ (from very negative to very positive), *from the reader perspective*. We hypothesize that the valence of a word can be estimated as a linear combination of its semantic similarities to a set of seed words and the valence ratings of these words, as follows:

$$\hat{v}(w_j) = a_0 + \sum_{n=1}^{N} a_i \; v(w_i) \; f(d(w_i, w_j)), \tag{3.2}$$

where w_j is the word we mean to characterize, $w_1...w_N$ are the seed words, $v(w_i)$ is the valence rating for seed word w_i, a_i is the weight corresponding to word w_i (that is estimated as described next), $d(w_i, w_j)$ is a measure of semantic similarity between words w_i and w_j (see Section 3.3.1.1) and $f()$ is some simple function, from the table 3.2. The function $f()$, which we will henceforth call the kernel of the equation, serves to rescale the similarity metric $d(w_i, w_j)$.

$$\begin{bmatrix} 1 & f(d(w_1,w_1))v(w_1) & \cdots & f(d(w_1,w_N))v(w_N) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & f(d(w_K,w_1))v(w_1) & \cdots & f(d(w_K,w_N))v(w_N) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 \\ v(w_1) \\ \vdots \\ v(w_K) \end{bmatrix} \tag{3.3}$$

Table 3.2: The functions of similarity used.

linear	$f(d(\bullet)) = d(\bullet)$
exp	$f(d(\bullet)) = e^{d(\bullet)}$
log	$f(d(\bullet)) = log(d(\bullet))$
sqrt	$f(d(\bullet)) = \sqrt{d(\bullet)}$

Assuming we have a training corpus of K words with known ratings and a set of $N < K$ seed words for which we need to estimate weights a_i, we can use (3.2) to create a system of K linear equations with $N + 1$ unknown variables as shown in (3.3); the N weights $a_1 ... a_N$ and the extra weight a_0 which acts as a DC offset (bias). The optimal values of these variables can be estimated using LMS. Once the weights of the seed words are estimated the valence of an unseen word w_j can be computed using (3.2).

3.3.1.1 Semantic Similarity Metrics

The valence estimator defined in (3.2) employs a metric $d(w_i, w_j)$ that computes the semantic similarity between words w_i and w_j. In this work, we use hit-based and text-based (contextual) similarity metrics.

Hit-based similarity metrics estimate the similarity between two words/terms using the frequency of co-existence within larger lexical units (sentences, documents). The underlying assumption is that terms that co-exist often are very likely to be related. A popular method to estimate co-occurrence is to pose conjunctive queries including both terms to a web search engine; the number of returned hits is an estimate of the frequency of co-occurrence [25]. Hit-based metrics do not depend on any language resources, e.g., ontologies, and do not require downloading documents or snippets, as is the case for context-based semantic similarities.

In the equations that follow, $w_i, \ldots, w_{i+n}$ are the query words, $\{D; w_i, \ldots, w_{i+n}\}$ is the set of results $\{D\}$ returned for these query words. The number of documents in each result set is noted as $\mid D; w_i, \ldots, w_{i+n} \mid$. We investigate the performance of four different hit-based metrics, defined next.

Jaccard coefficient computes similarity as:

$$J(w_i, w_j) = \frac{|\,D; w_i, w_j\,|}{|\,D; w_i\,| + |\,D; w_j\,| - |\,D; w_i, w_j\,|}. \tag{3.4}$$

It is bounded in the range $[0, 1]$.

Dice coefficient is a variation of the Jaccard coefficient, defined as:

$$C(w_i, w_j) = \frac{2\,|\,D; w_i, w_j\,|}{|\,D; w_i\,| + |\,D; w_j\,|}. \tag{3.5}$$

Mutual information [7] is an info-theoretic measure that derives the similarity between w_i and w_j via the dependence between their number of occurrences. Point-wise Mutual Information (PMI) is defined as:

$$I(w_i, w_j) = \log \frac{\frac{|D; w_i, w_j|}{|D|}}{\frac{|D; w_i|}{|D|}\frac{|D; w_j|}{|D|}}. \tag{3.6}$$

Mutual information is unbounded and can take any value in $[-\infty, +\infty]$. Positive values translate into similarity, negative values into dissimilarity (presence of one word tends to *exclude* the other) and zero into independence, lack of relation.

Google-based Semantic Relatedness Normalized Google Distance is a distance metric proposed in [12, 56] and defined as:

$$E(w_i, w_j) = \frac{\max\{L\} - \log\,|\,D; w_i, w_j\,|}{\log\,|\,D\,| - \min\{L\}}, \tag{3.7}$$

where $L = \{\log|\,D; w_i\,|,\ \log|\,D; w_j\,|\}$. This metric is unbounded, taking values in $[0, +\infty]$. [19] used Normalized Google Distance to define a bounded (in [0, 1]) metric, called Google-based Semantic Relatedness, defined as:

$$G(w_i, w_j) = e^{-2E(w_i, w_j)}. \tag{3.8}$$

Text-based similarity metrics compute cosine similarity between feature vectors extracted from word or term context, i.e., using a “bag-of-words” context model. The basic assumption behind these metrics is that similarity of context implies similarity of meaning, i.e., words that appear in similar lexical environment (left and right contexts) have a close semantic relation [43],[38]. “Bag-of-words” [26] models assume that the feature vector consists of words or terms that occur in text independently of each other. The context-based metrics presented here employ a context window of fixed size (K

words) for feature extraction. Specifically, the right and left contexts of length K are considered for each occurrence of a word or term of interest w in the corpus, i.e., $[v_{K,L} \cdots v_{2,L}\ v_{1,L}]w[v_{1,R}\ v_{2,R} \cdots v_{K,R}]$ where $v_{i,L}$ and $v_{i,R}$ represent the ith word to the left and to the right of w, respectively. The feature vector for word or term w is defined as $T_{w,K} = (t_{w,1}, t_{w,2} \cdots t_{w,N})$ where $t_{w,i}$ is a non-negative integer and K is the context window size. Note that the length of the feature vector is equal to the vocabulary size N, i.e., all words in the vocabulary are features. The ith feature value $t_{w,i}$ reflects the (frequency of) occurrence of vocabulary word v_i within the left or right context window K of (all occurrences of) the term w. The value of $t_{w,i}$ may be defined as a (normalized or unnormalized) function of the frequency of occurrence of feature i in the context of w. Once the feature weighting scheme is selected, the "bag-of-words"-based metric S^K computes the similarity between two words or terms, w_1 and w_2, as the cosine similarity of their corresponding feature vectors, $T_{w_1,K}$ and $T_{w_2,K}$ as follows, [26]:

$$S^K(w_1, w_2) = \frac{\sum_{i=1}^{N} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{N}(t_{w_1,i})^2}\sqrt{\sum_{i=1}^{N}(t_{w_2,i})^2}} \tag{3.9}$$

where K is the context window length and N is the vocabulary size. The cosine similarity metric assigns 0 similarity score when w_1, w_2 have no common context (completely dissimilar words), and 1 for identical words. Various feature weighting schemes can be used to compute the value of $t_{w,i}$. The binary weighting metric used in this work assigns weight $t_{w,i} = 1$ when the ith word in the vocabulary exists at the left or right context of at least one instance of the word w, and 0 otherwise. Alternative weighting schemes such as tf-idf are more popular, but we opt for binary weights to reduce computational complexity.

3.3.2 Sentence Level Tagging

The principle of compositionality [39] states that the meaning of a phrase or sentence is the sum of the meaning of its parts. One could readily extend this rule to affective interpretation. In fact, since affect can be measured in a metric space, the generalization of the principle of compositionality to affect could be interpreted as follows: to compute the valence of a sentence simply take the average valence of the words in that sentence.

The affective content of a sentence $s = w_1 w_2 ... w_N$ in the simple linear model is:

$$v(s) = \frac{1}{N} \sum_{i=1}^{N} v(w_i). \tag{3.10}$$

This simple linear fusion may prove to be inadequate for affective interpretation given that non-linear affective interaction between words (especially adjacent words) in the same sentence is common. Linear fusion may prove suboptimal since it weights equally words that have a strong and weak affective content. It also tends to give lower absolute valence scores to sentences that contain many neutral (non-content) words. Thus we also consider a normalized weighted average, in which words that have high absolute valence values are weighted more, as follows:

$$v(s) = \frac{1}{\sum\limits_{i=1}^{N} |v(w_i)|} \sum_{i=1}^{N} v(w_i)^2 \cdot \mathrm{sign}(v(w_i)), \tag{3.11}$$

where sign(.) is the signum function. One could also generalize to higher powers or to other non-linear scaling functions. Alternatively we consider non-linear min-max fusion, in which the word with the highest absolute valence value dominates the meaning of the sentence:

$$\begin{aligned} v(s) &= \max_i(|v(w_i)|) \cdot \mathrm{sign}(v(w_z)) \\ z &= \arg\max_i(|v(w_i)|) \end{aligned} \tag{3.12}$$

where arg max is the argument of the maximum. One could also consider combinations of linear and non-linear fusion methods, as well as, syntactic- and pragmatic-dependent fusion rules. However, more complex fusion methods are beyond the scope of this work that focuses on the evaluation of the affective lexicon creation algorithm.

3.4 Corpora and Experimental Procedure

Next we present the corpora used for training and evaluation of the proposed algorithms. In addition, the experimental procedure for semantic similarity computation, affective lexicon creation and sentence-level affective score computation is outlined.

3.4.1 Corpora

The main corpus used for creating the affective lexicon is the *Affective Norms for English Words* (ANEW) dataset. ANEW consists of 1034 words, rated in 3 continuous dimensions of arousal, valence and dominance. In this work, we only use the valence ratings provided in ANEW.

The second corpus used for evaluation of the affective lexicon creation algorithm is the General Inquirer (GINQ) corpus that contains 2005 negative and 1636 positive words. The General Inquirer corpus was created by merging words with multiple entries in the original lists of 2293 negative and 1914 positive words. It is comparable to the dataset used in [53, 54].

Both the ANEW and GINQ datasets were used for both training (seed words) and evaluation using cross-validation (as outlined below).

For the sentence level tagging evaluation (no training is performed here, only testing) the *SemEval 2007: Task 14* corpus is used [47]. This SemEval corpus contains 1000 news headlines manually rated in a fine-grained valence scale of $[-100, 100]$, which is rescaled to a $[-1, 1]$ for our experiments.

For the movie subtitle evaluation task, we use the subtitles of our movie corpus. It contains the subtitles of twelve thirty minute movie clips, a total of 5388 sentences. Start and end times of each utterance were extracted from the subtitles and each sentence was given a continuous valence rating equal to the average of the multimodal affective curve for the duration of the utterance.

3.4.2 Semantic Similarity Computation

In our experiments we utilized four different similarity metrics based on web co-occurrence, mentioned in section 3.3.1.1, namely, *Dice coefficient, Jaccard coefficient, point-wise mutual information (PMI) and Google-based Semantic Relatedness* as well as a single contextual similarity metric, cosine similarity with binary weights, calculated over snippets and documents.

All of the similarity metrics employed require a corpus in order to count number of appearances or collect context. The corpus we use in this work is the web and the data required to calculate the similarity metrics are collected by submitting queries to the Yahoo! search engine and collecting the response.

Hit-based similarity metrics require the individual (IND) words number of occurrences as well as the number of times that the two words co-exist. This co-occurrence may be within the boundaries of a single web document (page), in which case the corresponding hit count can be obtained by using an AND query (w_1 AND w_2). Alternatively we can restrict it by demanding that the two words co-occur within a set distance of 10 words, in which case the corresponding hit count can be obtained by using a NEAR query (w_1 NEAR w_2). NEAR queries are an undocumented feature of the Yahoo! engine and are similar though different to the ones produced by Altavista. Depending on they type of query used to obtain the co-occurence count, we will separate the similarity metrics into counts/AND and counts/NEAR. The number of seed words determines the number of queries that will be required. Assuming N seed words are selected, $N + 1$ queries will be required to rate each new word.

Context-based similarities require a sizable collection of documents that include either or both of the words under examination. To collect the required documents we pose IND queries to the Yahoo! search engine and collect the top $|D|$ (if available) results for each word. For each of these $|D|$ results we add some text to the corpus. The text in question may be the short excerpt (page sample) shown under each result called a snippet, typically one or two sentences automatically selected by the search engine, or the full web document that each result points to. Therefore we create two corpora, one containing all the collected snippets and one containing the corresponding documents. Depending on which corpus was used to calculate the similarity metrics, we separate them into documents/IND and snippets/IND. Once the documents are downloaded, the left and right contexts of all occurrences of w_1 and w_2 are examined and the corresponding feature vectors are constructed. The parameters of this calculation are the number $|D|$ of web documents used and the size K of the context window. In all experiments presented in this work $|D| = 1000$, whereas the values used for K are 1,2,5 and 10. In this scenario, $|D|$ snippets or documents are required to rate each new word.

3.4.3 Affective Lexicon Creation

Overall the presented experiments, presented in the following sections are:

- Cross-validation on ANEW (ANEW-CV)
- Cross-validation on GINQ (GINQ-CV)

- Training with GINQ, evaluation on ANEW (ANEW-N)
- Training with ANEW, evaluation on GINQ (GINQ-N)

In all cases the seed words are selected from the training set (training fold in the case of cross-validation). We also conducted the same experiments using the 14 seeds used in [54]. Furthermore we used the method outlined in [54], in it's original form (which will act as a baseline) as well as variations using different similarity metrics or seeds. In the following pages we will refer to the 14 seeds as "Turney's seeds" and the linear equation without normalization coefficients as "Turney's equation".

Given a set of candidate seeds (in most cases the entire training set), we apply a simple method to select the desired seeds. It seems, looking at Turney's method, but also confirmed by our experiments, that good seeds need to have a high absolute valence rating and they should be common words. It also proved beneficial (particularly when using Turney's equation) to ensure that the seed set is as close to *balanced* (sum of seed valence is zero) as possible. Therefore our selection method starts by sorting the positive and negative seeds separately, either by their valence rating (if they have continuous valence ratings) or number of hits returned by their queries (if they have binary valence ratings). Then positive and negative seeds are added to the seed set iteratively so as to minimize the absolute value of the sum of their valence ratings, yet maximize their absolute valence ratings (or frequencies), until the required number N is reached[1].

We provide results for a wide range of N values, from 10 to 1000 seeds. Unless mentioned otherwise, the seeds are selected among the training set, therefore on cross-validation experiments the seeds are different for each fold.

The semantic similarity between each of the N features and each of the words in the test set ("unseen" words) was computed, as discussed in the previous section. Next for each value of N, the optimal weights of the linear equation system matrix in (3.3) were estimated using LMS. Finally, for each word in the test set the valence ratings were computed using (3.2) and evaluated against the ground truth.

[1] It is worth mentioning that the method is very robust to the selection of seed words. We also attempted seed selection using wrappers [28], as well as completely random (not shown here), that resulted in minimal performance differences; the estimation procedure simply adjusts the weights of seed words accordingly. However the logic behind the selection method stands and has a major effect on performance when using Turney's equation.

Table 3.3: Training sample using 10 seed words.

w_i	$v(w_i)$	a_i	$v(w_i) \times a_i$
triumphant	0.96	1.48	1.42
rape	-0.94	0.72	-0.67
love	0.93	0.57	0.53
suicide	-0.94	3.09	-2.91
paradise	0.93	1.77	1.65
funeral	-0.90	0.53	-0.48
loved	0.91	1.53	1.40
rejected	-0.88	0.50	-0.44
joy	0.90	1.00	0.90
murderer	-0.87	1.99	-1.73
w_0 *(offset)*	1	-0.06	-0.06

A toy training example using $N = 10$ features and the Google semantic relatedness hit-based metric is shown in Table 3.3. The second column $v(w_i)$ shows the manually annotated valence of word w_i, while the third column a_i shows the corresponding linear weight computed by the LMS algorithm. Their product (final column) $v(w_i) \times a_i$ is a measure of the affective "shift" of the valence of each word per "unit of similarity" to that seed word (see also (3.2)). The last row in the table corresponds to the bias term a_0 in (3.2) that takes a small positive value. Note that the coefficients a_i take positive values and are not bounded in $[0, 1]$, although similarity metrics are bounded at $[0, 1]$ and target valence values are also bounded in $[-1, 1]$. There is no obvious intuition behind the a_i scores, e.g., it is not clear why "suicide" should receive much higher weighting than "funeral". The weights might be related to the semantic and affective variance of the seed words.

The experiments using both word dataset, one for training and one for testing are also conducted using the same method. Our goal here was not only to evaluate the proposed algorithm, but also investigate whether using seeds from one manually annotated corpus can robustly estimate valence ratings in another corpus.

The following objective evaluation metrics were used to measure the performance of the affective lexicon expansion algorithm: (i) Pearson correlation between the manually

$$\underbrace{\underset{0.57}{\textit{watching}}\ \underset{0.71}{\textit{cute}}\ \underset{0.50}{\textit{puppies}}\ \underset{0.00}{\textit{makes}}\ \underset{-0.11}{\textit{me}}\ \underset{0.82}{\textit{happy}}}_{\text{[linear: 0.41, weighted average: 0.64, max: 0.82]}}$$

Figure 3.1: Example of word rating fusion, showing the per-word ratings and the phrase ratings produced by the three fusion schemes.

labeled and automatically computed valence ratings and (ii) binary classification accuracy of positive vs negative relations, i.e., continuous ratings are produced, converted to binary decisions and compared to the ground truth.

3.4.4 Sentence Level Tagging

The *SemEval 2007: Task 14* and subtitles corpora were used to evaluate the various fusion methods for turning word into sentence ratings. All unseen words in the sentence corpus are added to the lexicon using the affective lexicon expansion algorithm outlined above. The model used to create the required ratings is trained using all of the words in the ANEW corpus as training samples and N of them as seed words. Then the ratings of the words are combined to create the sentence rating using linear fusion (3.10), weighted average fusion (3.11) or non-linear max fusion (3.12). In the first experiment (labeled "all words"), all words in a sentence are taken into account to compute the sentence score. In the second experiment (labeled "content words"), only the *verbs, nouns, adjectives and adverbs* are used. To identify content words part-of-speech tagging was performed using *TreeTagger* [45]. A toy example of this method can be seen in Figure 3.1.

In order to evaluate the performance of the sentence level affective scores we use the following metrics: (i) Pearson correlation between the manually labeled and automatically computed scores and (ii) classification accuracy for the 2-class (positive, negative) problem.

3.5 Results

3.5.1 Lexicon Baseline

The baseline we will use for all affective lexicon experiments is the method proposed in [53], of which our own is a generalization. The only comparable result in the original

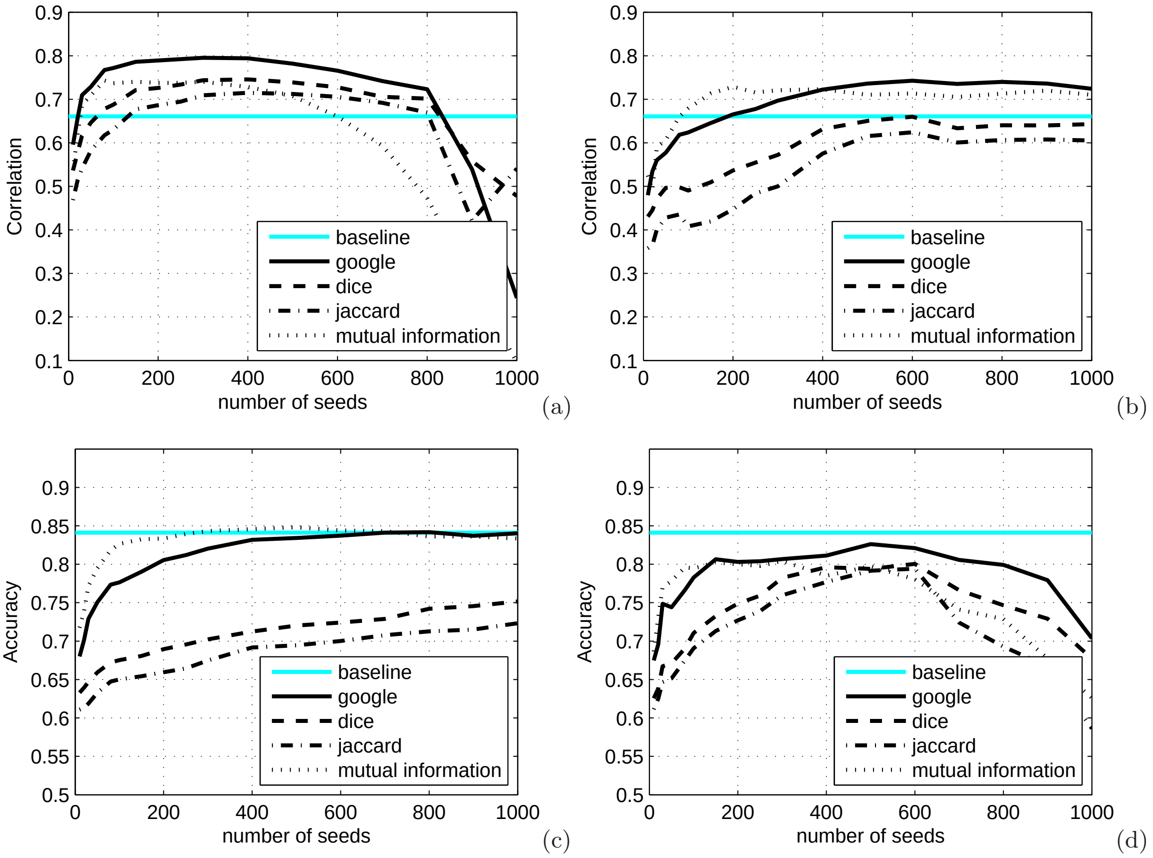


Figure 3.2: Performance of the affective lexicon creation algorithm using similarities based on AND counts. Results shown: correlation in the (a) ANEW-CV and (b) ANEW-N experiments and binary accuracy for the (c) GINQ-CV and (d) GINQ-N experiments.

