

A Model of Textual Affect Sensing using Real-World Knowledge

Hugo Liu
MIT Media Laboratory
20 Ames Street #320D
Cambridge, MA 02139, USA
+1 (617) 253-5334
hugo@media.mit.edu

Henry Lieberman
MIT Media Laboratory
20 Ames Street #320G
Cambridge, MA 02139, USA
+1 (617) 253-0315
lieber@media.mit.edu

Ted Selker
MIT Media Laboratory
20 Ames Street #322
Cambridge, MA 02139, USA
+1 (617) 253-6968
selker@media.mit.edu

ABSTRACT

This paper presents a novel way for assessing the affective qualities of natural language and a scenario for its use. Previous approaches to textual affect sensing have employed keyword spotting, lexical affinity, statistical methods, and hand-crafted models. This paper demonstrates a new approach, using large-scale real-world knowledge about the inherent affective nature of everyday situations (such as “getting into a car accident”) to classify sentences into “basic” emotion categories. This commonsense approach has new robustness implications.

Open Mind Commonsense was used as a real world corpus of 400,000 facts about the everyday world. Four linguistic models are combined for robustness as a society of commonsense-based affect recognition. These models cooperate and compete to classify the affect of text.

Such a system that analyzes affective qualities sentence by sentence is of practical value when people want to evaluate the text they are writing. As such, the system is tested in an email writing application. The results suggest that the approach is robust enough to enable plausible affective text user interfaces.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *interaction styles, natural language, theory and methods, graphical user interfaces (GUI)*; I.2.7 [Artificial Intelligence]: Natural Language Processing – *language models, language parsing and understanding, text analysis*.

General Terms

Algorithms, Design, Human Factors, Languages, Theory.

Keywords

Affective computing, affective UI, emotions, story understanding, commonsense reasoning, Open Mind Commonsense.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UI '03, January 12-15, 2003, Miami, Florida, USA.

Copyright 2003 ACM 1-58113-586-6/03/0001...\$5.00.

1. INTRODUCTION

One of the great challenges facing the community of human-computer interaction today is the design of intelligent user interfaces that are more natural and social. As Nass et. al’s studies on human-human and human-computer interactions suggest, people most naturally interact with their computers in a social and affectively meaningful way, just like with other people. [11] Successful social interaction means successful affective communication. Research on synthetic agents also supports this notion with the observation that a computer’s capacity for affective interaction plays a vital role in making agents believable. [1].

Researchers like Picard have recognized the potential and importance of affect to human-computer interaction, dubbing work in this field as “affective computing.” [13]. In order for intelligent user interfaces to make use of user affect, the user’s affective state must invariably first be recognized or sensed. Researchers have tried detecting the user’s affective state in many ways, such as, *inter alia*, through facial expressions, speech, physiological phenomena, and text. (see [13] for a survey of the literature).

We believe that text is a particularly important modality for sensing affect because the bulk of computer user interfaces today are textually based. It follows that the development of robust textual affect sensing technologies can have a substantial impact in transforming today’s socially impoverished text-based UIs into socially intelligent ones. In addition, improved textual sensing can reinforce the accuracy of sensing in other modalities, like speech or facial expressions.

Previous approaches for analyzing language for affect have commonly included keyword spotting, lexical affinity, statistical methods or hand-crafted models. This paper addresses textual affect sensing using an approach that introduces multiple corpus-based linguistic analyses of affect. A real-world generic knowledge base of commonsense called Open Mind is employed. The affect of text, *at the sentence level*, is classified into one of six “basic” categories of emotion, using Ekman’s categories of happy, sad, angry, fearful, disgusted, and surprised. [3]. The approach’s robustness comes from three sources: 1) a society of multiple models that compete with each other; 2) its ability to judge the affective qualities of *the underlying semantics*; and 3) its reliance on application-independent, large-scale knowledge, covering a broad section of affective commonsense about the real world.

1.1 Paper's Focus

A related paper on this work [8] argued for the utility of evaluating affect using user-independent commonsense knowledge, and explored the user interface aspects of an application based on the approach that offered users, typing into an email browser, some affective feedback about their story. This paper focuses on the architecture, practical methods, and linguistic models behind the textual affect sensing engine. A more detailed introduction of the way the Open Mind Commonsense knowledge base (OMCS) works and is used in the system, is given. This discussion revolves around how the system differs from, improves upon, and complements other textual approaches.

1.2 Paper's Organization

This paper is structured as follows: First, we frame our approach by comparing it to existing approaches in textual affect sensing, and briefly motivating the approach from the perspective of cognitive psychology. Second, we introduce the Open Mind Commonsense knowledge base, which is the basis of the first implementation of our approach. Third, we give a detailed treatment of the linguistic affect models used to evaluate text. Fourth, we present the architecture of the textual affect sensing engine. Fifth, we discuss how we evaluated our approach in the context of an experimental application. The paper concludes with a discussion of the implications of this work to the development of affective user interfaces, and proposals for future work.