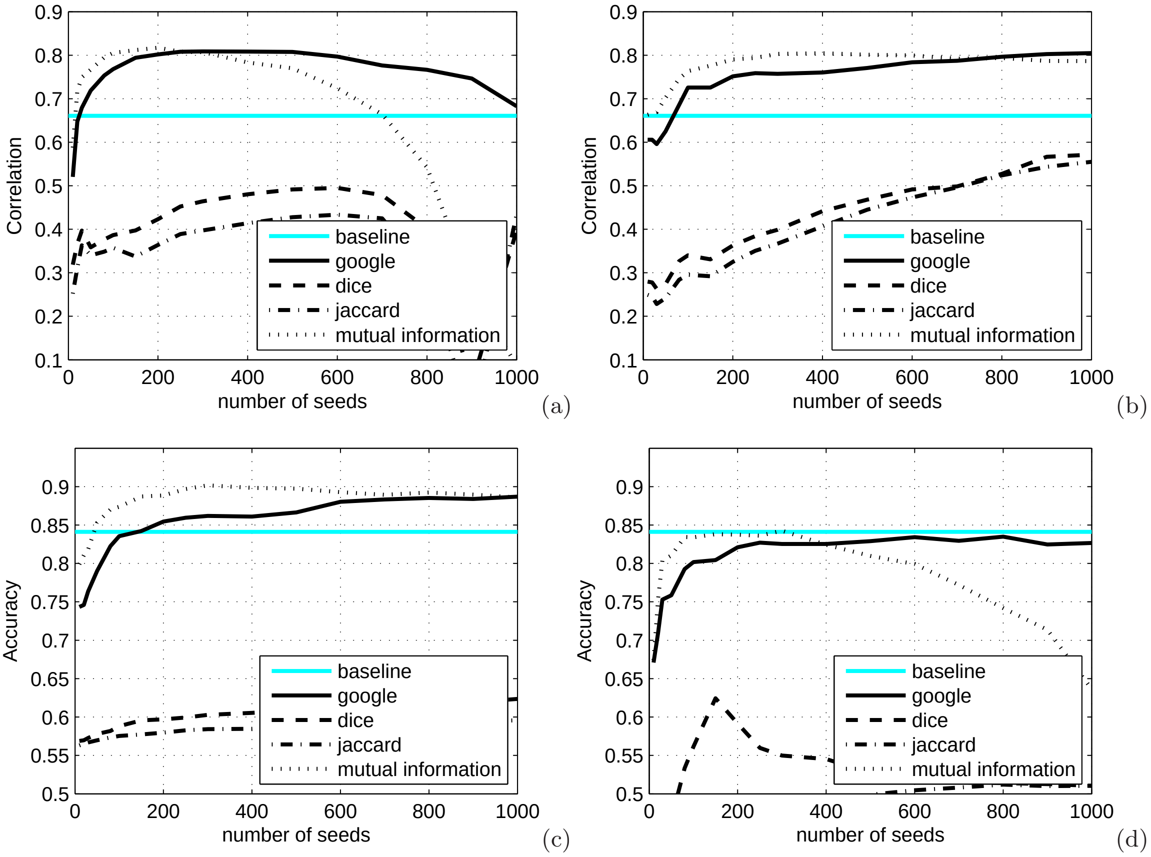


Figure 3.3: Performance of the affective lexicon creation algorithm using similarities based on NEAR counts. Results shown: correlation in the (a) ANEW-CV and (b) ANEW-N experiments and binary accuracy for the (c) GINQ-CV and (d) GINQ-N experiments.

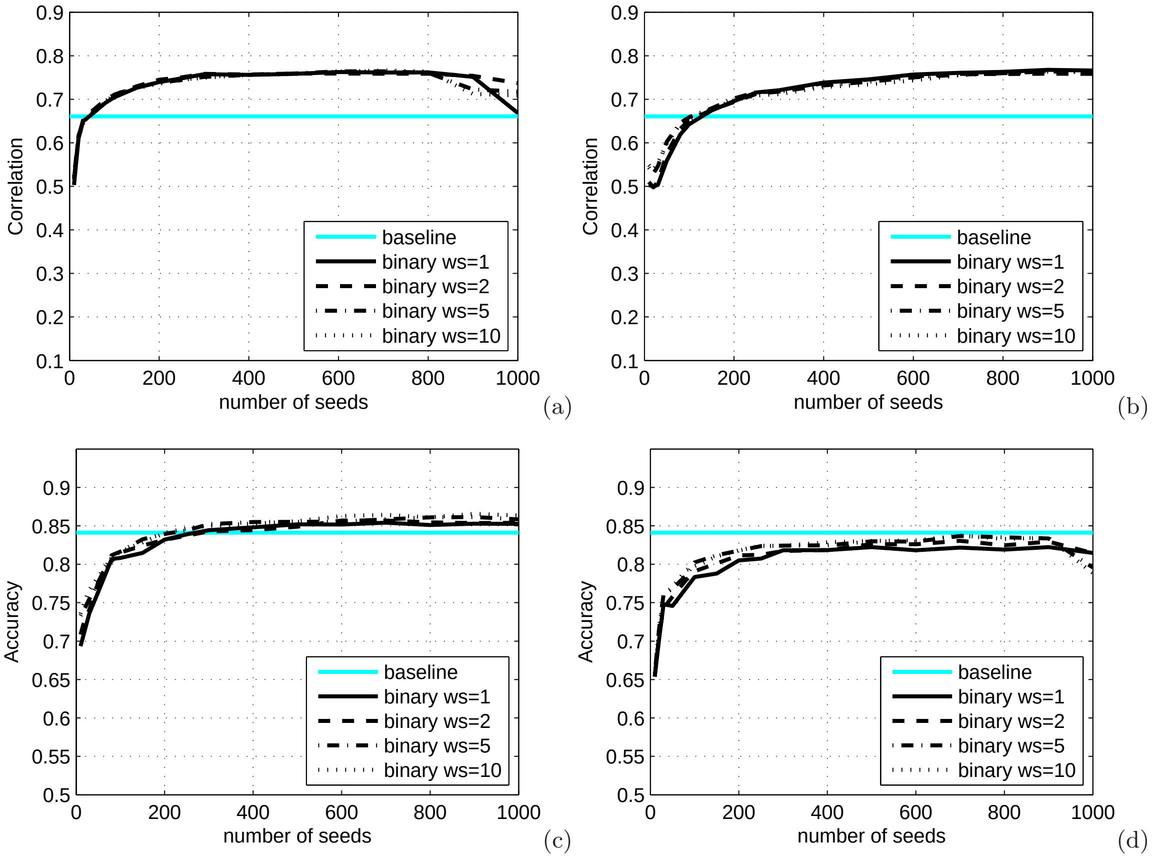


Figure 3.4: Performance of the affective lexicon creation algorithm using similarities based on IND snippets. Results shown: correlation in the (a) ANEW-CV and (b) ANEW-N experiments and binary accuracy for the (c) GINQ-CV and (d) GINQ-N experiments.

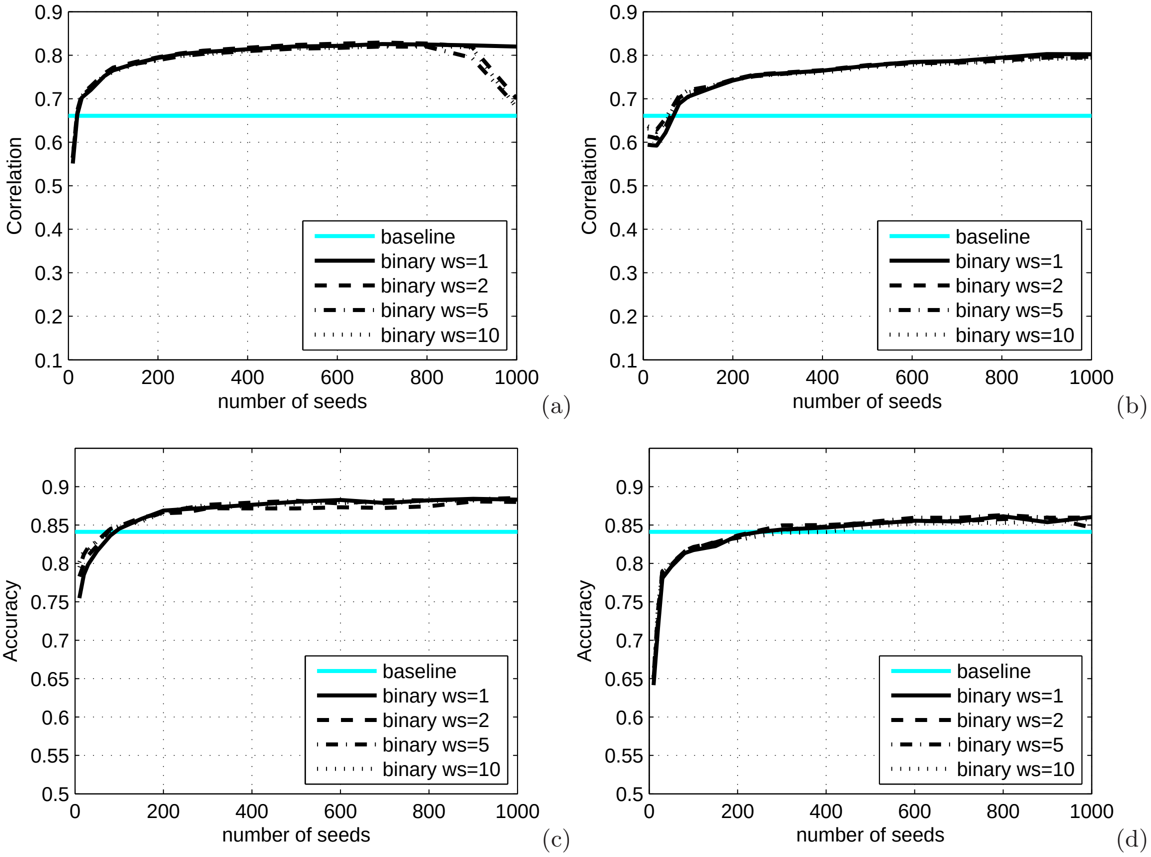


Figure 3.5: Performance of the affective lexicon creation algorithm using similarities based on IND documents. Results shown: correlation in the (a) ANEW-CV and (b) ANEW-N experiments and binary accuracy for the (c) GINQ-CVand (d) GINQ-Nexperiments.

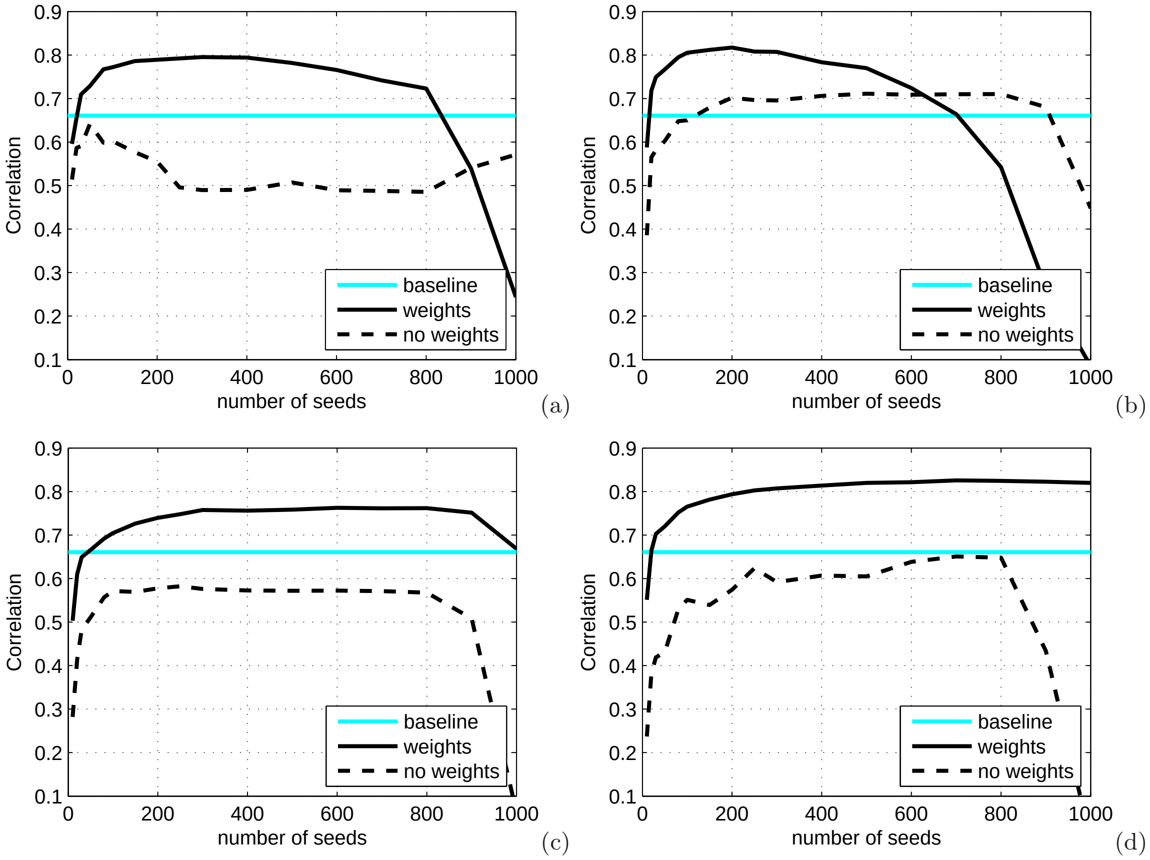


Figure 3.6: Performance of the affective lexicon creation algorithm on the ANEW-CV experiment comparing the existence or absence of weights a_i. Results shown for the correlation between the automatically computed and manually annotated scores. The similarity metrics used are the best performing ones based on (a) AND counts, (b) NEAR counts, (c) IND snippets and (d) IND documents.

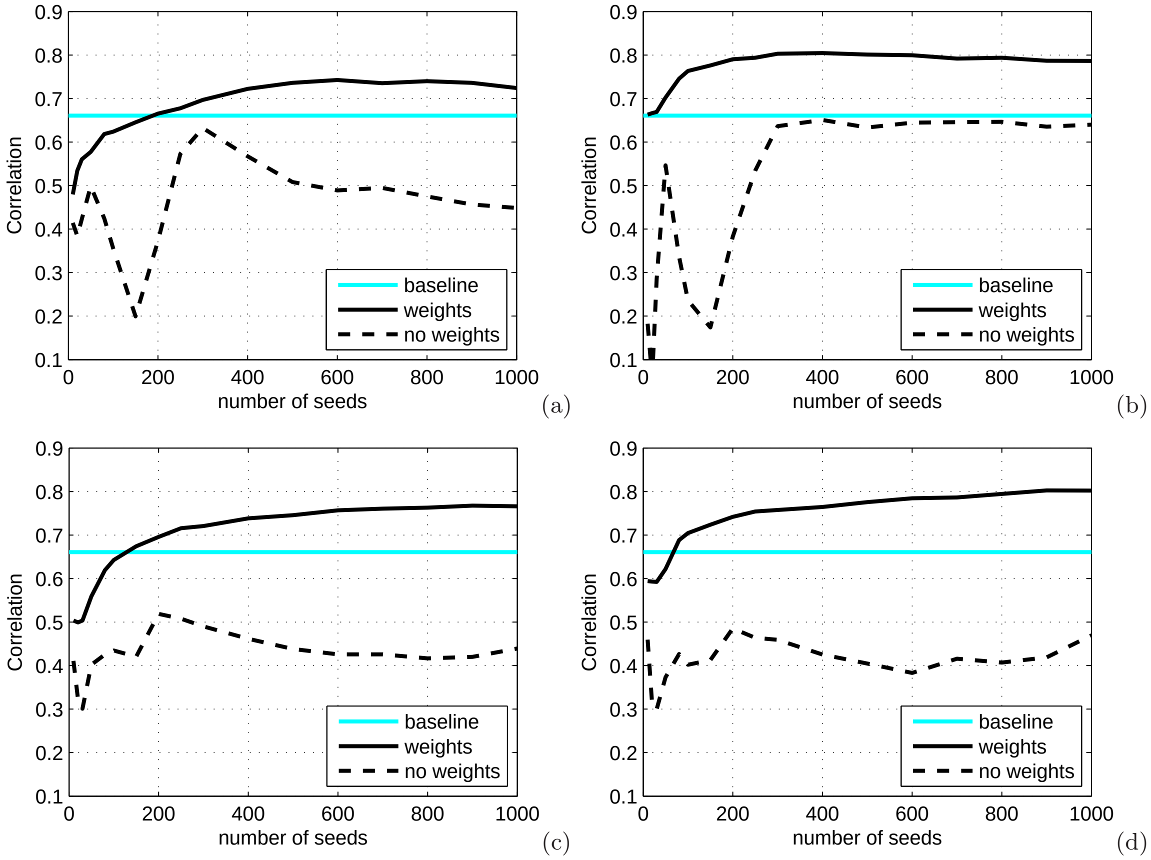


Figure 3.7: Performance of the affective lexicon creation algorithm on the ANEW-N experiment comparing the existence or absence of weights a_i. Results shown for the correlation between the automatically computed and manually annotated scores. The similarity metrics used are the best performing ones based on (a) AND counts, (b) NEAR counts, (c) IND snippets and (d) IND documents.