2. WHAT IS THE VALUE OF ANOTHER APPROACH?

There are already a handful of approaches to textual affect sensing, and a fair question to ask is: "Why do we need another one?" This section hopes to answer this question by examining the strengths and weaknesses of existing approaches. As our argument will show, a large-scale, real-world knowledge approach addresses many of these limitations.

2.1 Existing Approaches

Existing approaches can be grouped into the following categories, with few exceptions: 1) keyword spotting, 2) lexical affinity, 3) statistical natural language processing, and 4) hand-crafted models.

2.1.1 Keyword Spotting

The most naïve approach and probably also the most popular because of its accessibility and economy. Text is classified into affect categories based on the presence of fairly unambiguous affect words like "distressed", "enraged," and "happy." Elliott's Affective Reasoner [4], for example, watches for 198 affect keywords (e.g. distressed, enraged), plus affect intensity modifiers (e.g. extremely, somewhat, mildly), plus a handful of cue phrases (e.g. "did that", "wanted to"). Ortony's Affective Lexicon [12] provides an often-used source of affect words grouped into affective categories. The weaknesses of this approach lie in two areas: poor recognition of affect when negation is involved, and reliance on surface features. About its first weakness: while the approach will correctly classify the sentence, "today was a happy day," as being happy, it will likely fail on a sentence like "today wasn't a happy day at all." About its second weakness: the approach relies on the presence of obvious affect words which are only surface features of the prose. In practice, a lot of sentences convey affect through underlying meaning rather than affect adjectives. For

example, the text: "My husband just filed for divorce and he wants to take custody of my children away from me," certainly evokes strong emotions, but use no affect keywords, and therefore, cannot be classified using a keyword spotting approach.

2.1.2 Lexical Affinity

Slightly more sophisticated than keyword spotting. Detecting more than just obvious affect words, the approach assigns arbitrary words a probabilistic "affinity" for a particular emotion. For example, "accident" might be assigned a 75% probability of being indicating a negative affect, as in "car accident," "hurt by accident." These probabilities are usually trained from linguistic corpora. Though often outperforming pure keyword spotting, we see two problems with the approach. First, lexical affinity, operating solely on the word-level, can easily be tricked by sentences like "I avoided an accident," (negation) and "I met my girlfriend by accident" (other word senses). Second, lexical affinity probabilities are often biased toward text of a particular genre, dictated by the source of the linguistic corpora. This makes it difficult to develop a reusable, domain-independent model.

2.1.3 Statistical Natural Language Processing

This is another approach which has been applied to the problem of textual affect sensing. By feeding a machine learning algorithm a large training corpus of affectively annotated texts, it is possible for the system to not only learn the affective valence of affect keywords as in the keyword spotting approach, but such a system can also take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. Statistical methods such as latent semantic analysis (LSA) have been popular for affect classification of texts, and have been used by researchers on projects such as Gortzel's Webmind [5]. However, statistical methods are generally semantically weak, meaning that, with the exception of obvious affect keywords, other lexical or co-occurrence elements in a statistical model have little predictive value individually. As a result, statistical text classifiers only work with acceptable accuracy when given a sufficiently large text input. So while these methods may be able to affectively classify the user's text on the page or paragraph-level, they will not work well on smaller text units such as sentences.

2.1.4 Hand-Crafted Models

In the tradition of Schank and Dyer, among others, affect sensing is seen as a deep story understanding problem. Dyer's DAYDREAMER models affective states through hand-crafted models of affect based on psychological theories about human needs, goals, and desires. [2] Because of the thorough nature of the approach, its application requires a deep understanding and analysis of the text. The generalizability of this approach to arbitrary text is limited because the symbolic modeling of scripts, plans, goals, and plot units must be hand-crafted, and a deeper understanding of text is required than what the state-of-the-art in semantic parsing can provide.

2.2 An Approach Based on Large-Scale Real-World Knowledge

Given a multitude of existing approaches, why develop a new approach? While existing approaches have their applications, they all fail at the robust affect classification of small pieces of domain-independent text such as sentences. We argue that the

ability to sense differences in affect when progressing from sentence to sentence is important to a host of interactive applications that won't work given page- or half-page- level sensing capabilities. Examples of such applications include synthetic agents that want to give affective response to the user input at the sentence level, affective text-to-speech systems, and context-aware systems where user utterances are sparse.

We suggest that using large-scale real-world knowledge to tackle the textual affect-sensing problem is a novel approach that addresses many of the robustness and size-of-input issues associated with existing approaches.

Rather than looking at surface features of the text like in keyword spotting, our approach evaluates the affective qualities of the *underlying semantic content* of text. Real-world knowledge allows us to sense the emotions of text even when affect keywords are absent.

Whereas semantically weaker statistical NLP require page or half-page inputs for reasonable accuracy, semantically stronger commonsense *knowledge* can sense emotions on the sentence-level, and thereby enable many interesting applications mentioned above.

Hand-crafted models tend to have smaller coverage and require deeper semantic understanding than can be achieved over arbitrary text domains. In utilizing a large-scale commonsense knowledge base like OMCS, we avoid having to hand-craft knowledge, and benefit from the robustness derived from the greater breadth and coverage offered by such a resource.

Though in some ways similar to lexical affinity (i.e. concepts and everyday situations have a probabilistic emotional valence), our approach is more general and robust. Unlike lexical affinity, our approach models affect at the event- (or sentence-) level, and at the conceptual level; therefore, our text analyzer is less likely to be tricked by structural features like negation or ambiguity at the word-level. In addition, our knowledge comes uniquely from a large body of commonsense, whereas lexical affinity typically mines its statistical model from annotated corpora and dictionaries.

Though in this paper our commonsense-based approach is discussed and implemented in isolation of these other aforementioned approaches, it should be noted that all of these methods have their merits and can play a role in mutual disambiguation and mutual reinforcement.

2.3 Brief Grounding for Our Approach

The approach put forth in this paper entails the notion that there is some user-independent commonality in people's affective knowledge of and attitudes toward everyday situations and the everyday world which is somehow connected to people's commonsense about the world. Support for this can be found in the works of, inter alia, Aristotle, Damasio, Ortony [12], W. James [6], and Minsky [9]. Aristotle, Damasio, and Ortony have explained that emotions are an integral part of human cognition of the everyday world, and Minsky has gone further to suggest in *The Emotion Machine*, that much of people's affective attitudes and knowledge is an integral part of their commonsense model of the world. Psychologist William James also noted that, just as with the rest of commonsense, the recognition of emotion in language depends on traditions and cultures, so people may not always understand another culture's expression of emotions.

Having framed our approach in terms of existing approaches and theoretical considerations, the rest of this paper will focus on more practical considerations, such as how a small society of commonsense affect models was constructed and integrated into a textual affect sensing architecture.

3. THE OPEN MIND COMMONSENSE CORPUS

Our approach relies on having large-scale real-world knowledge about people's common affective attitudes toward situations, things, people, and actions. If we want our affective sensing engine to be robust, we will have to supply it with a great breadth of knowledge that reflects the immensity and diversity of everyday knowledge.

Generic commonsense knowledge bases are the best candidate sources of such knowledge because affective commonsense is generally a subset of this knowledge, and such knowledge bases are generally rather large, on the order of hundreds of thousands to millions of pieces of world knowledge.

We are aware of three large-scale generic knowledge bases of commonsense: Cyc [7], Open Mind Common Sense (OMCS) [14], and ThoughtTreasure [10]. Cyc is the largest of the three, with over 3 million assertions about the world, followed by OMCS, with close to half a million sentences in its corpus. ThoughtTreasure has around 100,000 concepts and relations.

The implementation discussed in this paper mines knowledge out of OMCS because its English-sentence representation of knowledge is rather easy to manipulate and analyze using shallow language parsers. In the future, we expect to also incorporate knowledge from the other two commonsense knowledge sources. One caveat is that many of the analysis techniques and possible knowledge representations may be specific to the representation of the corpus used. Our system uses OMCS, so the discussion of our system necessarily entails discussion of some OMCS-specific methods.

In OMCS, commonsense is represented by English sentences that fit into 20 or so sentence patterns expressing a variety of different commonsense relations between concepts. Here are some examples of knowledge in OMCS (sentence pattern words are italicized):

(Non-affective)

- *An activity* a doctor *can do* is examine the patient.
- *You are likely to find* rollercoasters *in* an amusement park.
- *The effect of* eating dinner *is* loss of appetite.

(Affective)

- *Some people find* ghosts *to be* scary.
- *A person wants* popularity.
- *A consequence of* riding a rollercoaster *may be* excitement.

From OMCS, we first extract a subset of the sentences which contain some affective commonsense. This represents approximately 10% of the whole OMCS corpus. The identification of these sentences is heuristic, accomplished mainly through keyword spotting of known emotion adjectives (e.g. "happy," "sad," "frightening,"), nouns (e.g. "depression," "delight," "joyous"), and verbs (e.g. "scare," "cry," "love"), taken mainly from Or-

tony’s Affective Lexicon [12]. These affect keywords serve as “emotion grounds” in sentences, because their affective valences are fairly stable across different contexts.