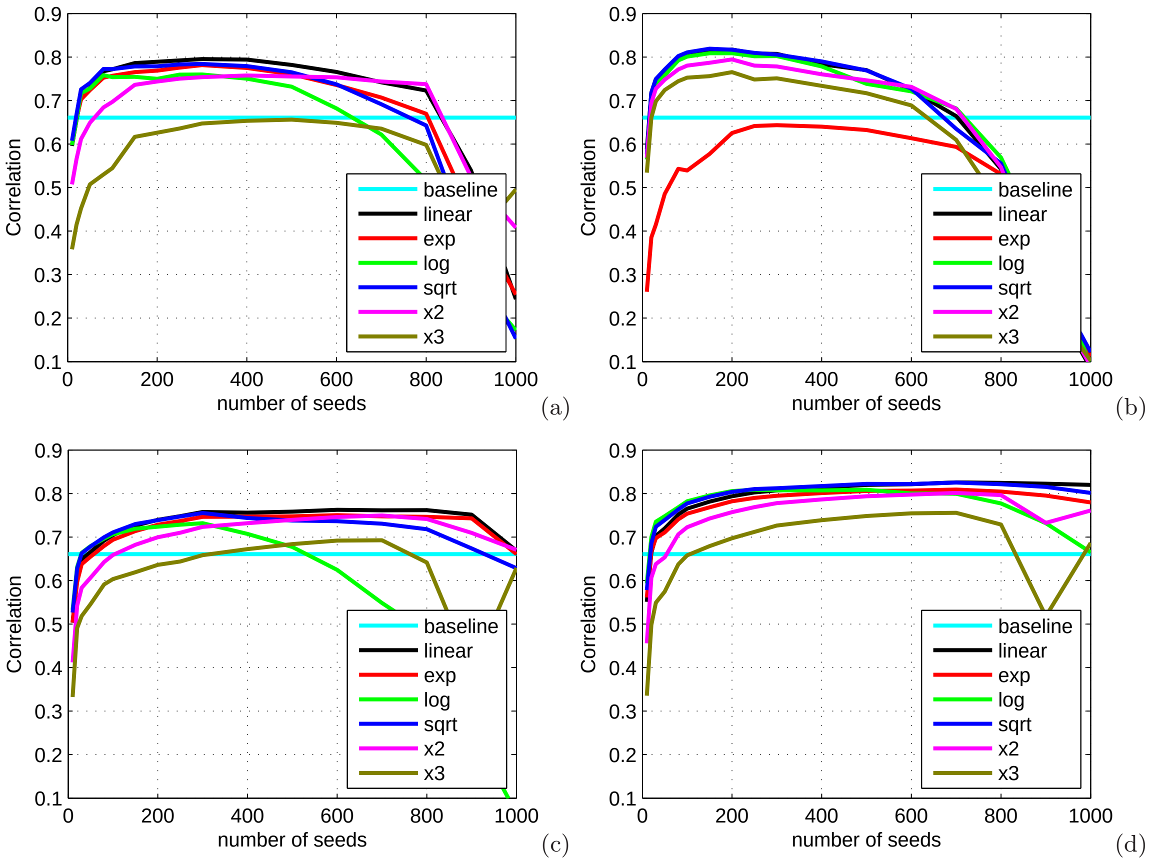


Figure 3.8: Performance of the affective lexicon creation algorithm on the ANEW-CV experiment, comparing different kernels $f(\cdot)$. Results are shown for the correlation between the automatically computed and manually annotated scores. The similarity metrics used are the best performing ones based on (a) AND counts, (b) NEAR counts, (c) IND snippets and (d) IND documents.

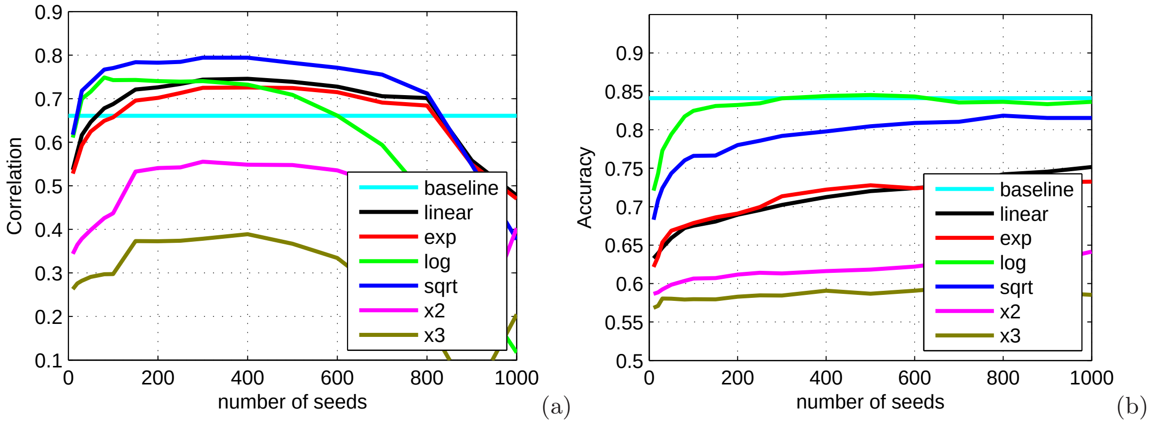


Figure 3.9: Performance of the affective lexicon creation algorithm using the DICE similarity metric based on AND counts and comparing kernels $f(\cdot)$. Results are shown for (a) the correlation between the automatically computed and manually annotated scores of the ANEW-CV experiment and (b) binary accuracy of the GINQ-CV experiment.

paper is that of binary accuracy for the GINQ dataset (82.8%), so in order to have a more meaningful comparison we repeated the GINQ experiment and added the ANEW experiment for the method. The results achieved were: 0.66 correlation and 0.82 binary accuracy in the ANEW dataset and 0.84 accuracy in the GINQ dataset, beating the score in the original paper. These three scores are our baselines and appear as cyan threshold lines on every performance graph.

3.5.2 Affective Lexicon Creation

Similarity metric selection The first set of results compare the different performance metrics evaluated for each data type, using an equation with a linear kernel. In Figure 3.2 the performance achieved by all performance metrics based on AND counts is shown, with Google semantic relatedness being the clear winner. In Figure 3.3, the performance achieved by all performance metrics based on NEAR counts is shown. Here the differences are smaller, but mutual information is the best performing metric. In both cases the Jaccard and Dice coefficients fail to achieve good results, their problem is one of scaling as we will see later.

In Figure 3.4 and Figure 3.5 performance is shown when using contextual similarities based on snippets and documents respectively. In both cases the parameter, which is

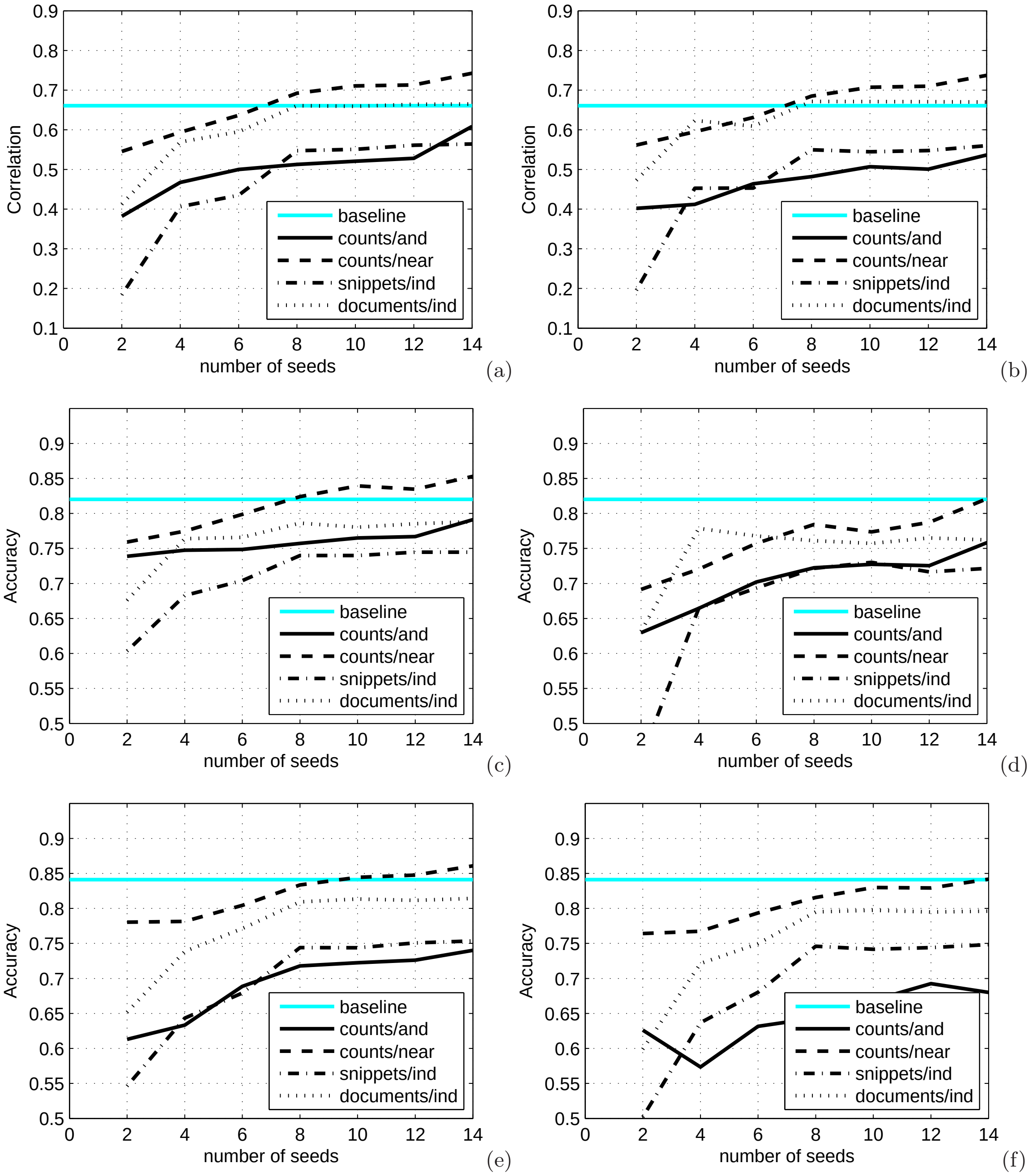


Figure 3.10: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the similarity metric type, using the best performing similarity per type and the 14 Turney seeds. Results shown: the correlation between the automatically computed and manually annotated scores in the (a) ANEW-CV and (b) ANEW-N experiments, accuracy for the (c) ANEW-CV and (d) ANEW-N experiments. and binary accuracy for the (e) GINQ-CV and (f) GINQ-N experiments.

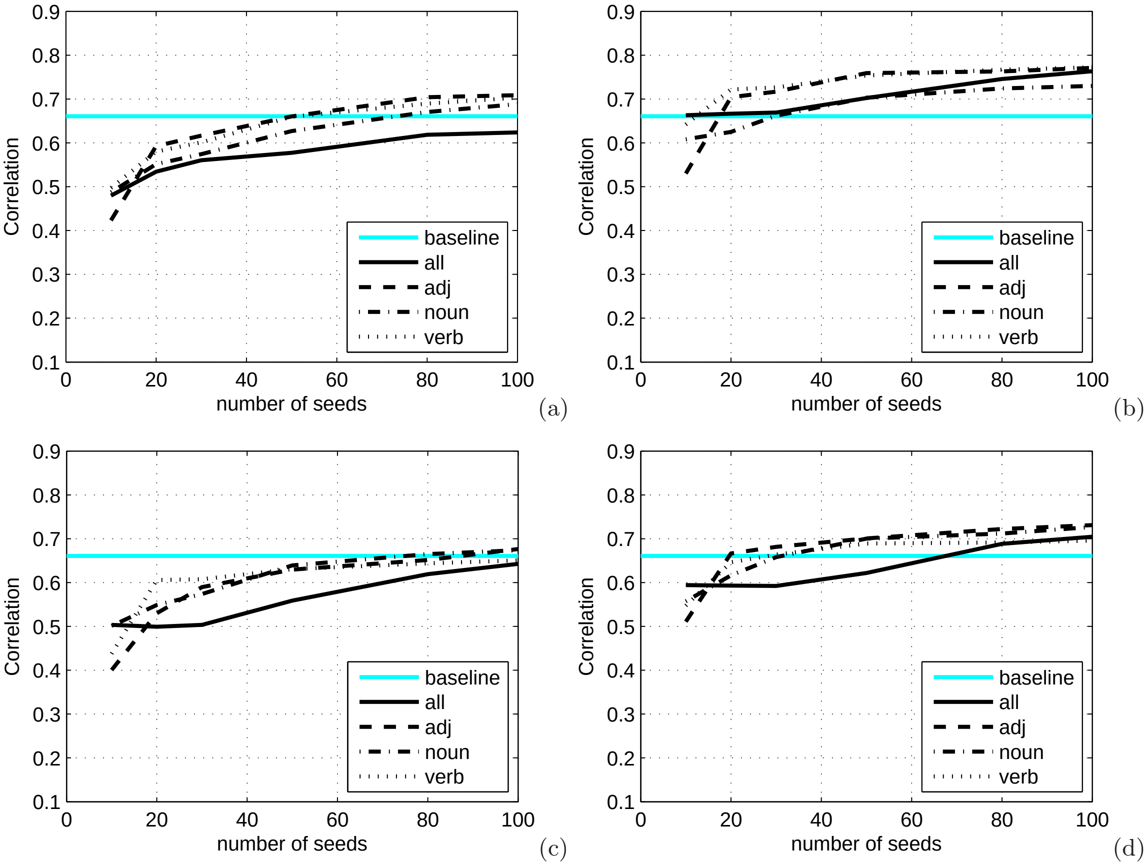


Figure 3.11: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the part of speech tag of the seed words, using the best performing similarity metric per type. Results are shown (using the manually annotated scores as the ground truth) for the correlation between the automatically computed and manually annotated scores in the ANEW-N experiment, when using similarity metrics based on (a) AND counts, (b) NEAR counts, (c) IND snippets and (d) IND documents.

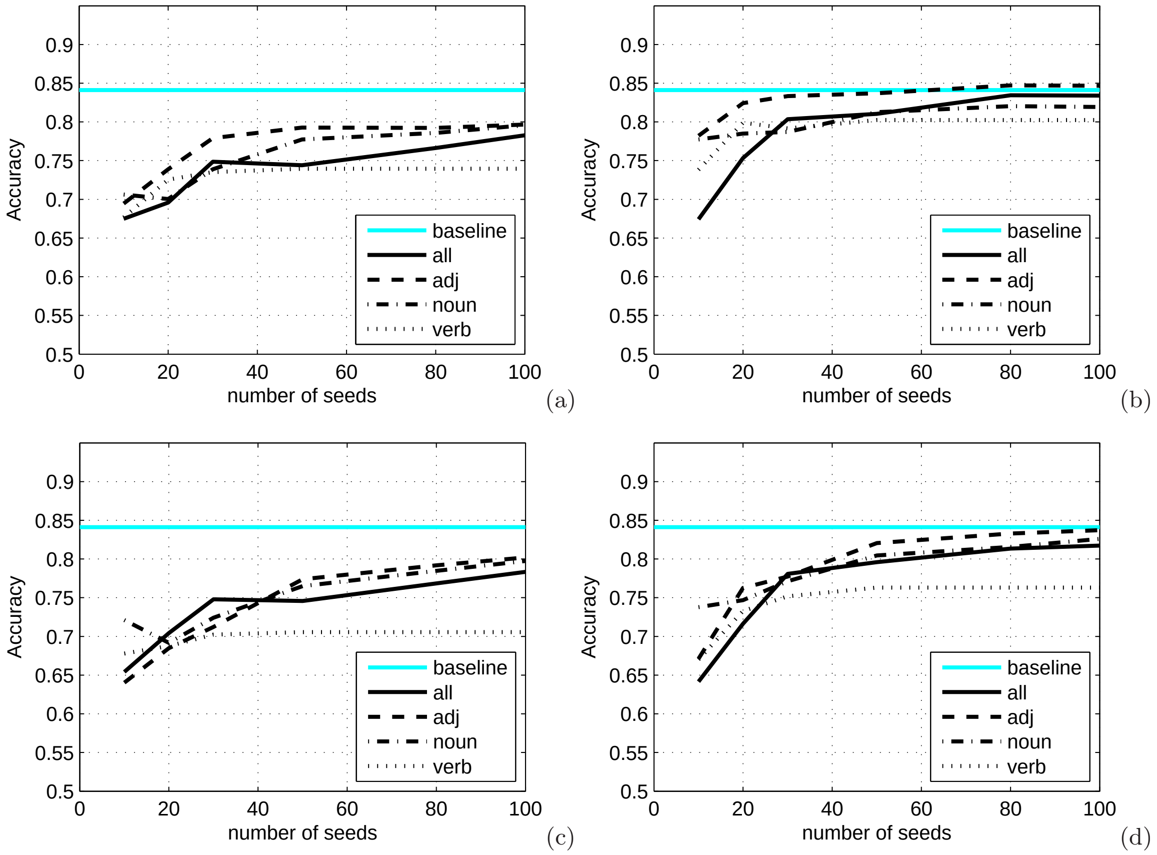


Figure 3.12: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the part of speech tag of the seed words, using the best performing similarity metric per type. Results are shown (using the manually annotated scores as the ground truth) for the correlation between the automatically computed and manually annotated scores in the GINQ-N experiment, when using similarity metrics based on (a) AND counts, (b) NEAR counts, (c) IND snippets and (d) IND documents.

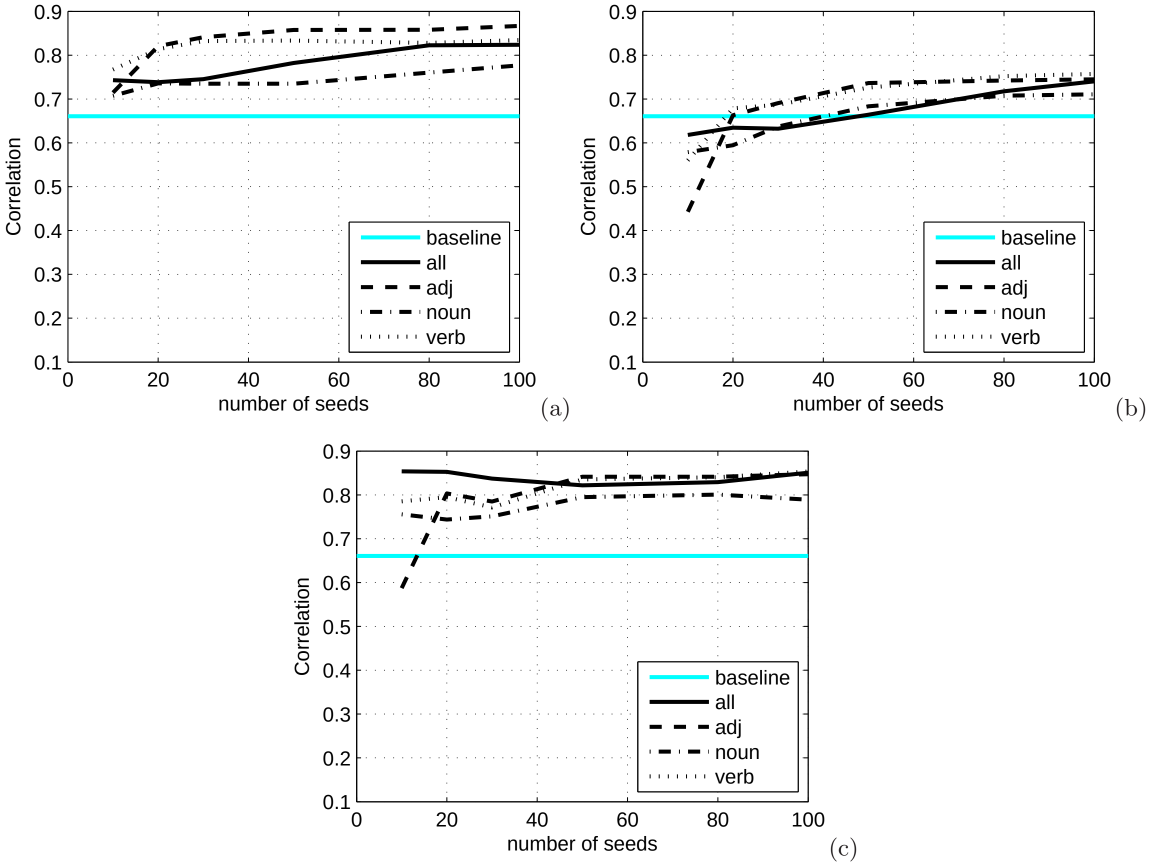
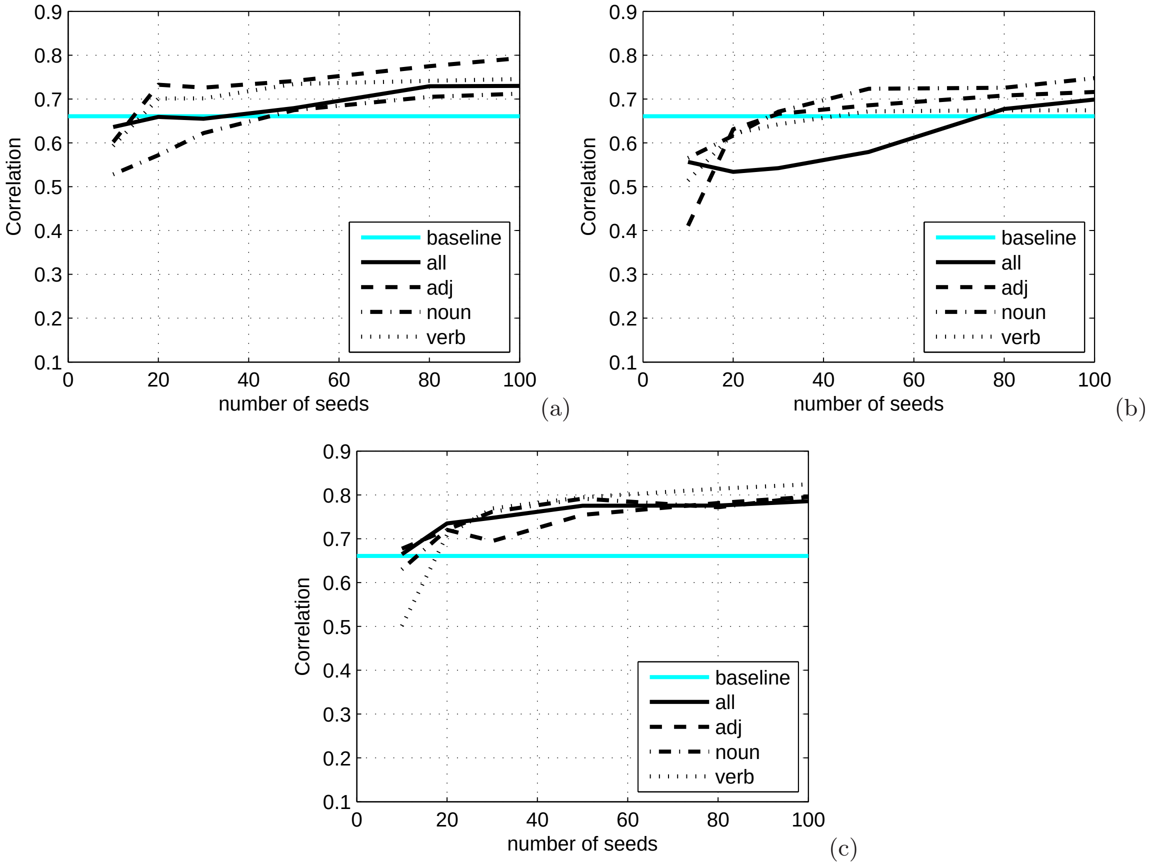


Figure 3.13: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the part of speech tag of the seed words, using the best performing similarity metric per type. Results are shown (using the manually annotated scores as the ground truth) for the correlation between the automatically computed and manually annotated scores in the ANEW-N experiment (a) adjectives, (b) nouns and (c) verbs, when using similarity metrics based on NEAR counts.