4. A SMALL SOCIETY OF COMMONSENSE-BASED LINGUISTIC AFFECT MODELS

After identifying a subset of the commonsense knowledge that pertains to emotions, we build a commonsense affect model enabling the analysis of the affective qualities of a user’s text. In truth such a model is a small society of different models that compete with and complement one another. All of the models have homogeneously structured entries, each of which have a value of the form:

[*a* happy, *b* sad, *c* anger, *d* fear, *e* disgust, *f* surprise]

In each tuple, *a-f* are scalars greater than 0.0, representing the magnitude of the valence of the entry with respect to a particular emotion.

As a starting point, we work with the six so-called “basic” emotions enumerated above, based on Ekman’s research on universal facial expressions [3]. It should be noted that our approach can be grounded in any set of “basic emotions” which can be discerned through affect keywords, which include, most prominently, sets proposed by Ekman, Frijda, W. James, and Plutchik. For a complete review of proposals for “basic emotions”, see [12].

4.1 The Models

In the remainder of this section, we give an in-depth review of each of the affect models generated from OMCS, we talk briefly about how we generate them from OMCS, and we discuss smoothing models on the inter-sentence level.

4.1.1 Subject-Verb-Object-Object Model

This model represents a declarative sentence as a subject-verb-object-object frame. For example, the sentence “Getting into a car accident can be scary,” would be represented by the frame:

[<Subject>: ep_person_class*, <Verb>: get_into, <Object1>: car accident, <Object2>:]

whose value is:

[0 happy, 0 sad, 0 anger, 1.0 fear, 0 disgust, 0 surprise]

In this example, we refer to “scary” as an “emotion ground” because it confers an affective quality to the event in the sentence by association.

In this sentence, there are two verb chunks, “getting into,” and “can be.” “Can be” refers to the relation between an event and an emotion, so this relation is used to assign the event “getting into a car accident” a value. For the event phrase, the subject is omitted, but we know from this relation (sentence template) in OMCS that the implicit subject is a person, so we fill the subject slot with a default person object. The verb is “get_into” insofar as it is a phrasal verb. The object1 slot is a noun chunk in this case, but may be an adjective chunk. The object2 slot is empty in this example, but in general, either object slot may be noun and adjective chunks, prepositional phrases or complement clauses.

This example does not cover the model’s treatment of negation or multiple SVOOs in one sentence. Negation is handled as a modifier to a subject, object, or verb. If there are multiple verb chunks

in a sentence, and thus multiple SVOOs, then each a heuristic disambiguation strategy will try to infer the most relevant candidate and discard the rest.

The strength of this model is accuracy, as it preserves sentence-level event context. SVOO is the most specific of our models, and best preserves the accuracy of the affective knowledge. Proper handling of negations prevents opposite examples from triggering an entry. The limitation of SVOO however, is that because it is rather specific, it will not always be applicable. We try to make SVOO slightly more robust by semantic class generalization techniques, which we discuss later in the paper.

4.1.2 Concept-Level Unigram Model

For this model, concepts are extracted from each sentence. By concepts, we mean verbs, noun phrases, and standalone adjective phrases. Concept, which are obviously affectively neutral by themselves (e.g. “get,” “have”) are excluded using a stop list. Each concept is given the value of the emotion ground in the sentence. For example, in the sentence: “Car accidents can be scary,” the following concept is extracted and is given a value:

[<Concept>: “car accident”]

Value:

[0 happy, 0 sad, 0 anger, 1.0 fear, 0 disgust, 0 surprise]

Negations are handled roughly by fusing the prefix “not_” to the affected verb. Noun phrases which contain adjectival modifiers are generalized by stripping the adjectives. Then, both the original and generalized noun phrases are added to the model, with the generalized NP necessarily receiving a discounted value.

Concept-level unigrams are not as accurate as SVOOs because they relate concepts *out of sentence-level context* to certain affective states. However, this model is more often applicable than SVOO because it is more independent of the surface structure of language (the specific syntax and word-choices through which knowledge is conveyed).

4.1.3 Concept-Level Valence Model

This model differs from the above-mentioned concept-level unigram model in the value. Rather than the usual six-element tuple, the value is just a vector between -1.0 and 1.0 , indicating that a word has positive or negative connotations.

Evaluating valence affords us the opportunity to incorporate external semantic resources (e.g. dictionaries) that give us very approximate into our affect model.

Associated with this model are hand-coded rules to interpret concept-level valences at the event level. For example, knowing the valences of “wreck” and “my car,” we can deduce that the sentence, “I wrecked my car” has negative affect. To make this deduction, we invoke the following rule:

narrator neg-verb pos-object \rightarrow neg-valence

I WRECKED MY CAR

Although this model does not give us complete mappings into the six emotions, it produces a more reliable *gist* than the concept-level unigram model because it takes event-level context into account. It is useful in disambiguating a story sentence that the other models judge to fall on the cusp of a positive emotion and a negative emotion.

4.1.4 Modifier Unigram Model

This model assigns a six-emotion tuple to each verb and adverbial modifier found in a sentence. The motivation behind this is that sometimes modifiers are wholly responsible for carrying the emotion of a verb or noun phrase, like in the sentences:

“Moldy bread is disgusting”, “Fresh bread is delicious”.

4.2 Generating Models

In constructing each of the aforementioned models, we first choose a bag of affect keywords, pre-classified into the six basic emotions. These words act as “emotion grounds” with which we can interpret the OMCS sentences. To build up the models, we make a first pass in which emotion grounds propagate their value to other concepts/SVOOs/modifiers (this is model-specific) in the same sentence. This propagation can be likened to spreading activation or undirected inference. To improve coverage (number of concepts with an affect value), we make a second and a third pass over the entirety of OMCS, propagating the affect value of concepts that have a non-zero value to concepts in the same sentence which have a zero value. After each propagation, the affect value is discounted by a factor d . An example of propagation for the concept-level unigram model is given here (six-tuple refers to happy, sad, anger, fear, disgust, and surprise):

Something exciting is both happy and surprising.

(pass 1: exciting: [1,0,0,0,0,1])

Rollercoasters are exciting.

(assume $d=0.5$, pass 2: rollercoaster: [0.5,0,0,0,0,0.5])

Rollercoasters are typically found at amusement parks.

(pass 3: amusement park: [0.25,0,0,0,0,0.25])

With the completed commonsense-based affect models, we can evaluate texts and on the sentence-level, and sense affect in terms of the six Ekman emotions or determine affect as “neutral,” which means that there is not enough information or confidence to make a classification. Each sentence is classified by running each of the models on that sentence and applying a weighted scoring function to the results of each model.

4.3 Smoothing Models

After sentences have been annotated with one of the six basic emotions or “neutral,” we apply various techniques aimed at smoothing the transition of emotions from one sentence to the next. These techniques operate on the assumption that the sentences in a story have coherence, and therefore, some affective continuity.

4.3.1 Decay

The most basic smoothing is decay. A good example of this is when a sentence annotated as “surprised” is followed by two or more neutral sentences. The neutral annotations are most likely the result of sentences not handled well by our models. And the emotion surprise is not likely to transform abruptly into neutral. Decay in this case would revise the annotation of the first neutral sentence to 50% surprised, or whatever the decay rate is decided to be.

4.3.2 Interpolation

Another related smoothing technique is interpolation. For example, if a neutral sentence is between two angry sentences, then one of the hand-coded interpolation rules will revise the annotation of

the middle sentence to 75% angry, or whatever the interpolation factor is set to be.

4.3.3 Global mood

A third smoothing technique is global mood. In storytelling, larger sections of text such as the paragraph or even the entire story establish and preserve moods. Computationally, we can analyze the raw scores and affect annotations to figure out the mood of a paragraph and of an entire story. We can then add a memory of these moods onto each sentence within the paragraph or in the story, respectively.

4.3.4 Meta-Emotion

A fourth technique, and perhaps the most interesting, is the *meta-emotion*. We observe that certain emotions not part of the six basic emotions can emerge out of patterns of the six basic emotions. This is highly desirable because it gives the six basic emotions more expressive ability, and allows them to have more RGB-like properties (to make an analogy to computer color blending). Several examples of meta-emotions are given below:

Frustration – Repetition of low-magnitude anger

Relief – Fear followed by happy

Horror – Sudden high-magnitude fear

Contentment – Persistent low-level happy

Of course, as with any pattern recognition, meta-emotion detection is not fool-proof, but since meta-emotions are meant to express more natural transitions between emotions, they will generally fail-softly, in that the wrong meta-emotion won’t be far off the right meta-emotion.

Having discussed in great detail our models and methods in isolation, we now synthesize these ideas into an overall system architecture.

5. ARCHITECTURE OF AN AFFECT SENSING ENGINE

The architecture of the affect sensing engine follows closely from the approach outlined in the previous section. It can be viewed in two parts: 1) the Model Trainer; and 2) the Text Analyzer.

5.1 Model Trainer Architecture

The Model Trainer architecture has three sequential modules: 1) Linguistic Processing Suite; 2) Affective Commonsense Filter & Grounder, and 3) Propagation Trainer.

1) The raw OMCS corpus of ½ million sentences first undergoes linguistic processing. Because OMCS sentences follow a sentence-template based ontology we first rewrite such sentences as a binary relation. To satisfy the representational needs of the different models, we also perform a suite of linguistic processing including part-of-speech tagging, phrase chunking, constituent parsing, subject-verb-object-object identification, and semantic class generalization (e.g. “I” → narrator; “People” → ep_person_class).