(a) (b) (c)

Figure 3.14: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the part of speech tag of the seed words, using the best performing similarity metric per type. Results are shown (using the manually annotated scores as the ground truth) for the correlation between the automatically computed and manually annotated scores in the ANEW-N experiment (a) adjectives, (b) nouns and (c) verbs, when using similarity metrics based on IND documents.

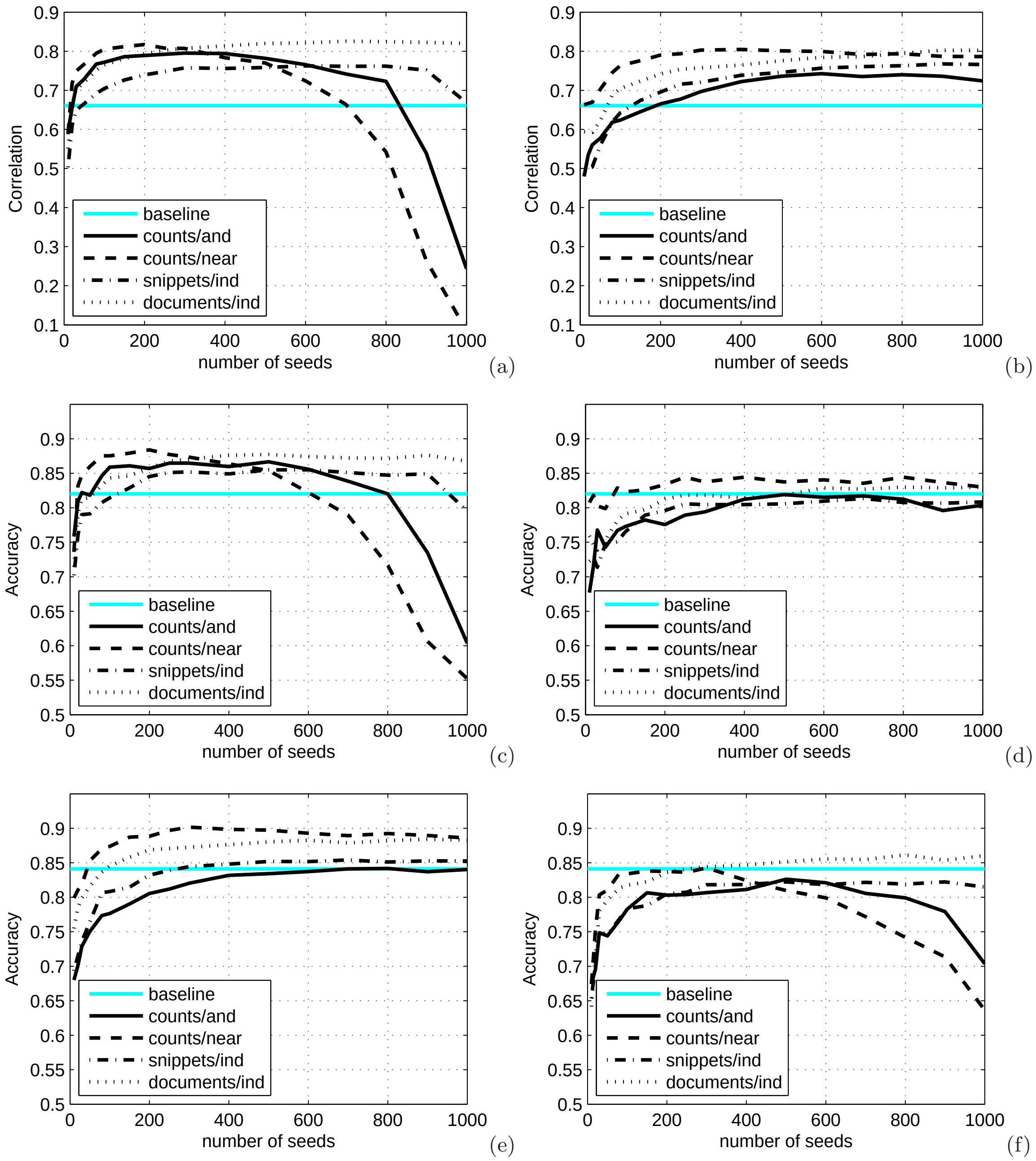


Figure 3.15: Performance of the affective lexicon creation algorithm as a function of the number of seeds words and the similarity metric type, using the best performing similarity per type. Results are shown (using the manually annotated scores as the ground truth) for: the correlation between the automatically computed and manually annotated scores in the ANEW-CV (a) and ANEW-N (b) experiments, accuracy for the ANEW-CV (c) and ANEW-N (d) experiments. and binary accuracy for the GINQ-CV (e) and GINQ-N (f) experiments.

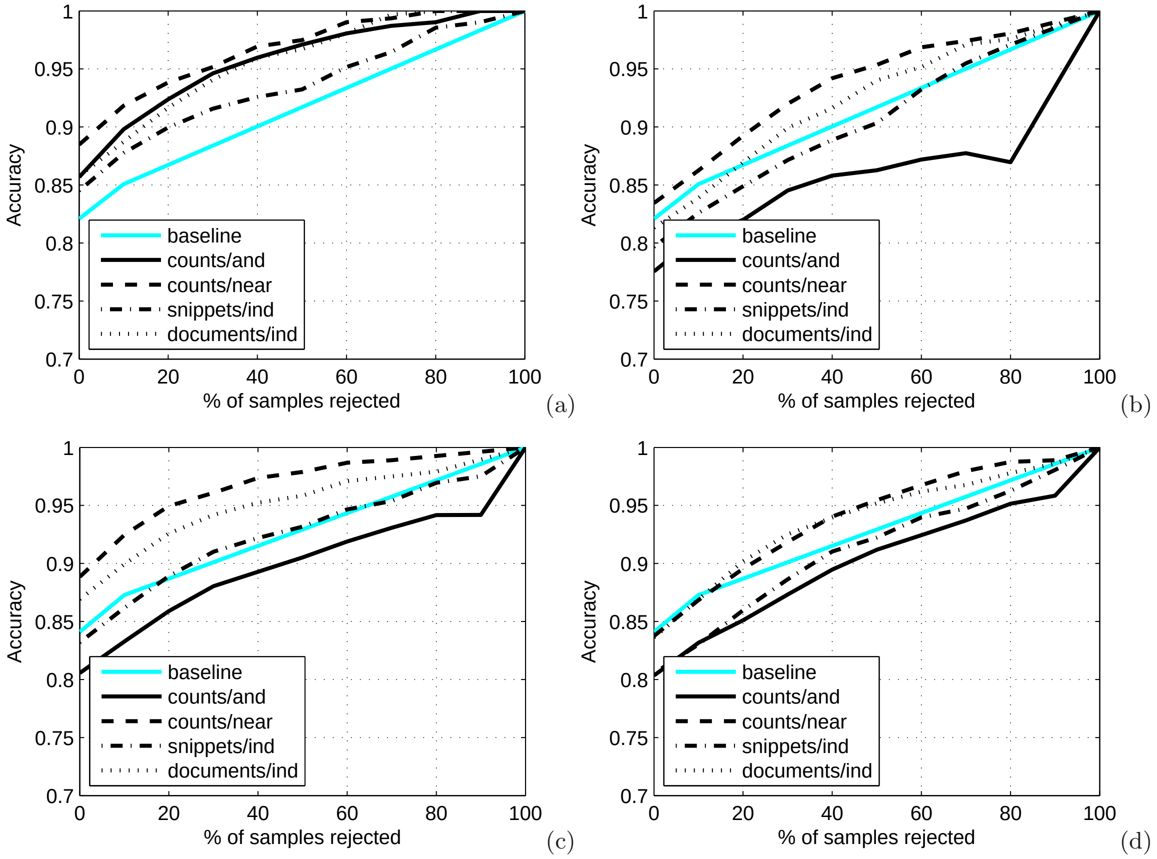


Figure 3.16: Rejection-Accuracy graphs for the (a) ANEW-CV, (b) ANEW-N, (c) GINQ-CV and (d) GINQ-N tasks. Accuracy is presented as a function of the percentage of samples we disregard as “unsure”. The similarity metrics used are the best per type.

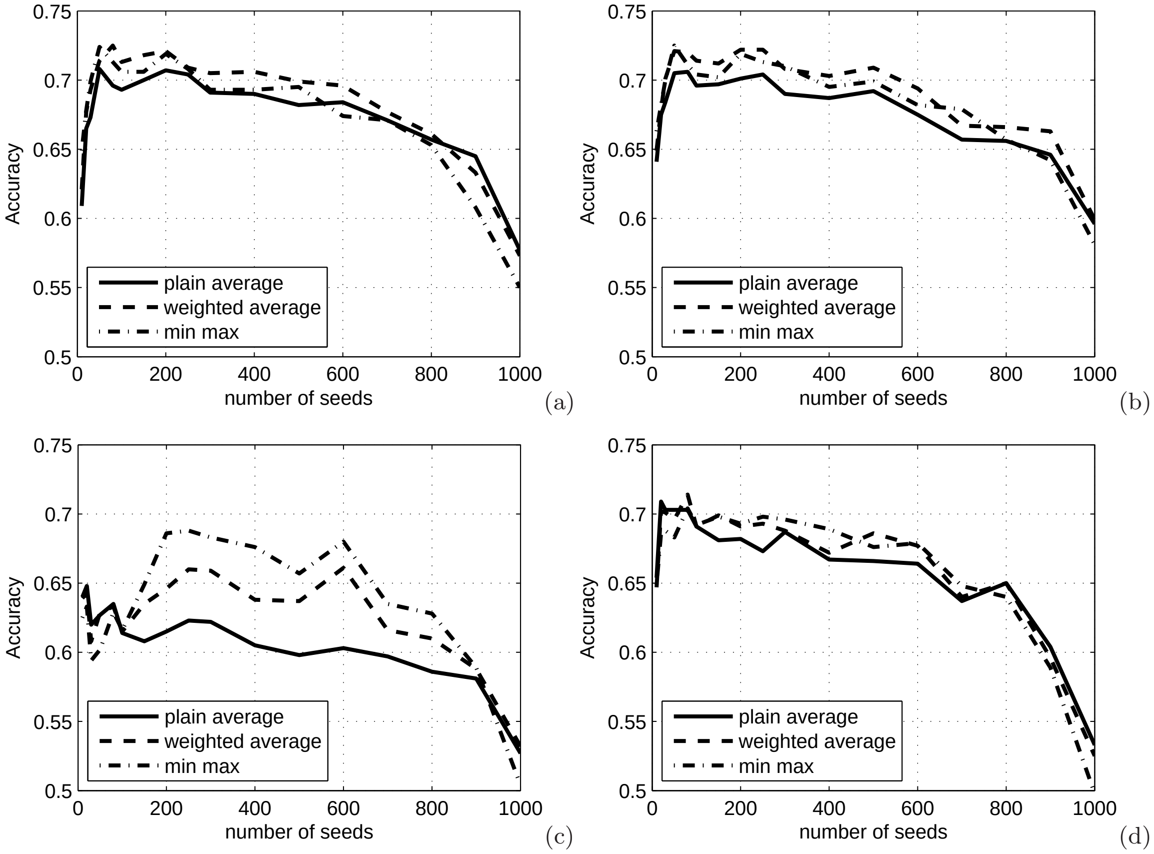


Figure 3.17: Performance of the phrase rating creation algorithm as a function of the number of seeds words and the word fusion method. Results are shown (using the manually annotated scores as the ground truth) for the binary accuracy, when using AND counts, Google semantic relatedness and (a) all or (b) only content words and using NEAR counts, Mutual Information and (c) all or (d) only content words.

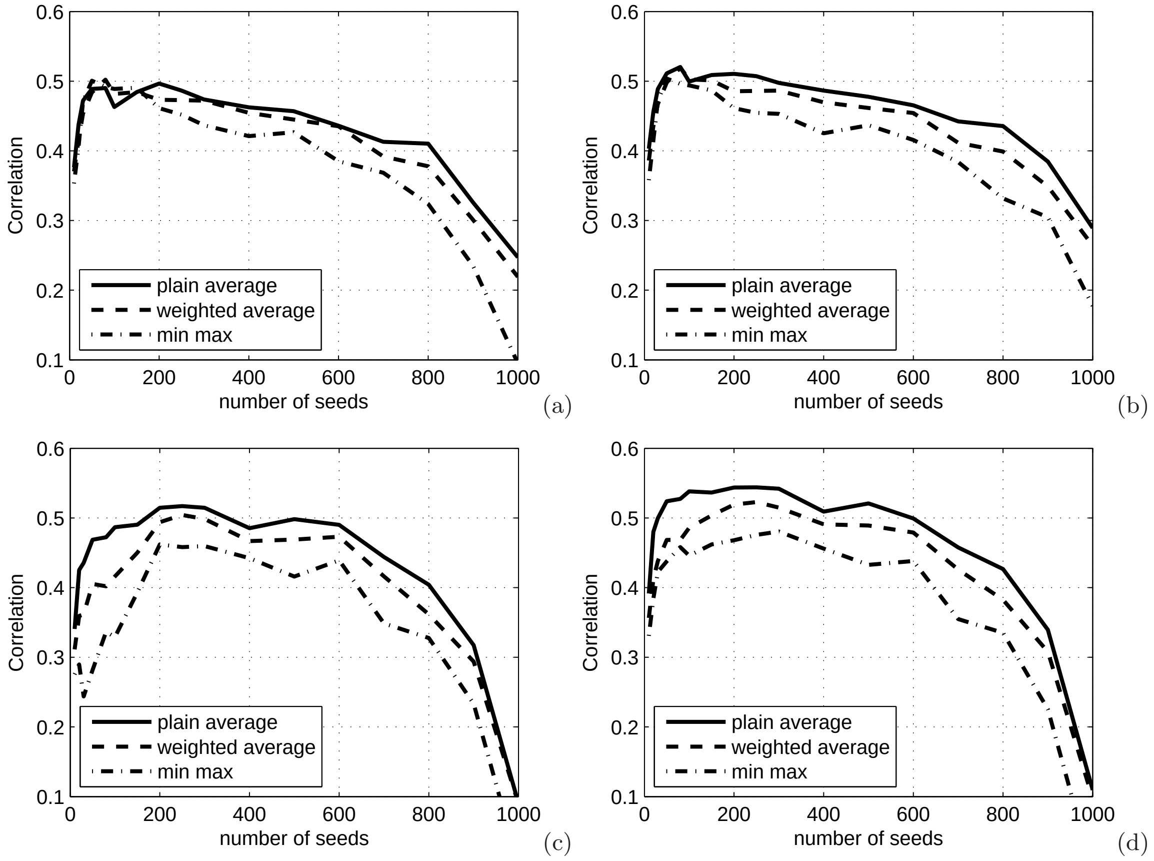


Figure 3.18: Performance of the phrase rating creation algorithm as a function of the number of seeds words and the word fusion method. Results are shown (using the manually annotated scores as the ground truth) for the correlation to the ground truth, when using AND counts, Google semantic relatedness and (a) all or (b) only content words and using NEAR counts, Mutual Information and (c) all or (d) only content words.

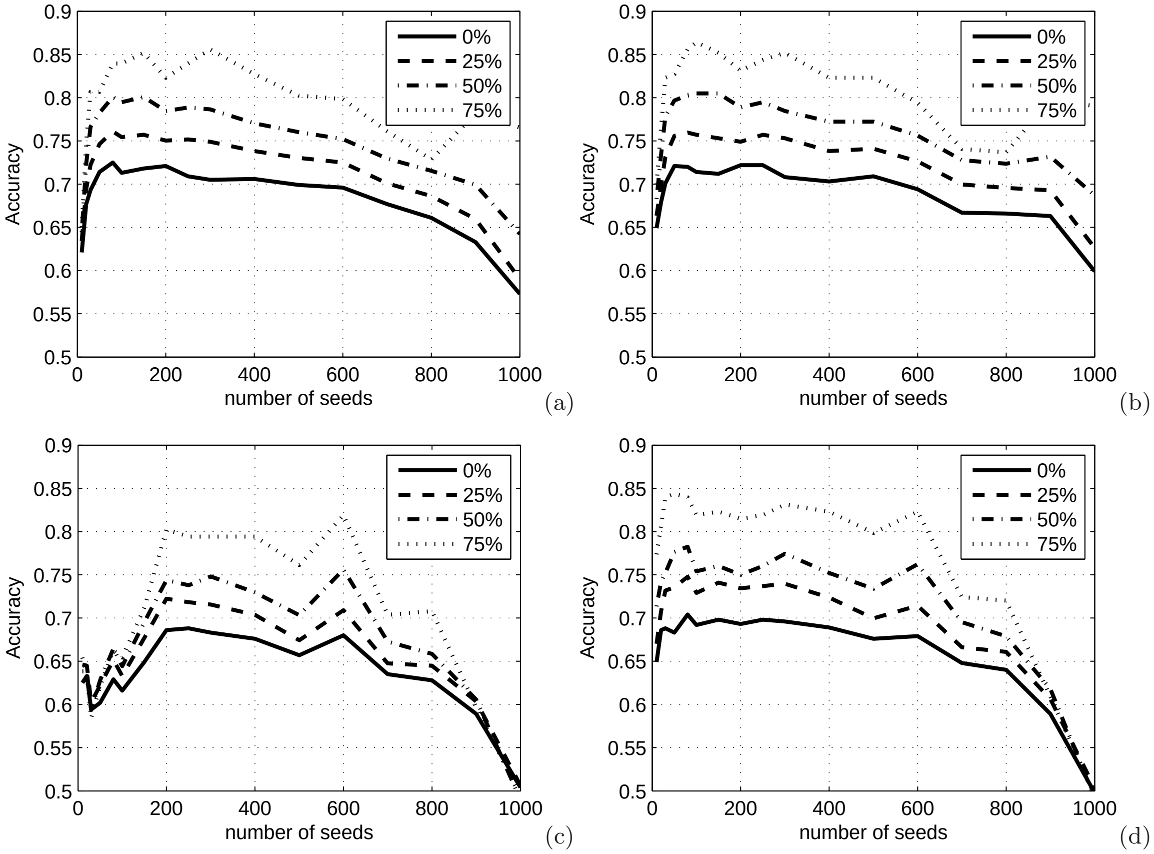


Figure 3.19: Performance of the phrase rating creation algorithm as a function of the number of seeds words and the sample rejection percentage. Results are shown (using the manually annotated scores as the ground truth) for the binary accuracy, when using AND counts, Google semantic relatedness, weighted average fusion and (a) all or (b) only content words and using NEAR counts, Mutual Information, min-max fusion and (c) all or (d) only content words.

the size of the context window, seems to have no appreciable effect. For both of them we choose the window size of 1 as the "best".

Note the drop in performance past 500 seeds or so, when the ANEW dataset is used for training and how it disappears when the GINQ corpus is used for training. This indicates that we need at least 2-3 training samples per seed word to avoid over-fitting. The effect is much less pronounced when contextual similarities are used. In terms of absolute performance, almost all variations can easily improve on the baseline when tasked with creating accurate continuous ratings, but that is not true for the binary classification task.

All graphs beyond, unless mentioned otherwise, use these 4 top performing metrics per data source.

Effect of weights To gauge the significance of weights a_i we simply compare the performance of a linear kernel equation with and without these weights. The results are shown in Figure 3.7 and Figure 3.6 for the ANEW-N and ANEW-CV experiments respectively. It is clear that the addition of weights has a profound effect on performance. Also interesting is the fact that when using NEAR counts and mutual information (Turney's model with different seeds) performance eventually gets close or over that achieved with the original 14 seeds, even without the use of training/weights.

Kernels comparison Results shown so far used a linear equation kernel $f()$, which in effect is a lack of kernel. Figure 3.8 shows a comparison of the kernels in Table 3.2 when used in conjunction with the best performing similarity metrics in the ANEW-CV experiment. The overall results show little reason to use a more complex kernel when using a well behaving similarity metric, no kernel can produce a result that is appreciably higher than those achieved by the linear kernel. However that changes when using less suitable similarity metrics. In previous steps we eliminated the Dice coefficient since it failed to achieve high performance, however that can change drastically when applying kernels, as shown in Figure 3.9. With a logarithmic or exponential kernel the Dice coefficient becomes competitive with the top performers. However, suboptimal similarity metrics combined with kernels still fail to outperform the best similarity metrics combined with the linear kernel.

Seeds In all of our experiments the seed words are selected from the training set, with all words in the training set being candidates. In this section we examine performance using different seeds.

Figure 3.10 shows performance achieved when we combine the best performing similarity metrics and the linear kernel equation with the 14 seeds used by Turney, as shown in Table 3.1. Probably the most interesting thing in the graphs is how those in the left column (cross-validation) are almost identical to those in the right column, particularly when the GINQ dataset is evaluated. Perhaps the reason is that the 14 Turney seeds all belong to the GINQ dataset (but not to the ANEW dataset). Another reason could be that the effect of training simply becomes more important as the number of seeds increases. In terms of absolute performance, adding weights does improve on the original method, but only when using mutual information based on NEAR counts. The performance achieved in the GINQ-N experiment (0.84 accuracy) is the highest ever achieved for this task using these 14 seeds.

An obvious modification to the seed selection algorithm would be to take into account the candidates' part of speech (POS) tags and select only those with specific tags: after all, Turney's seeds are all adjectives. To do this we gather all possible POS tags per word from WordNet. Each word may have multiple possible POS tags (Turney's seeds do), but to enhance any differences we only take into account words with no part of speech ambiguity: words that only have one possible part of speech tag. Figure 3.11 and Figure 3.12 show the effect of using seeds of only one part of speech tag in the ANEW-N and GINQ-N experiments respectively. In most cases the result is a net gain over using words of any POS tag as seeds. It should be noted that these graphs only go up to 100 seeds, since there are simply not enough seed candidates, so this gain exhibited when using a low number of seeds is more than offset by the availability of more candidate seeds. Also note the relative performance: in most cases adjectives are the best seeds, however their difference with nouns is not necessarily significant, depending on the similarity metric. This indicates that the similarity metric used and, possibly the POS tags of the evaluated words, affect the optimal choice of seed word POS tag. We perform partial evaluation, focusing on the performance exhibited when the words being rated have specific POS tags (again the words examined have only one possible POS tag). In Figure 3.13 and Figure 3.5 the performance of different types of seeds at evaluating different types of target words is shown, when the similarity metric used is based on NEAR counts and IND documents respectively. For contextual similarities, the best performance is achieved when the seeds have the same POS tag as the targeted words: the best seeds when it comes to predicting ratings for nouns are other nouns. That is clearly not the case when using NEAR counts, where nouns are better defined by adjectives.

Overall, the evidence suggest creating a more complex hierarchical model, utilizing different seeds depending on the input word. There are however practical issues in such an implementation, most importantly that we do not know the POS tags of annotated lexicon words nor do we have any way to handle words with multiple possible POS tags. Regarding the use of seeds bearing only one specific POS tag, it doesn't seem worth it for these experiments: the net gain when using few seeds is not substantial enough and is offset at higher seed numbers due to simply having far less seed candidates available, so models without a POS tag limitation eventually achieve higher performance. However it is an alternative if resources are more available than corpora/queries.

Overall Figure 3.15 shows the overall performance of the proposed method per task when using the best performing similarity metrics and the linear kernel equation. Overall NEAR counts (mutual information) and IND documents (contextual) produce the best results, being clearly better than AND counts (google semantic relatedness) and IND snippets respectively.

In the case of counts this shows that proximity plays a role in accurately determining similarity, though is unclear how important it is and what the "threshold" may be. Perhaps a maximum distance of 10 words is sub-optimal, perhaps an approach taking into account text structure may be better, such as requiring words to co-occur within a paragraph. However, unlike the results reported in [53] and [54] the method does not fall apart when using AND queries: the results are worse, but not bad. Documents performing better than snippets seems more straightforward: the corpus created by documents is simply much bigger than the one created by snippets.

Looking at absolute performance numbers, accuracy and correlation reach maximums of 0.82/0.88 and 0.81/0.84 for the ANEW-CV and ANEW-N experiments respectively. Unfortunately no comparable numbers exist in literature using the ANEW dataset, though of course they are much higher than those achieved using Turney's method. Accuracy in the GINQ corpus reaches 90% for the GINQ-CV experiment,

News corpus, Correlation							
		All words			Content words		
		avg	w.avg	max	avg	w.avg	max
AND	dice	0.45	0.44	0.39	0.48	0.45	0.42
	jaccard	0.44	0.42	0.36	0.48	0.44	0.41
	google	**0.50**	0.47	0.46	**0.51**	0.49	0.46
	PMI	0.45	0.40	0.39	0.47	0.41	0.39
NEAR	dice	0.19	0.11	0.03	0.29	0.19	0.19
	jaccard	0.16	0.10	0.03	0.26	0.19	0.19
	google	0.43	0.33	0.31	0.51	0.45	0.48
	PMI	**0.51**	0.49	0.46	**0.54**	0.52	0.47

News corpus, Accuracy							
		All words			Content words		
		avg	w.avg	max	avg	w.avg	max
AND	dice	0.66	0.66	0.67	0.69	0.70	0.70
	jaccard	0.61	0.62	0.63	0.68	0.68	0.67
	google	0.71	**0.72**	0.72	0.70	**0.72**	0.72
	PMI	0.66	0.69	0.68	0.68	0.68	0.68
NEAR	dice	0.50	0.50	0.50	0.55	0.56	0.56
	jaccard	0.49	0.49	0.49	0.52	0.52	0.53
	google	0.67	0.65	0.66	0.68	0.72	**0.72**
	PMI	0.62	0.65	**0.69**	0.68	0.69	0.69

Table 3.4: Correlation and Classification Accuracy on the SemEval Dataset for all metrics and fusion schemes.

which is in fact lower than the 91.3% reached in [52] and 93.1% reached in [22] and 86% for the GINQ-N experiment which is significantly better than the previous best in literature, the 82.8% in [53].

Figure 3.16 shows accuracy-rejection graphs for the 4 tasks. In all cases we use our method to create ratings for the entire test corpus, then disregard the words that have the lowest absolute valence (closer to zero), effectively taking the absolute rating as a confidence measure. These graphs show accuracy as a function of the percentage of samples that are ignored. Predictably, given the equivalent accuracy and much higher correlation, our method performs very well in these as well.

Subtitles corpus, Correlation							
		All words			Content words		
		avg	w.avg	max	avg	w.avg	max
AND	dice	0	0	0	0.02	0.02	0.02
	jaccard	0	0	0.01	0.02	0.02	0.02
	google	**0.05**	0.04	0.04	**0.06**	0.06	0.05
	PMI	0.04	0.03	0.04	0.05	0.05	0.05
NEAR	dice	-0.02	-0.02	-0.02	-0.03	-0.03	-0.04
	jaccard	-0.02	-0.01	-0.02	-0.02	-0.02	-0.03
	google	0.02	0.02	-0.01	0.02	0.02	-0.02
	PMI	0.04	**0.04**	0.03	0.04	**0.04**	0.03
Subtitles corpus, Accuracy							
		All words			Content words		
		avg	w.avg	max	avg	w.avg	max
AND	dice	0.56	0.56	0.55	0.56	0.56	0.55
	jaccard	0.57	0.57	0.57	0.57	0.57	0.57
	google	**0.57**	0.57	0.56	**0.58**	0.57	0.56
	PMI	0.57	0.56	0.56	0.57	0.56	0.56
NEAR	dice	0.6	0.6	0.6	0.6	0.6	0.6
	jaccard	**0.60**	**0.60**	0.6	0.6	**0.60**	0.6
	google	0.53	0.51	0.51	0.53	0.51	0.5
	PMI	0.59	0.59	0.59	0.59	0.59	0.59

Table 3.5: Correlation and Classification Accuracy on the Subtitle Dataset for all metrics and fusion schemes.

3.5.3 Sentence Level Tagging

Figure 3.17 and Figure 3.18 show the performance of the sentence level affective tagging algorithm on the SemEval corpus, as binary accuracy and correlation respectively, versus the number of seed words used to create the required word ratings and the fusion scheme used to combine them. Results are shown for the case when all words are taken into account and the case where only content words are selected. The similarity metrics used to create the needed word ratings are hit-based. Overall, selecting only the content words has a small positive effect on correlation, however that translates into a big boost in terms of accuracy when using NEAR counts. This indicates a bias issue, which is resolved removing non-content words. In terms of fusion schemes, the plain average produces the highest correlation scores, while weighted average does better in terms of accuracy. Similarly, when comparing the performance of AND and NEAR counts, AND counts produce higher accuracy, while NEAR counts perform better with regards to correlation. Given how small these differences are, they are likely to be caused by each method's accuracy in classifying sentences with very low absolute valence. To examine, we calculate accuracy while ignoring sentences with an actual absolute valence rating that puts them in the lowest $x\%$, where $x \in \{25, 50, 75\}$. The results are shown in Figure 3.19. As expected, performance improves if we only look at sentences with a higher absolute affective score. In this case there is a clear winner, the method using AND counts provides better results with the increased rejection. Table 3.4 shows performance for all similarity metrics and fusion schemes evaluated, when the number of seeds used to created the word ratings is 200. The relative performance of the various similarity metrics mirrors their performance in word level tests, with Google semantic relatedness performing best when using AND counts and mutual information performing best when using NEAR counts. Deciding between these two is less straightforward. Regardless, the achieved correlation of 0.54 is higher than any previous in literature, with the best system in [47] achieving 0.5. The binary accuracy of 0.72 is much higher than the 0.62 reported in [51] and the 0.66 in [36], higher than the 0.71 reported in [37] using a very complex compositional model and almost as high as the 0.728 achieved in [10] while using 10-fold cross-validation, manually annotated words and multiple resources to estimate the effect of modifiers and negations.

All in all, the results are encouraging, considering that our proposed method uses very little prior knowledge, no linguistic rules and rather simple fusion schemes. Even including non content words has the very minimal effect of adding a slight bias.

Table 3.5 shows performance in the subtitle dataset. Overall the results are very bad, barely beating randomness. The poor results show the added complexity of this task compared to unimodal polarity detection in text. More likely it points to the significance of factors we ignored: interactions across sentences and across modalities. It is reasonable to assume that context acts as a modifier on the affective interpretation of each utterance. Furthermore, here, it is not just lexical context that contributes to the ground truth, but also multimedia context: from the voice tone of the actor, to his facial expression, to the movie's setting.

3.6 Conclusions

We proposed an affective lexicon creation/expansion algorithm that estimates a continuous valence score for unseen words from a set of manually labeled seed words and semantic similarity ratings. The proposed affect estimator is trained using LMS and feature selection. Once trained, the affect estimator was used to compute the valence ratings for unseen words in a fully automatic unsupervised manner that did not require any external linguistic resource (e.g., ontologies), using semantic similarity scores computed from web documents. The lexicon creation algorithm achieved very good results on the ANEW and General Inquirer datasets, achieving higher performance than any method in literature. All of the modifications to the original method by Turney proved useful, even if only in specific contexts. The method can be adapted to work well using any data source and any similarity metric we tried, allowing easy adaptation to any constraints on queries, space and computational complexity as well as any starting lexicon. One obvious use of such a method would be to create affective lexica for languages other than English, for which resources like WordNet do not exist. In addition, preliminary results on sentence level valence estimation show that simple linear and non-linear fusion schemes achieve performance that is at least at a par with the state-of-the-art for the SemEval task.

Although the results are encouraging, this work represents only the first step towards applying machine learning methods to the problem of analysing the affective content of words and sentences. Alternative semantic similarity or relatedness metrics could also be investigated, better fusion models that incorporate syntactic and pragmatic information can also prove instrumental in achieving improved sentence-level valence scores. Overall, the proposed method creates very accurate ratings and is a good starting point for future research. An important advantage of the proposed method is it's simplicity: the only requirements are a few hundred labeled seed words and a web search engine.

Chapter 4

Conclusions

In Chapter 2 we an annotated database of affect and our experiments in tracking the affective contents of the movies using HMMs. Overall the results are promising, even surprisingly good in the case of predicting arousal. Compared to previous work on much more limited domains, we experienced a, perhaps predictable, regression in feature complexity: we found that very generic descriptors achieved the best results, while some very popular features like motion and tempo failed to provide an improvement.

Feature-level fusion proved inadequate for subtitle information, leading to a more in depth exploration of affective text modeling, as shown in Chapter 3. The proposed affective lexicon creation algorithm performed well in virtually any task we tried and is extremely versatile with regards to the nature of the starting lexicon, the type of similarity metrics used and the seed word selection strategy. Also unlike contemporary methods based on WordNet, this method can create ratings for (proper) nouns and potentially be applied to languages other than English. Sentence experiments were conducted simply to verify the applicability of the lexicon to sentence tasks and again achieved state-of-the-art results. While the method is generally successful, it still fails in the subtitle classification task.

Despite the failure to merge the two parts of this work into a single movie-oriented solution, we feel that results achieved overall are good enough and interesting enough to consider this research endeavour a success.

4.1 Future Work

Eventually the audio-visual and text parts of this work will have to be merged, but before that there are potential improvements to be applied to both.

The audio-visual model performs acceptably well, but is fairly light on the video features. The use of many times more audio than video features leads to a system that is inherently biased and that may very well fail when the cinematographic style is less dependent on music. So a more in depth look at the visual parts is in order. Then there is the audio-visual fusion scheme, which will have to be improved; this is not a simple task and constitutes a field of research by itself. There will also have to be more work on incorporating high-level semantics into the model, as they are obviously important modifiers of affect. One idea that seems feasible using today's technology, is using face and speaker recognition to identify movie characters.

On the affective text front, there doesn't seem to be much room for improvement with regards to the lexicon creation method, apart perhaps from using more sophisticated similarity metrics as they become available. There are possible improvements when the method is used to create a lexicon for a sentence classification task. So far we have not considered part of speech or senses when creating word ratings, however these can potentially be incorporated by the use of specialized web queries when collecting data. Our sentence model is very naive disregarding any syntactic information, so that is an obvious step forward in that regards. Applying this same model to movies will require much deeper exploration of the interactions between sentences, as well as, taking into account *temporal* distance, a factor that has never been inspected before in this context. Finally the two streams will have to be merged, which again points to the multimodal fusion problem.

Appendix A

Annotation - Analysis

This section shows some of the results obtained during the analysis of users' annotations. It should be noted that almost none of the results that follow shows statistically significant differences.

A.1 Values analysis

Figure A.1 shows the comparison, by ANOVA analysis, of users' mean arousal, valence and absolute valence depending on whether they enjoyed the clip. There is no significant dependency, though the overall trend is as expected.

A.2 Time delay

Figure A.6 shows an example of two annotations performed for the same clip, by the same user. The user had not watched the movie before the annotation and it is clear that he shows a significant delay in reacting to the movie during the first annotation.

To assess and compare time delay we use the cross-correlation between user annotations and the expected emotion (the average, after removal of outliers). Then we use the questionnaire answers as grouping factors and perform ANOVA analysis. Figures A.7, A.8 and A.9 show relative time delays when using prior knowledge, annotator name and enjoyment of the clip as grouping factors. Prior knowledge only has an appreciable effect on valence: having prior knowledge means a noticeably faster response time, suggesting that the self-evaluation of valence is a more complex cognitive process than the self-evaluation of arousal. This is consistent with fMRI research showing that

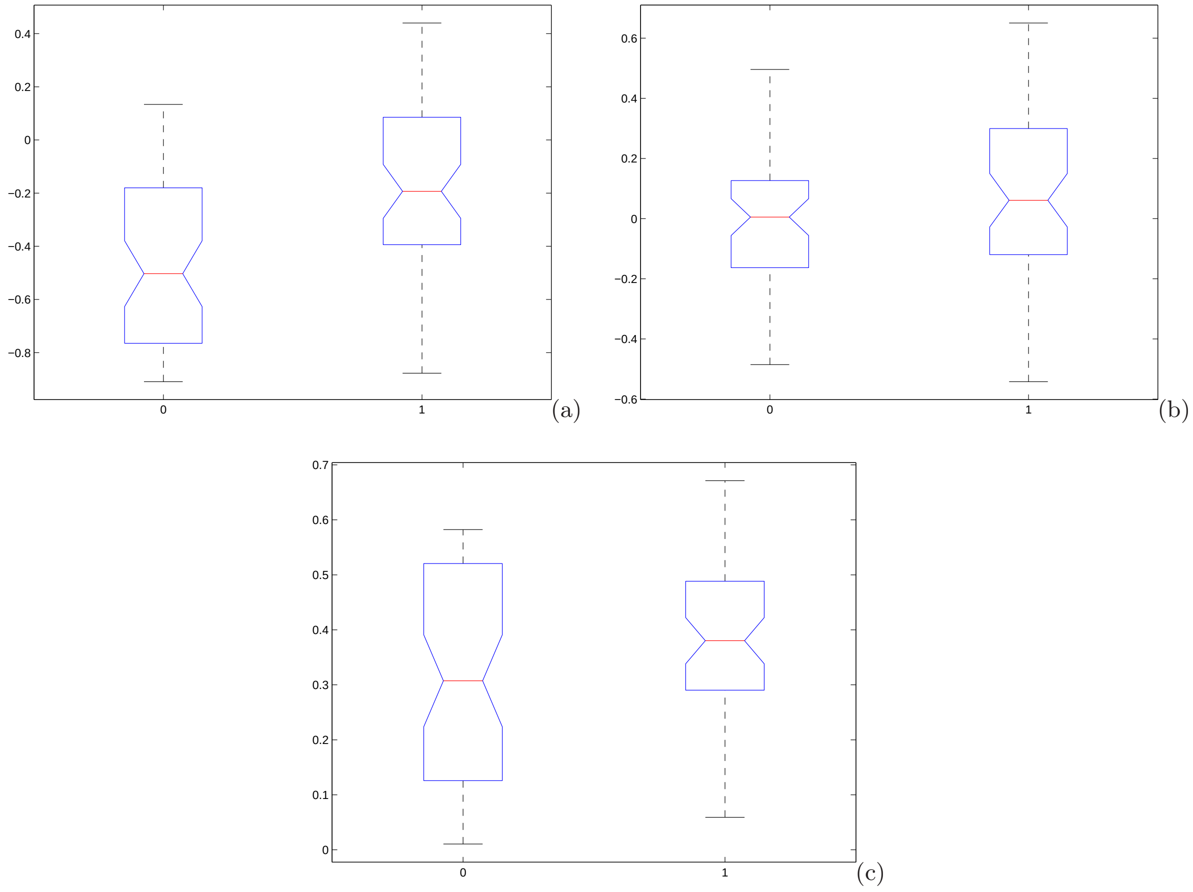


Figure A.1: ANOVA graphs for the normalized mean arousal (a), valence (b) and absolute valence (c) given the annotator liked(1) or disliked(0) the clip