2) From this parsed OMCS corpus, we use emotion ground keywords classified by the six Ekman emotions, first to filter the affective commonsense from the whole OMCS, and second to tag the emotion keywords with “grounds” in preparation for training the models.

3) In the third module, the propagation trainer propagates the affect valence from the emotion grounds to concepts related through commonsense relations, and from those concepts to yet other concepts. Each propagation discounts the valence by some factor *d* e.g. 0.5. This propagation can be viewed as analogous to undirected inference, or spreading activation over a semantic network of concept nodes connected by commonsense relation edges.

5.2 Text Analyzer Architecture

The text analyzer architecture can be decomposed into five sequential modules: 1) Text Segmenter, 2) Linguistic Processing Suite, 3) Story Interpreter, 4) Smoother, 5) Expressor

The incoming story text is first segmented into paragraphs, sentences, then into independent clauses, which are the smallest story units that can capture an event.

In the story interpreter module, each parsed and processed sentence is evaluated against the trained models and a weighted scoring function generates a six-tuple score. Disambiguation metrics help to map this final score into an emotion annotation. In the output of this module each sentence will be annotated with one of the six basic emotions, or “neutral” and will also be annotated with the total score.

The next module, Smoother, performs pattern matching over these emotion annotations and re-annotates each sentence to reflect the smoothing strategies of valence delay, interpolation, global mood, and meta-emotions. The new annotations will have the form: `<emotion='e1' valence='50%'>` but may have additional `global_mood` and `global_mood_valence` fields. The re-annotated sentences are then expressed by the Expressor, which symbolizes a placeholder for some output modality or application.

6. EVALUATION THROUGH A PROTOTYPE APPLICATION

For an interesting way to test the technical merits of our approach, we incorporated our affect sensing engine into Chernov face - style feedback in an affectively responsive email composer called EmpathyBuddy. This experimental system allows us to test the robustness of our approach against plausible everyday use, rather than against a formal corpus. We also motivate an in-application test with the curiosity: “Can this approach work well enough to make a practical impact on the design of affective user interfaces?” We believe that the user study evaluation shows just that.

This section is subdivided into three subsections: 1) an overview of EmpathyBuddy user interface; 2) a scenario of a user interaction with the system and 3) a summary of our user study evaluation. For a fuller discussion on EmpathyBuddy and its evaluation, see [8].

6.1 EmpathyBuddy

EmpathyBuddy is an email browser with a Chernov-style face embedded in its window that emotes in sync with the affective quality of text being typed by the user (shown in Figure 1). EmpathyBuddy’s faces express the six basic Ekman emotions plus decayed versions of the six basic emotions, and also four transitory meta-emotions.

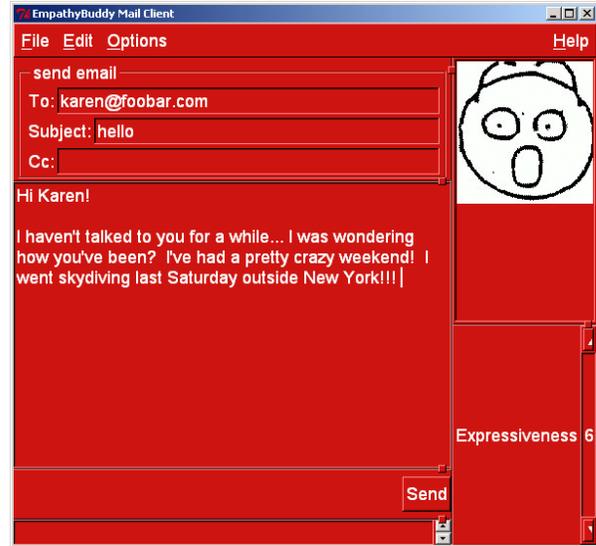


Figure 1. EmpathyBuddy Email Agent

The layout of the email browser is meant to be familiar to the user. At the upper left corner are email header fields. In the lower left is the email body text box. A demon frequently polls the text in the body and analyzes the text using the emotion sensing engine. The avatar in the upper left changes faces to try to match the dynamic affective context of the story. Because of limitations of pre-drawn facial expressions, aspects of the affect annotations outputted by the sensing engine, such as global mood, cannot be fully expressed.

6.2 A User’s Encounter with the System

Let us walk through a scenario in which the user writes an email to her mom telling her that she bought a new car but uneventfully wrecked it. Thankfully though, she was not hurt. Figure 2 gives a walkthrough of this scenario. This is a useful scenario because it highlights some of the more advanced behavior of the affect sensing engine.



Figure 2: User Scenario

In the sentence, “it’s a gorgeous new sports car!” the engine’s models are not yet certain about the affect of sports cars. They show that this sentence is ambiguous and that it could be one of two emotions: surprise or anger. Three disambiguation features all concluded that the correct emotion was surprise. First, according to the conceptual valence model, this sentence was characterized by positive emotion, and since surprise is positive whereas anger is not, this feature chose surprise. Second, the previous sentence disambiguation feature prefers surprise because that emotion also occurred in the previous sentence. Third, according to the fail-soft strategy of only showing anger and disgust in extreme cases, anger would have also been disallowed from occurring here.

The last two sentences are a good illustration of meta-emotion smoothing. The sentence, “I got into an accident and I crashed it” evoked fear, while the sentence “Thankfully, I wasn’t hurt” evoked happy. However, it would seem rather unnatural to change suddenly from a frightened expression to a happy expression. Humans don’t easily forget the anxiety they held two seconds ago! The affect sensing engine recognizes the pattern of moving from fear to happy as a meta-emotion. It decides that happy should actually be revised into relief (from anxiety). It does so accordingly, and the emotion displayed by the EmpathyBuddy avatar reflects this.

6.3 User Study and Evaluation

A 20-person user study was conducted to quantitatively measure the “performance” of the EmpathyBuddy email browser in a practical scenario. Each user was asked to perform the task: “send an email to someone and tell them a brief but interesting story about something you did recently.” They were asked to type into three interfaces, given in random order. All three interfaces use the look and feel of the EmpathyBuddy mail client, differing only in the behavior of the face. The baseline is “Neutral face,” which displays a static neutral-looking face. To control for the deliberate selection of faces by the affect sensing engine, a second baseline called “Alternating, Randomized faces” was created, which displays a randomly selected face at the end of each sentence. The third client is EmpathyBuddy.

Users were asked to evaluate the system against four aspects: entertainment, interactivity, intelligence, and adoption, as shown in Figure 3. (Each line segment bisecting the ends of the bars represents one standard deviation above and below the mean.)

The results of our user testing suggest that the textual affect sensing engine works well enough to bring measurable benefit to an affective user interface application. In light of the fact that user’s inputs closely mirror real application use, the results also speak of the robustness of our engine.

We were gratified that the client using the affect sensing engine was judged to behave the most intelligently of the three, and that users expressed enthusiasm for wanting to adopt EmpathyBuddy as their regular mail client.

Two results were rather unexpected. First, the randomized faces client was judged to be slightly more entertaining than EmpathyBuddy. User feedback suggests one possible explanation: the randomized faces had a quicker turnover in faces and were more outrageous (e.g. showing disgust, anger with equal probability)! Though it is satisfying to note that quick turnover of faces was not as important as the relevance of faces, to the idea of interactivity.

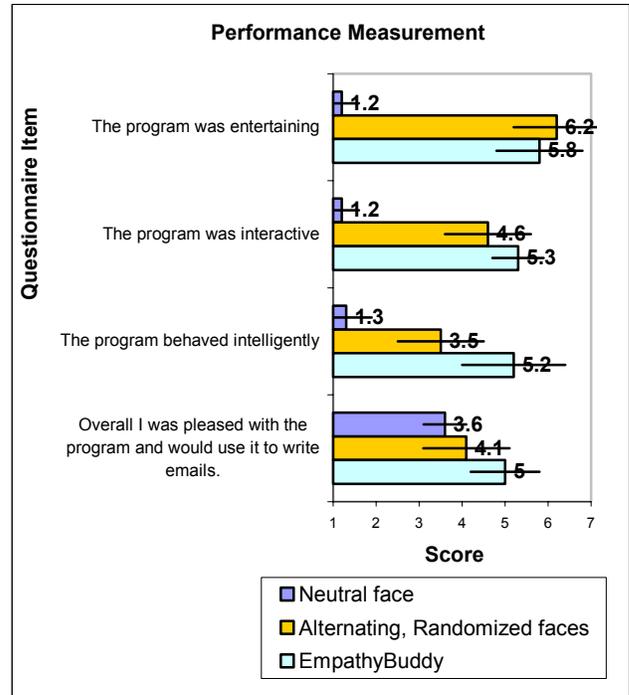


Figure 3. User Testing Questionnaire Results

Second, the randomized faces client scored better than expected on the question of system adoption, even though there was no AI behind it. Users told us that they were so bored of their static email clients that they were more than willing to flock to something—anything—more interactive and entertaining!

7. CONCLUSION AND FUTURE WORK

This paper presents a novel approach to classifying the affect of text into the six Ekman emotion categories of: happy, sad, angry, fearful, disgusted, and surprised. By leveraging a real-world knowledge base called Open Mind with 400,000 pieces of knowledge, we can evaluate the affective nature of the *underlying semantics* of sentences, and in a robust way. This approach addresses many of the limitations of four other tried approaches to textual affect classification.