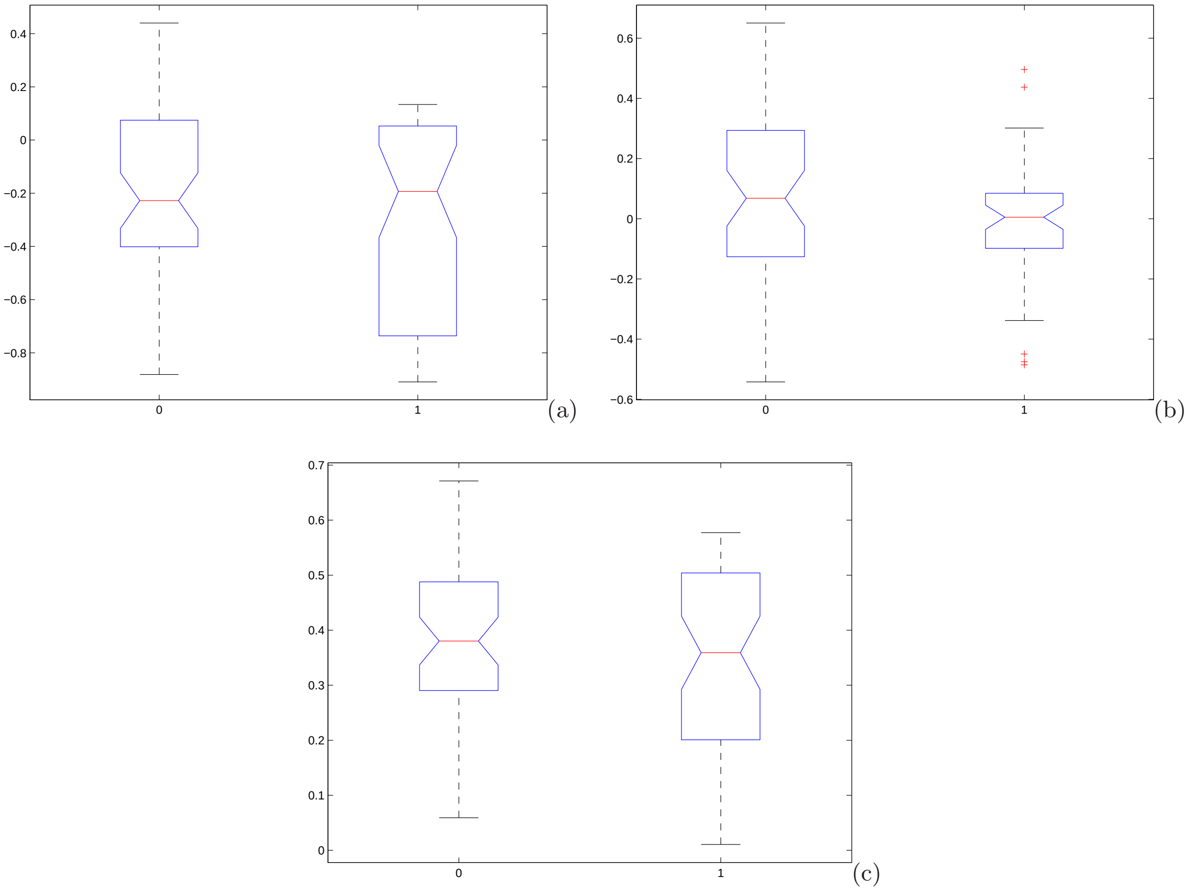


Figure A.2: ANOVA graphs for the normalized mean arousal (a), valence (b) and absolute valence (c) valence given the annotator was bored (1) or not (0) of the clip

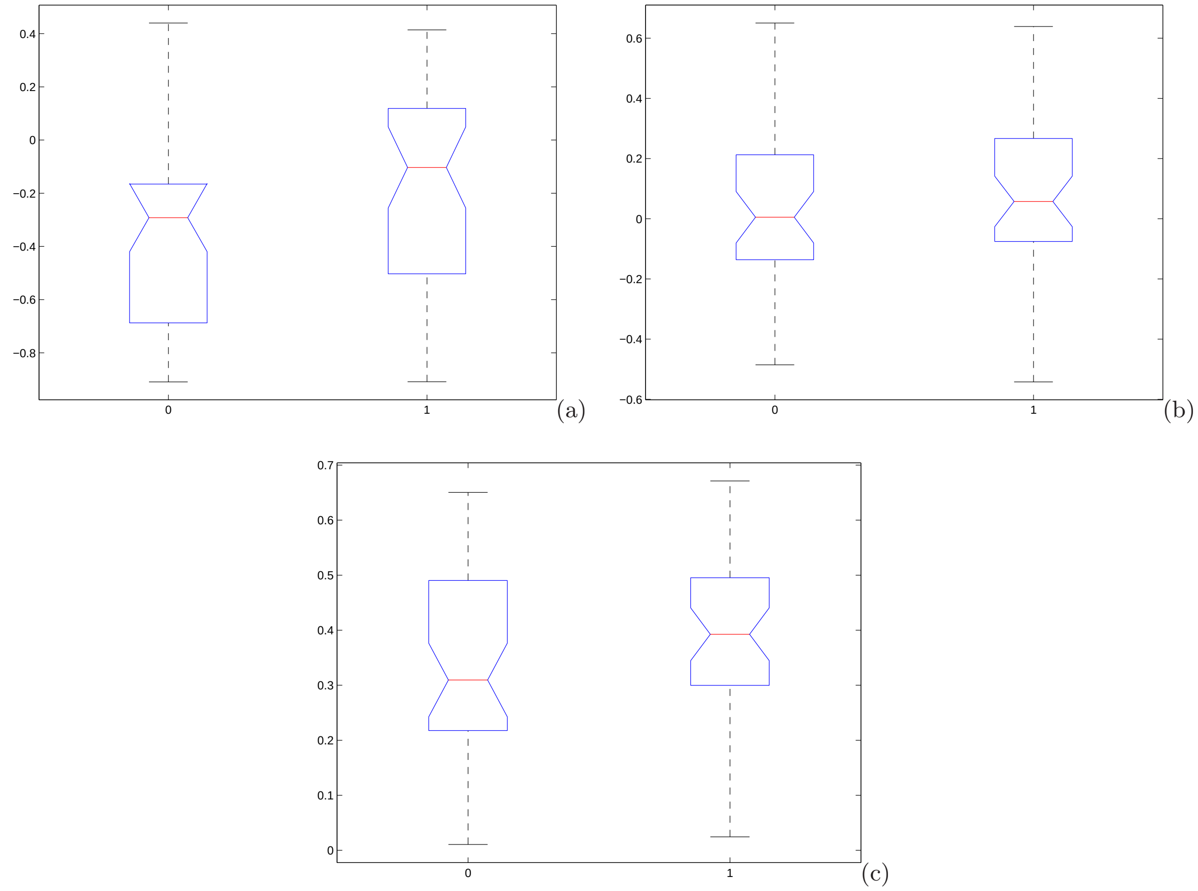


Figure A.3: ANOVA graphs for the normalized mean arousal (a), valence (b) and absolute valence (c) given the annotator had watched the same movie before (1) or not (0)

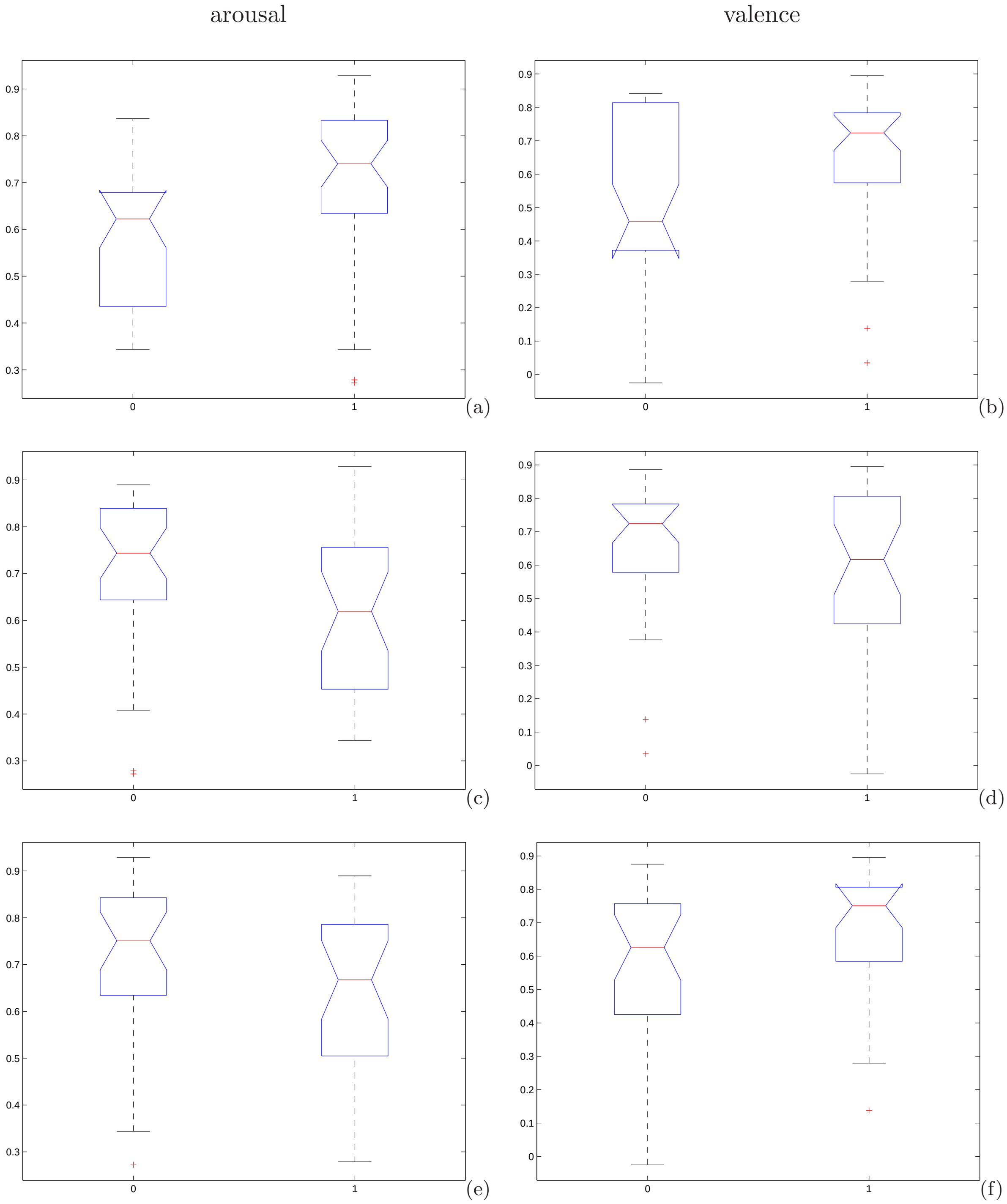


Figure A.4: ANOVA graphs for the correlation between each user's first and second annotations. Grouping factors: liked(1) or disliked(0) the clip (a)-(b), was bored (1) or not (0) of the clip (c)-(d), had watched the same movie before (1) or not (0) (e)-(f)

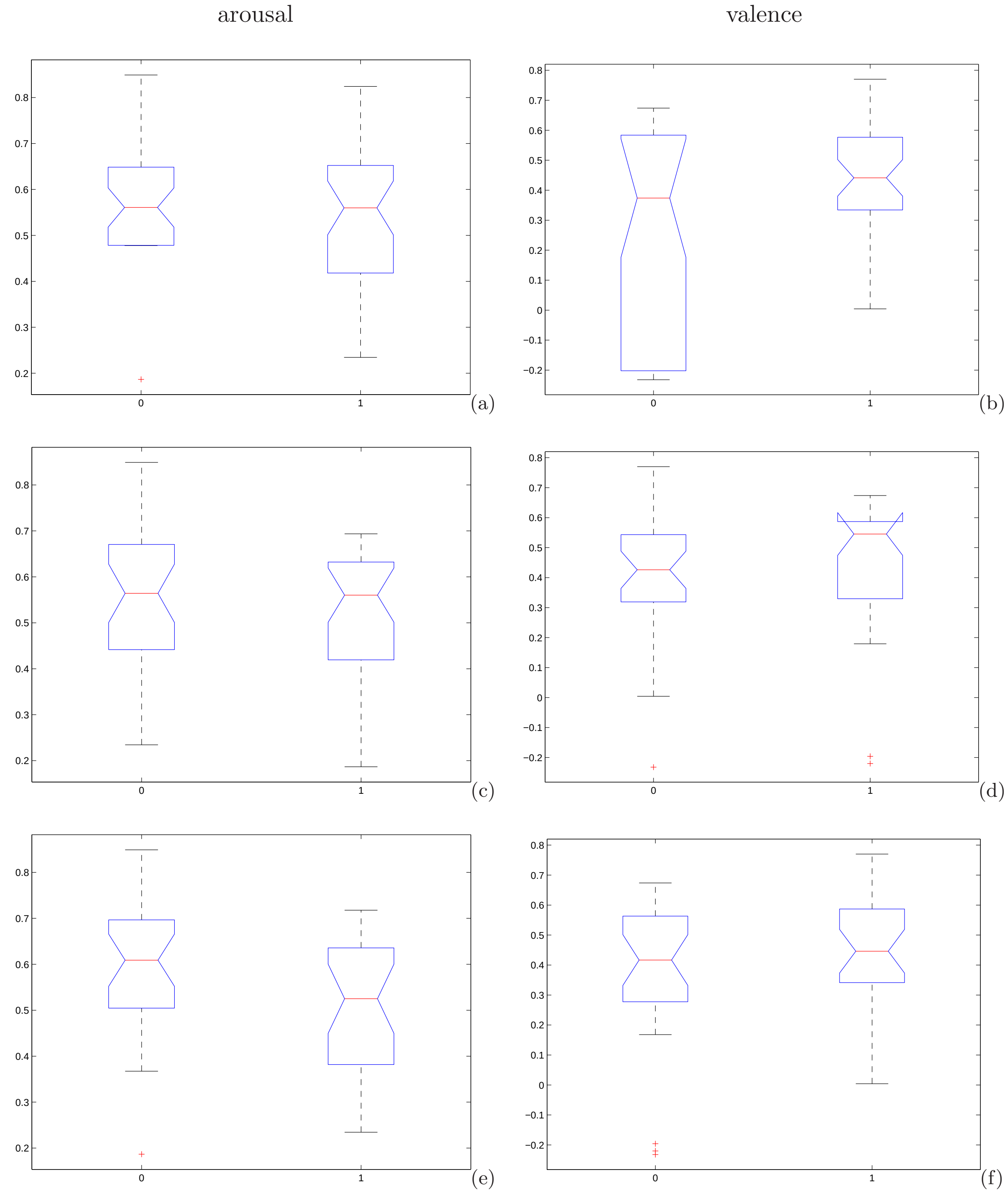


Figure A.5: ANOVA graphs for the correlation between each user's annotation and the intended emotion annotation. Grouping factors: liked(1) or disliked(0) the clip (a)-(b), was bored (1) or not (0) of the clip (c)-(d), had watched the same movie before (1) or not (0) (e)-(f)

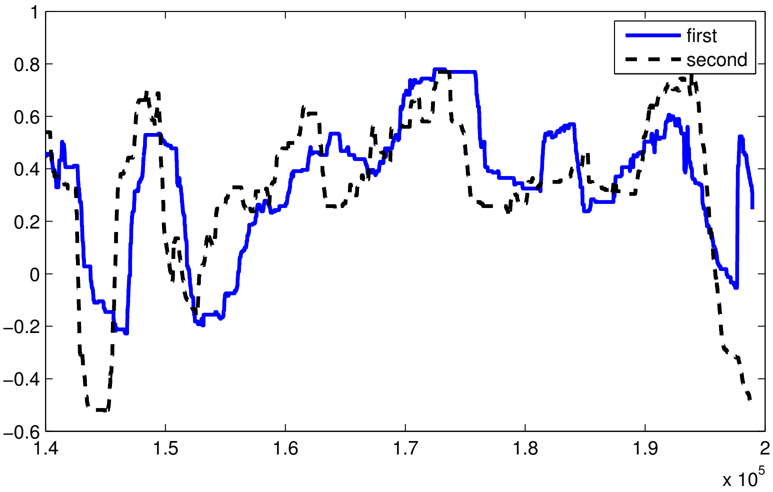


Figure A.6: Excerpt from a user's two arousal annotations for Million Dollar Baby, showing a clear time-shift of around 10 seconds

self-report of valence is more dependent on knowledge-based interpretation, making knowledge itself more important. Looking at the comparison per annotator, the expert of the group (nikos) who had watched all clips multiple times before annotating is just average when it comes to arousal, but very fast when it comes to valence, matching the previous find. Enjoyment of the clip on the other hand has an effect on arousal, with users that enjoyed a clip having a *slower* response time. Perhaps an explanation is the varied degree of immersion implied by enjoyability, however that would suggest that annotators enjoying the movie neglect their annotation, something not too flattering.

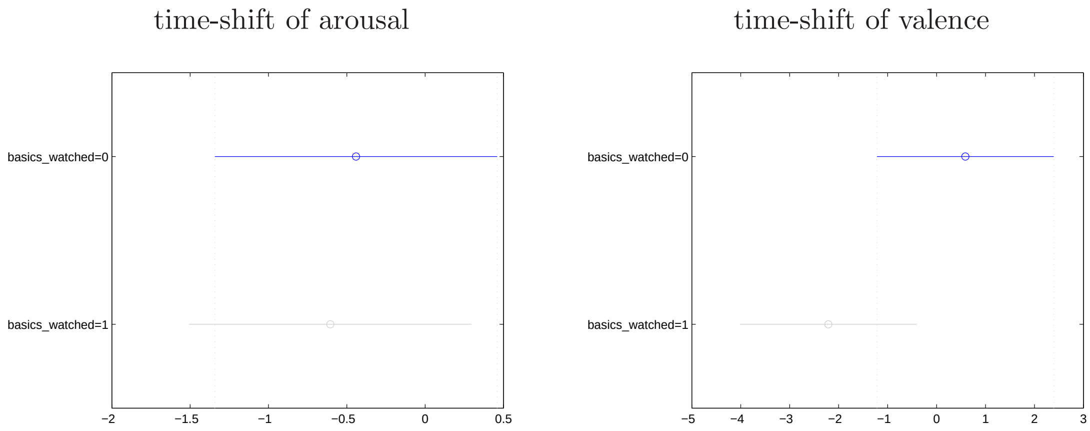


Figure A.7: Anova graphs for the arousal and valence time-shifts, grouping factor: have watched the movie and have some recollection.

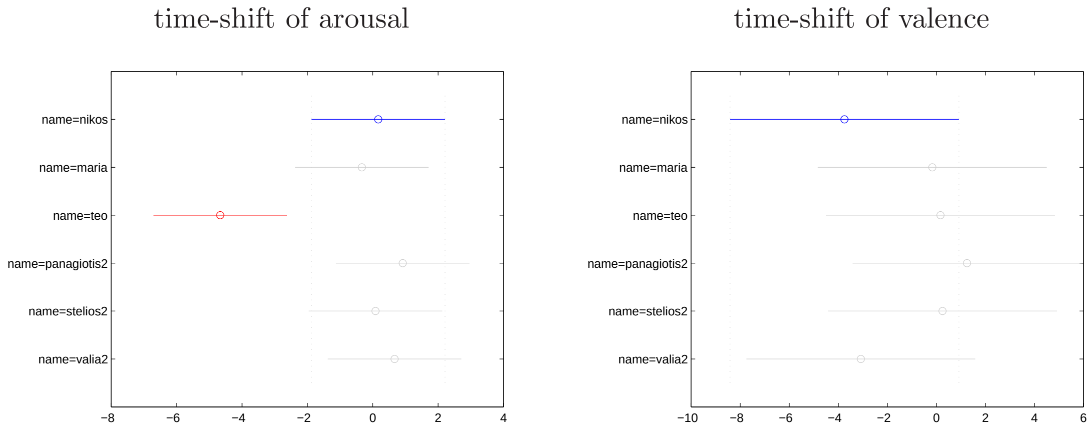


Figure A.8: Anova graphs for the arousal and valence time-shifts, grouping factor: annotator. Note that the "expert" of the group is average when it comes to arousal, but very fast when it comes to valence

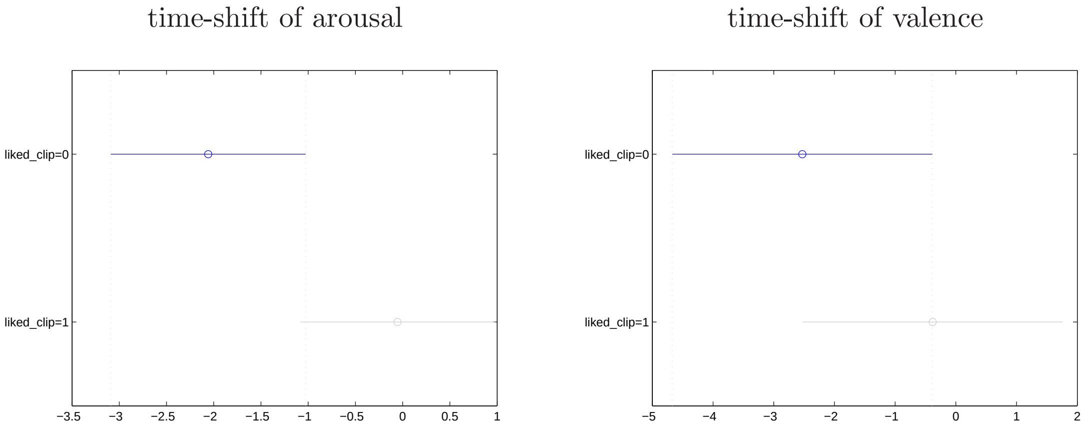


Figure A.9: Anova graphs for the arousal and valence time-shifts, grouping factor: enjoyed the clip. For some reason enjoyment slows the reaction time?

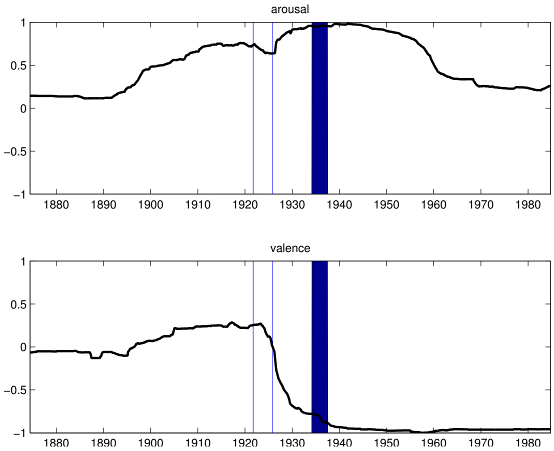
Appendix B

Events VS Affect

We performed an annotation of events on two clips in order to examine how the manifestation of expectation is presented on our affective curves. Our initial thoughts on the subject suggested that defining the temporal limits of an event was not a trivial matter. As such we annotated hierarchically; starting from minimal events then annotating larger events containing these minimals. An example can be seen in the Figure B.1, taken from Million Dollar baby; we start by annotating the minimal event "hero breaks her neck" then expand backwards to create larger events. Nominally we should also expand forward in time, creating onset-event-fade arcs, however we are currently interested in prediction which should manifest immediately prior to the event.

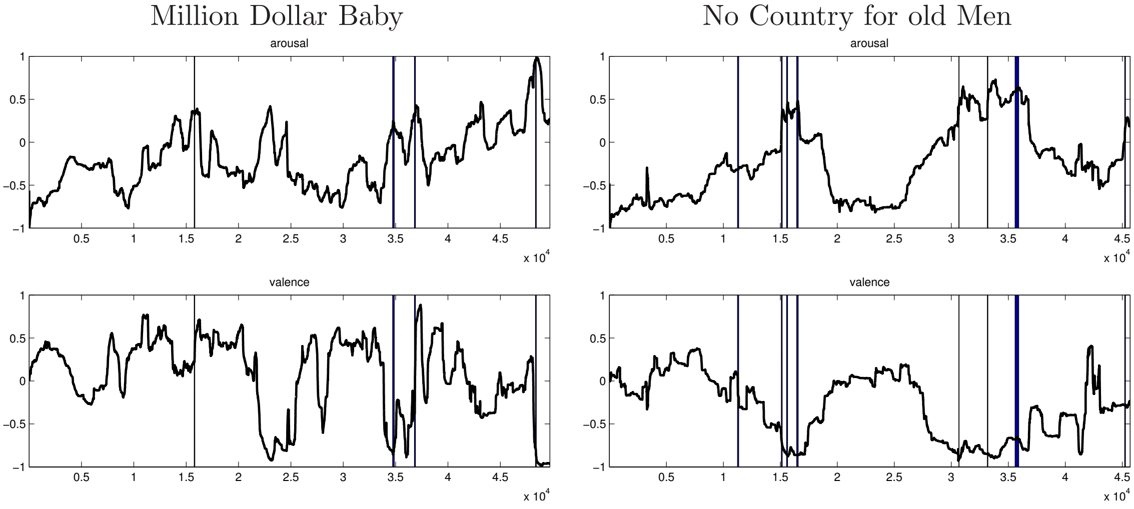
Figure B.2 shows the positions of all events we annotated on both clips. Figures B.3 and B.4 show a more detailed view of some of them.

To examine the response to these events we compared (visually) how annotators responded to them during their first and second annotations. A couple of examples can be seen in Figures B.5 and B.6. These particular examples show that the second annotations do shift so as to better align events with major changes in affective state.



The shaded area represents the impact, the first vertical line shows the beginning of the sequence (antagonist glaring at the hero), the second vertical line shows the point when image/sound go to slow-motion.

Figure B.1: Milliond Dollar Baby EV4: Hero breaks her neck.



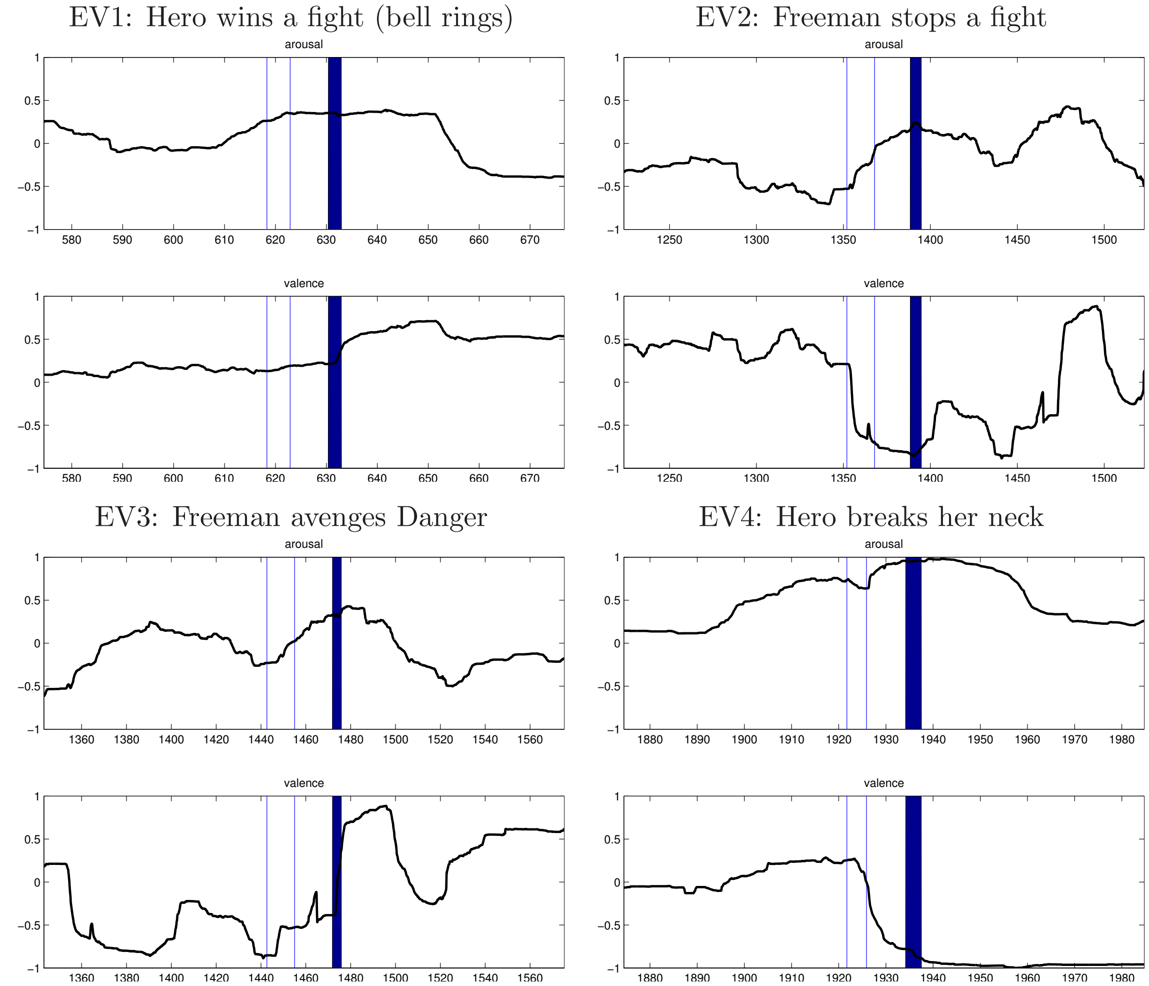
Million Dollar Baby

No Country for old Men

EVENTS
Hero wins a fight (bell rings)
Freeman stops a fight
Freeman avenges Danger
Hero breaks her neck

EVENTS
Killer checks room (twice)
Killer shoots a man
Killer shoots a man
Killer shoots a man
Hero shoots at door
Truck driver gets shot
Hero shoots, corners the killer
Killer blows up a car

Figure B.2: Graphs and tables showing our annotated events for each clip



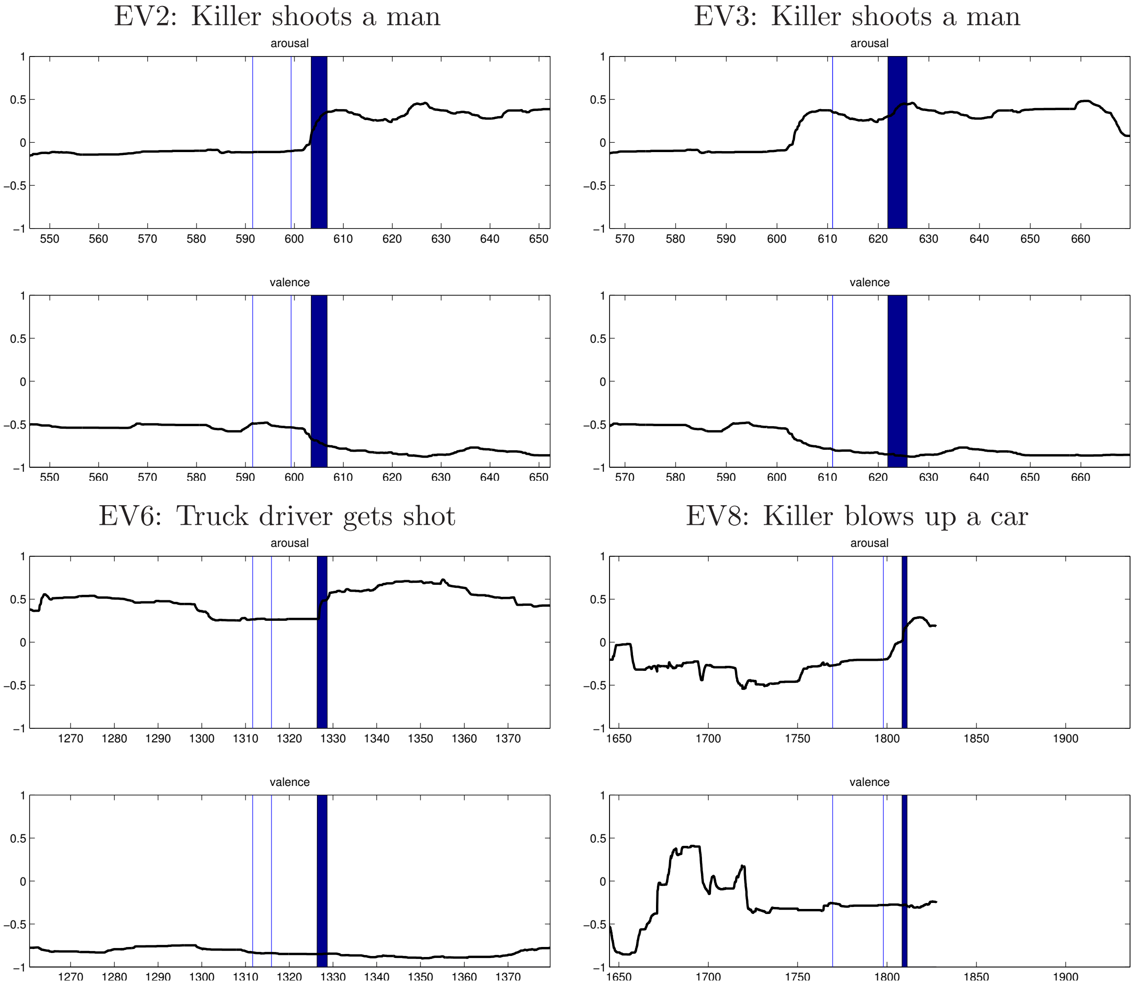
EV1: Hero wins a fight (bell rings)

EV2: Freeman stops a fight

EV3: Freeman avenges Danger

EV4: Hero breaks her neck

Figure B.3: Events from Million Dollar Baby



EV2: Killer shoots a man

EV3: Killer shoots a man

EV6: Truck driver gets shot

EV8: Killer blows up a car

Figure B.4: Events from No Country for Old Men

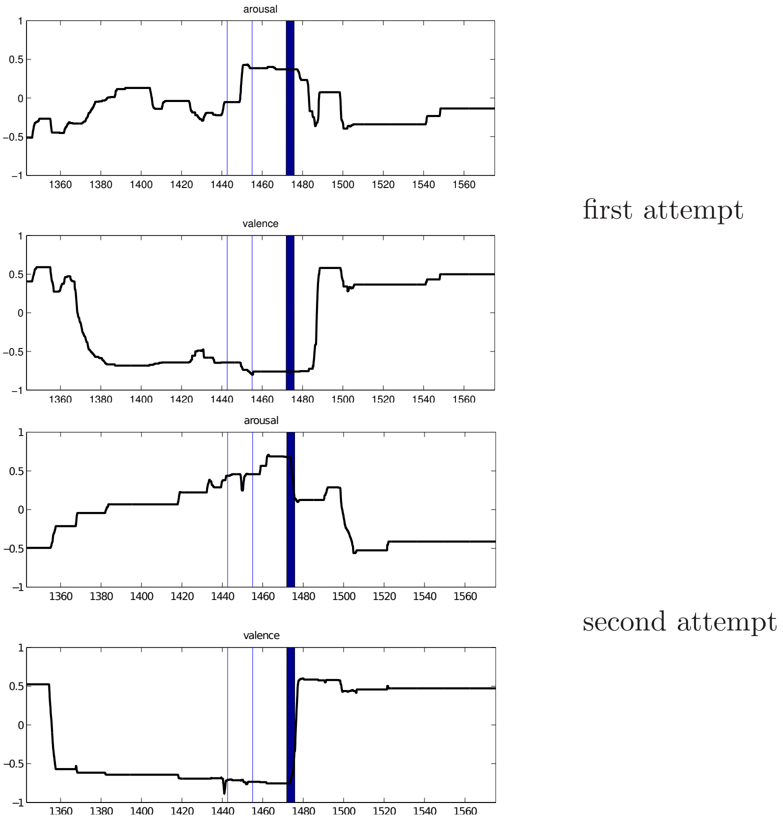
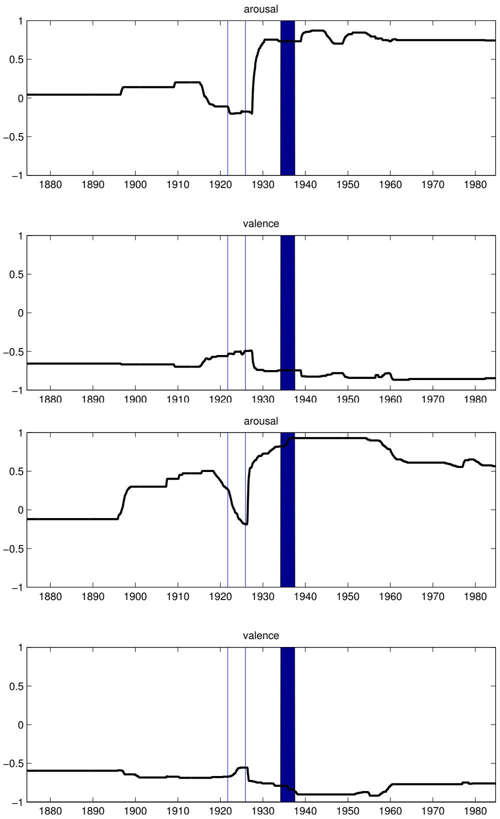


Figure B.5: MDB Event 3: Freeman avenges Danger.



first attempt

second attempt

Figure B.6: MDB Event 4: Hero breaks her neck.

Appendix C

Multimedia Results

This appendix contains some extra results and information for the multimedia experiments detailed in Chapter 2. Figure C.1 shows a more exhaustive list of audio features used in our experiments, organized by source.

We performed a sort of forward selection using a wrapper approach, however we could not practically evaluate every combination (exhaustive search) or even perform best-first selection by incrementing by a single feature at a time. Instead we evaluated in small groups. Some of the results can be seen in Table C.1. This first round of selection gave us MFCCs as the best performing group of features, so we used them as a base and ran further experiments attempting to improve on their performance by adding features to a set already containing MFCCs and their deltas and accelerations. Some results from this round can be seen in Table C.2. The final feature sets used for our experiments are shown in Table 2.4.

- HTK
  - MFCC_0_E_D_A_Z (42)
  - PLP_0_E_D_A_Z (42)
- MARSYAS
  - Chroma (14)
  - Spectral Crest Factor (19)
  - Spectral Flatness Measure (24)
  - Spectral Centroid, Flux and Rolloff (3)
  - Pitch, raw and smoothed (5)
- OpenSMILE
  - log Mel Frequency Band Energy (8)
  - Pitch, Jitter, Jitter of Jitter, Shimmer (4)
  - Line Spectral Pairs (8)
  - pitch(mean,range,variance,slope), intensity(mean,range) (6)
- MIR Toolbox
  - Tempo, Pulse Clarity, Event Density, Spectral Skewness, Spectral kurtosis, Spectral Flatness, Rhythm Irregularity, Inharmonicity (8)
  - Standard deviation of RMS, Maximum value of summarized fluctuation, Key clarity average, Mode average, Averaged spectral novelty (5)
  - Average of RMS, Maximum value of summarized fluctuation, Spectral Centroid average, Spectral spread average, Entropy of smoothed and collapsed spectrogram (5)
- NTUA
  - maximum average Teager energy, mean instant amplitude, mean instant frequency

Figure C.1: Full list of evaluated audio features, grouped by source.

Table C.1: Comparison of different feature groups on the performance of arousal and valence prediction

			arousal		
Features	ACC	ACC±1	D.CORR	MSQE	C.CORR
MFCC	0.22	0.53	0.34	0.20	0.46
Chroma	0.21	0.50	0.37	0.22	0.47
Spectral Crest Factor	0.16	0.44	0.11	0.25	0.20
Spectral Flatness Measure	0.17	0.43	0.11	0.27	0.20
Spectral Centroid, Flux, Rolloff	0.20	0.46	0.22	0.31	0.30
MIR features	0.20	0.49	0.30	0.23	0.43
ntua audio	0.20	0.51	0.29	0.22	0.40
PLP	0.22	0.52	0.34	0.21	0.47
log Mel Freq. Band Energy	0.19	0.47	0.26	0.23	0.35
Line Spectral Pairs	0.17	0.45	0.13	0.27	0.20
Kotropoulos	0.20	0.51	0.33	0.20	0.44
Pitch, jitter, jitter of jitter, shimmer	0.14	0.40	0.01	0.32	-0.02

			valence		
Features	ACC	ACC±1	D.CORR	MSQE	C.CORR
MFCC	0.20	0.51	0.10	0.30	0.16
Chroma	0.19	0.49	0.07	0.40	0.11
Spectral Crest Factor	0.17	0.47	0.03	0.32	0.06
Spectral Flatness Measure	0.17	0.47	0.07	0.31	0.11
Spectral Centroid, Flux and Rolloff	0.18	0.50	0.06	0.33	0.09
MIR features	0.18	0.48	0.05	0.34	0.07
ntua audio	0.19	0.51	0.05	0.33	0.05
PLP	0.21	0.51	0.11	0.30	0.16
log Mel Frequency Band Energy	0.18	0.49	0.06	0.35	0.09
Line Spectral Pairs	0.17	0.47	0.05	0.33	0.07
Kotropoulos	0.17	0.44	-0.04	0.44	-0.06
Pitch, Jitter, Jitter of Jitter, Shimmer	0.20	0.50	0.00	0.32	0.01

Table C.2: Comparison of different feature groups on the performance of arousal and valence prediction. Each group is added to a feature set already containing MFCCs.