First, while keyword spotting relies on surface features of text, our more robust approach evaluates the affective nature of the underlying story semantics. Second, lexical affinity is not robust because structural features, such as negation, can trick it; in contrast, our approach employs competing linguistic models that factor in structural features. Third, while statistical techniques can be effective, they require a large input and they are not transparent to explanation. Our approach is robust enough to work on the sentence level, which is an important barrier that allows for much more interactive behavior. It is also easy to explain the reasons for a particular classification, by examining a trace of the commonsense inferences involved in an evaluation decision. Fourth, our approach is more flexible than hand-crafted models because the knowledge source, OMCS, is a large, multi-purpose, collaboratively built resource. Compared to a hand-crafted model, OMCS is more unbiased, domain-independent, easier to produce and extend, and easier to use for multiple purposes.

To test the suitability of the approach in isolation for affective interactive applications, we incorporated the affect sensing engine into Chernov face -style feedback in an affectively responsive email browser called EmpathyBuddy. User evaluations suggest that the approach and implementation are robust enough to be used by everyday people in an everyday task like email.

The most immediate future work will examine how a real-world knowledge approach can complement each of the existing approaches, and will evaluate the precision and recall of all the approaches against a standardized test corpus or human annotator. One of the limitations of the current approach is that the affect sensing operates rather independently of the user and story contexts, instead, using only commonsense context. As a result, the sensing engine in its current state may be inadequate for long-term user-to-agent interactions in which it is expected that the agent learns about the user, as a friend might; it may also be inadequate for interpreting the affect of heavily contextualized stories such as a novel, for example. In future work, we hope to overcome some of these limitations by investigating how our approach can be extended to incorporate a user model, and perform some tracking of story context (requiring deeper story understanding). In this paper we refer to our approach as a method for *affect sensing*; but if we manage to incorporate important contexts into a future system in a deep way, then we might well be on the way to the holy grail of *affect understanding*.

Finally, in future work, we wish to examine some of the many possible uses for a rather robust, plausible, domain-independent, sentence-level affect analyzer, beyond the trivial application presented in this paper. Such technology could aid the development of affective user interfaces for synthetic agents, emotional characters in multi-user gaming, affective storytelling, emotional prosody, and other applications we have yet to dream up.

8. ACKNOWLEDGMENTS

The authors thank Push Singh and Cindy Mason for their invaluable feedback during the preparation of this paper.

9. REFERENCES

- [1] Bates, J. The Role of Emotion in Believable Agents. Communications of the ACM. 37(7), 1994, pp. 122-125.
- [2] Dyer, M.G. Emotions and Their Computations: Three Computer Models. Cognition and Emotion, Vol. 1, No. 3, 323-347, 1987.
- [3] Ekman, P. Facial expression of emotion. American Psychologist, 48, 384-392. 1993.
- [4] Elliott, C. The Affective Reasoner: A Process Model of Emotions in a Multi-agent System. PhD thesis, Northwestern University, May 1992. The Institute for the Learning Sciences, Technical Report No. 32.
- [5] Goertzel, B., Silverman, K., Hartley, C., Bugaj, S., and Ross, M. The Baby Webmind Project. Proceedings of AISB 2000.
- [6] James, W. What is an emotion? In C. Calhoun & R. C. Solomon (Eds.), What is an emotion? Classic readings in philosophical psychology (pp. 127-141). New York: Oxford University Press. 1984.
- [7] Lenat, D.B. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11):33-38, 1995.
- [8] Liu, H., Lieberman, H., and Selker, T. Automatic Affective Feedback in an Email Browser. MIT Media Lab Software Agents Group Technical Report. November, 2002. At: <http://web.media.mit.edu/~hugo/>.
- [9] Minsky, M. The Emotion Machine. Pantheon, not yet available in hardcopy. Drafts available: <http://web.media.mit.edu/~minsky/>.
- [10] Mueller, E.T. ThoughtTreasure: A natural language/commonsense platform (Online). <http://www.signiform.com/tt/htm/overview.htm>. 1998.
- [11] Nass, C.I., Stener, J.S., and Tanber, E. Computers are social actors. In Proceedings of CHI '94, (Boston, MA), pp. 72-78, April 1994.
- [12] Ortony A., Clore, G. L., and Collins, A. The cognitive structure of emotions. New York: Cambridge University Press. 1988
- [13] Picard, R. W., Affective Computing, The MIT Press, Mass., 1997.
- [14] Singh, P., The public acquisition of commonsense knowledge. Proceedings of AAAI Spring Symposium. Palo Alto, CA, AAAI. 2002.