MFCC plus features on Arousal					
MFCC and:	ACC	ACC±1	D.CORR	MSQE	C.CORR
Chroma	0.24	0.54	0.38	0.23	0.51
Spectral Crest Factor	0.19	0.46	0.23	0.22	0.38
Spectral Flatness Measure	0.19	0.48	0.20	0.21	0.33
Spectral Centroid, Flux and Rolloff	0.23	0.53	0.32	0.21	0.45
ntua audio	0.21	0.51	0.31	0.21	0.43

MFCC plus features on Valence					
MFCC and:	ACC	ACC±1	D.CORR	MSQE	C.CORR
Chroma	0.16	0.43	0.04	0.38	0.04
Spectral Crest Factor	0.18	0.47	0.07	0.31	0.14
Spectral Flatness Measure	0.18	0.48	0.09	0.30	0.16
Spectral Centroid, Flux and Rolloff	0.20	0.51	0.11	0.29	0.18
ntua audio	0.18	0.49	0.04	0.32	0.07

Appendix D

Similarity VS co-occurence

In Chapter 3 we used multiple hit-based similarity metrics with varying success. We also found that rescaling some of the less well performing metrics via simple functions could improve performance significantly. We believe that is due to the differences between metrics regarding how they scale relatively to the co-occurrence hit count.

Plotting relative frequencies and their resultant similarity metrics is not possible, since it is a multivariate problem. As an alternative we show multiple plots, where each plot has fixed IND counts and the only variable is the co-occurence count. What changes between the different plots is the ratio of IND counts: they may be equal (1 to 1) or one is a multiple of the other. In all graphs the x axis represents the value of the co-occurence count as a percentage of the minimum of IND counts. The y axis values have no meaning, since for representation purposes the similarity metrics are peak-to-peak normalized. Figure D.1 shows the results.

As we expected, these graphs explain why some metrics are less suitable to the task. The metrics are clearly not connected by a linear relation, since they scale very differently to the increasing co-occurrence hit count, but we use them as part of a linear equation, assuming they are linear to an ideal “affective similarity”. We can assume that the scaling of this affective similarity would be most similar to those of PMI and Google semantic relatedness, our best performing metrics. Dice and Jaccard coefficients (which generally perform poorly) deviate significantly from that, but they can be adapted by using a scaling function less steep than linear, as shown in the graphs for the logarithm of Jaccard coefficient.

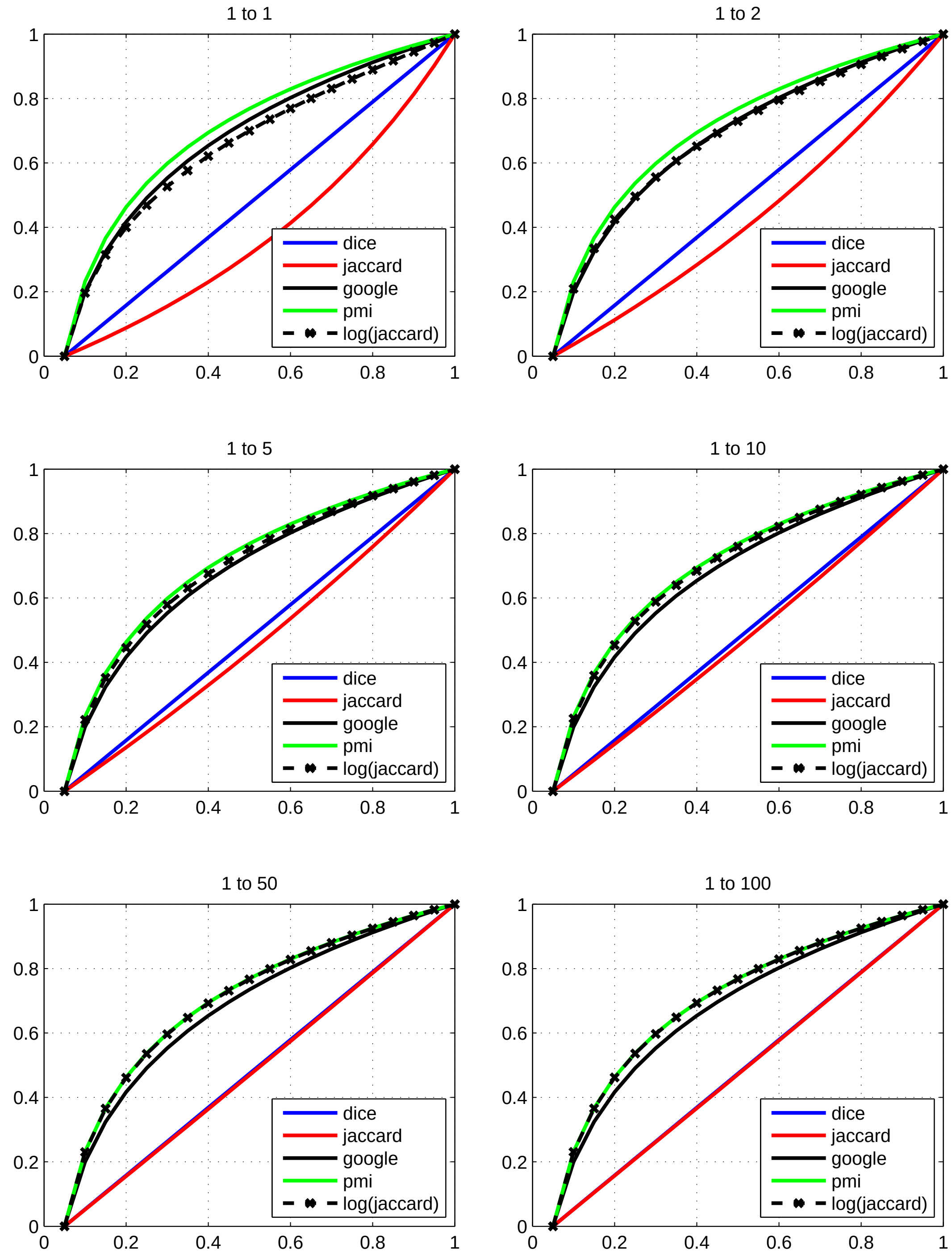


Figure D.1: Mapping of co-occurence count to similarity

Appendix E

ChIMP results

The ChIMP database was used to evaluate the method on spontaneous spoken dialog interaction. The ChIMP corpus contains 15,585 manually annotated spoken utterances, with each utterance labeled with one of three emotional state tags: neutral, polite, and frustrated [62]. While the labels reflect emotional states, their valence rating is not obvious. In order to adapt the affective model to the ChIMP task, the discrete sentence level valence scores were mapped as follows: frustrated was assigned a valence value of -1, neutral was 0 and polite was 1. To bootstrap the valence scores for each word in the ChIMP corpus, we used the average sentence-level scores for all sentence where that word appeared. Finally, the ANEW equation system matrix was augmented with all the words in the ChIMP corpus and the valence model in (3.3) was estimated using LMS. Note that for this training process a 10-fold cross validation experiment was run on the ChIMP corpus sentences. The relative *weight* of the ChIMP corpus adaptation data was varied by adding the respective lines multiple times to the augmented system matrix, e.g., adding each line twice gives a weight of $w = 2$. We tested weights of $w = 1$, $w = 2$, and using only the samples from ChIMP as training samples (denoted as $w = \infty$). The valence boundary between frustrated and other classes was selected based on the a-priori probability distribution for each class, and is simply the Bayesian decision boundary (similarly between polite and other classes). In Table E.1, the two-class sentence-level classification accuracy is shown for the ChIMP corpus (polite vs other: "P vs O", frustrated vs other: "F vs O"). For the baseline ChIMP experiment, 200 words from the ANEW corpus were used to train the affective model in (1) using the linear similarity function. For the adaptation experiments, the parameter w denotes

Table E.1: Sentence classification accuracy for the ChIMP baseline and ChIMP adapted tasks.

Sentence Classification Accuracy			
	avg	w.avg	max
ChIMP (P vs O) baseline	**0.70**	0.69	0.54
ChIMP (P vs O) adapt $w = 1$	**0.74**	0.70	0.67
ChIMP (P vs O) adapt $w = 2$	**0.77**	0.74	0.71
ChIMP (P vs O) adapt $w = \infty$	**0.84**	0.82	0.75
ChIMP (F vs O) baseline	0.53	0.62	**0.66**
ChIMP (F vs O) adapt $w = 1$	0.51	**0.58**	0.57
ChIMP (F vs O) adapt $w = 2$	0.49	**0.53**	0.53
ChIMP (F vs O) adapt $w = \infty$	0.52	0.52	0.52

the weighting given to the in-domain ChIMP data, i.e., number of times the adaptation equation were repeated in the system matrix (2). Results are shown for the three fusion methods (average, weighted average, maximum).

Paper [61] achieves 81% accuracy in politeness detection and 61.7% accuracy in frustration detection. With regards to politeness detection, performance of the baseline (unsupervised) model is lower than that quoted in [61] for lexical features. Performance improves significantly by adapting the affective model using in-domain ChIMP data reaching up to 84% accuracy for linear fusion and surpassing [61]. The best results for frustration detection task are achieved with the baseline model and max fusion schemes at 66% (again better than [61]). It is interesting to note that in-domain adaptation does not improve frustration classification. A possible explanation is that there is a high lexical variability when expressing frustration, thus, the limited adaptation data does not help much. Also frustration may be expressed with a single word that has very negative valence, as a result, max fusion works best here. Overall, very good results are achieved using a domain-independent affective model to classify politeness and frustration. However, the appropriate adaptation and sentence-level fusion schemes seem to be very much task-dependent.

Bibliography

[1] A. Andreevskaia and S. Bergler, "Semantic tag extraction from WordNet glosses," in *Proc. LREC*, pp. 413–416, 2006. 26

[2] A. Andreevskaia and S. Bergler, "CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging," in *Proc. SemEval*, pp. 117–120, 2007. 27

[3] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke, "Prosody-based automatic detection of annoyance and frustration in human-computer dialog," in *Proc. ICSLP*, pp. 2037–2040, 2002. 23

[4] A. Austin, E. Moore, U. Gupta, and P. Chordia, "Characterization of movie genre based on music score," in *Proc. ICASSP*, pp. 421–424, 2010. 6

[5] K. Balog, G. Mishne, and M. de Rijke, "Why are they excited? identifying and explaining spikes in blog mood levels," in *Proc. EACL*, pp. 207–210, 2006. 23

[6] J. Beskow, L. Cerrato, B. Granstrm, D. House, M. Nordenberg, M. Nordstrand, and G. Svanfeldt, "Expressive animated agents for affective dialogue systems.," in *ADS* (E. Andr, L. Dybkjr, W. Minker, and P. Heisterkamp, eds.), vol. 3068 of *Lecture Notes in Computer Science*, pp. 240–243, Springer, 2004. 23

[7] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. of International Conference on World Wide Web*, pp. 757–766, 2007. 27, 30

[8] M. Bradley and P. Lang, "Affective norms for english words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1.," The Center for Research in Psychophysiology, University of Florida, 1999. 24

[9] A. Budanitsky and G. Hirst, “Evaluating WordNet-based measures of semantic distance,” *Computational Linguistics*, vol. 32, pp. 13–47, 2006. 27

[10] J. Carrillo de Albornoz, L. Plaza, and P. Gervs, “A hybrid approach to emotional sentence polarity and intensity classification,” in *Proc. CoNLL*, pp. 153–161, 2010. 61

[11] F.-R. Chaumartin, “UPAR7: A knowledge-based system for headline sentiment tagging,” in *Proc. SemEval*, pp. 422–425, 2007. 27

[12] R. L. Cilibrasi and P. M. Vitnyi, “The Google similarity distance,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2007. 30

[13] R. Cowie, E. Douglas-Cowie, S. Savvidou, E. McMahon, M. Sawey, and M. Schröder, “’FEELTRACE’: An instrument for recording perceived emotion in real time,” in *ISCA Workshop on Speech & Emotion*, pp. 19–24, 2000. 8

[14] R. Dietz and A. Lang, “Affective agents: Effects of agent affect on arousal, attention, liking and learning,” in *Proc. Cognitive Technology Conference*, 1999. 6, 12

[15] A. Esuli and F. Sebastiani, “Determining the semantic orientation of terms through gloss classification,” in *Proc. CIKM*, pp. 617–624, 2005. 26

[16] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *Proc. LREC*, pp. 417–422, 2006. 24

[17] G. Evangelopoulos, A. Zlatintsi, G. Skoumas, K. Rapantzikos, A. Potamianos, P. Maragos, and Y. Avrithis, “Video event detection and summarization using audio, visual and text saliency,” in *Proc. ICASSP*, pp. 3553–3556, 2009. 7

[18] F. Eyben, M. Wollmer, and B. Schuller, “Openear – introducing the munich open-source emotion and affect recognition toolkit,” in *Proc. ACII 2009*, pp. 1–6, 2009. 17

[19] J. Gracia, R. Trillo, M. Espinoza, and E. Mena, “Querying the web: A multi-ontology disambiguation method,” in *Proc. of International Conference on Web Engineering*, pp. 241–248, 2006. 30

[20] M. Grimm and K. Kroschel, "Evaluation of natural emotions using self assessment manikins," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 381–385, 2005. 8

[21] A. Hanjalic, "Extracting moods from pictures and sounds," *IEEE Signal Processing Magazine*, pp. 90–100, Mar. 2006. 6, 7, 12

[22] A. Hassan and D. Radev, "Identifying text polarity using random walks," in *Proc. ACL*, pp. 395–403, 2010. 26, 59

[23] V. Hatzivassiloglou and K. McKeown, "Predicting the Semantic Orientation of Adjectives," in *Proc. ACL*, pp. 174–181, 1997. 24

[24] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. SIGKDD*, KDD '04, pp. 168–177, ACM, 2004. 23

[25] E. Iosif and A. Potamianos, "Unsupervised Semantic Similarity Computation Between Terms Using Web Documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1637–1647, 2009. 27, 29

[26] E. Iosif, A. Tegos, A. Pangos, E. Fosler-Lussier, and A. Potamianos, "Combining statistical similarity measures for automatic induction of semantic classes," in *Proc. IEEE/ACL Workshop Spoken Language Technology*, pp. 86–89, 2006. 30, 31

[27] H. Kang, "Affective content detection using HMMs," in *Proc. ACM Multimedia*, pp. 259–262, 2003. 6, 13

[28] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997. 35

[29] O. Lartillot, P. Toiviainen, and T. Eerola, "A matlab toolbox for music information retrieval," in *Data Analysis, Machine Learning and Applications*, pp. 261–268, Springer Berlin Heidelberg, 2008. 17

[30] C. M. Lee and S. S. Narayanan, "Toward detecting emotions in spoken dialogs.," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 293–303, 2005. 23

[31] C. M. Lee, S. S. Narayanan, and R. Pieraccini, "Combining acoustic and language information for emotion," in *Proc. ICSLP*, pp. 873–876, 2002. 23

[32] L. Lloyd, D. Kechagias, and S. Skiena, "Lydia: A system for large-scale news analysis," in *Proc. SPIRE*, no. 3772 in Lecture Notes in Computer Science, pp. 161–166, 2005. 23

[33] N. Malandrakis, A. Potamianos, G. Evangelopoulos, and A. Zlatintsi, "A supervised approach to movie emotion tracking," in *Proc. ICASSP*, pp. 2376–2379, 2011. 5

[34] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Emotiword: Affective lexicon creation with application to interaction and multimedia data," in *Proc. MUSCLE International Workshop on Computational Intelligence for Multimedia Understanding*, 2011. 23

[35] N. Malandrakis, A. Potamianos, E. Iosif, and S. Narayanan, "Kernel models for affective lexicon creation," in *Proc. Interspeech*, pp. 2977–2980, 2011. 23

[36] K. Moilanen and S. Pulman, "Sentiment Composition," in *Proc. RANLP*, pp. 378–382, 2007. 61

[37] K. Moilanen, S. Pulman, and Y. Zhang, "Packed feelings and ordered sentiments: Sentiment parsing with quasi-compositional polarity sequencing and compression," in *Proc. WASSA Workshop at ECAI*, pp. 36–43, 2010. 27, 61

[38] A. Pargellis, E. Fosler-Lussier, C.-H. Lee, A. Potamianos, and A. Tsai, "Auto-induced semantic classes," *Speech Communication*, vol. 43, pp. 183–203, 2004. 27, 30

[39] F. J. Pelletier, "The principle of semantic compositionality," *Topoi*, vol. 13, pp. 11–24, 1994. 31

[40] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing attitude and affect in text: Theory and Applications*, pp. 1–10, Springer Verlag, 2006. 27

[41] A. Purandare and D. J. Litman, "Humor: Prosody analysis and automatic recognition for f*r*i*e*n*d*s*.," in *Proc. EMNLP*, pp. 208–215, 2006. 18, 23

[42] K. Rapantzikos, N. Tsapatsoulis, Y. Avrithis, and S. Kollias, "Bottom-up spatiotemporal visual attention model for video analysis," *Image Processing, IET*, vol. 1, pp. 237 –248, June 2007. 17

[43] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627–633, 1965. 30

[44] A. Savitzky and M. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, 1964. 18

[45] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proc. International Conference on New Methods in Language Processing*, vol. 12, pp. 44–49, 1994. 37

[46] P. Stone, D. Dunphy, M. Smith, and D. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press, 1966. 24

[47] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proc. SemEval*, pp. 70–74, 2007. 23, 33, 61

[48] C. Strapparava and A. Valitutti, "WordNet-Affect: an affective extension of WordNet," in *Proc. LREC*, vol. 4, pp. 1083–1086, 2004. 24, 26

[49] C. Strapparava, A. Valitutti, and O. Stock, "The affective weight of lexicon," in *Proc. LREC*, pp. 423–426, 2006. 25

[50] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Proc. LREC*, pp. 427–432, 2006. 25, 27

[51] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 1, pp. 1–41, 2010. 61

[52] H. Takamura, T. Inui, and M. Okumura, "Extracting semantic orientations of words using spin model," in *Proc. ACL*, pp. 133–140, 2005. 26, 59

[53] P. Turney and M. L. Littman, "Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Technical report ERC-1094 (NRC 44929)," National Research Council of Canada, 2002. 24, 25, 27, 33, 37, 58, 59

[54] P. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Transactions on Information Systems*, vol. 21, pp. 315–346, 2003. 25, 27, 33, 35, 58

[55] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*, vol. 48, no. 9, pp. 1162–1181, 2006. 6

[56] P. M. Vitnyi, "Universal similarity," in *Proc. of Information Theory Workshop on Coding and Complexity*, pp. 238–243, 2005. 30

[57] H. L. Wang and L. F. Cheong, "Affective understanding in film," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 689–704, June 2006. 7

[58] J. Wiebe and R. Mihalcea, "Word sense and subjectivity," in *Proc. COLING/ACL*, pp. 1065–1072, 2006. 23

[59] M. Xu, L. Chia, and J. Jin, "Affective content analysis in comedy and horror videos by audio emotional event detection," in *Proc. ICME*, pp. 622–625, 2005. 6

[60] M. Xu, J. Jin, S. Luo, and L. Duan, "Hierarchical movie affective content analysis based on arousal and valence features," in *Proc. ACM Multimedia*, pp. 677–680, 2008. 6

[61] S. Yildirim, C. M. Lee, S. Lee, A. Potamianos, and S. Narayanan, "Detecting politeness and frustration state of a child in a conversational computer game," in *Proc. Interspeech*, pp. 2209–2212, 2005. 92

[62] S. Yildirim, S. Narayanan, and A. Potamianos, "Detecting emotional state of a child in a conversational computer game," *Computer Speech and Language*, vol. 25, pp. 29–44, January 2011. 91