ORNL/TM-2010/277

Inferring Group Processes From Computer-Mediated Affective Text Analysis

February 2011

Prepared by
Jack C. Schryver
Edmon Begoli
Ajith C. Jose
Christopher Griffin

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2010/277

Computational Sciences and Engineering Division

INFERRING GROUP PROCESSES FROM COMPUTER-MEDIATED AFFECTIVE TEXT ANALYSIS

Jack C. Schryver
Edmon Begoli
Ajith C. Jose
Christopher Griffin

Date Published: February 2011

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

Page

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

ABSTRACT

Political communications in the form of unstructured text convey rich connotative meaning that can reveal underlying group social processes. Previous research has focused on sentiment analysis at the document level, but we extend this analysis to sub-document levels through a detailed analysis of affective relationships between entities extracted from a document. Instead of pure sentiment analysis, which is just positive or negative, we explore nuances of affective meaning in 22 affect categories. Our affect propagation algorithm automatically calculates and displays extracted affective relationships among entities in graphical form in our prototype (TEAMSTER), starting with seed lists of affect terms. Several useful metrics are defined to infer underlying group processes by aggregating affective relationships discovered in a text. Our approach has been validated with annotated documents from the MPQA corpus, achieving a performance gain of 74% over comparable random guessers.

1. INTRODUCTION TO AFFECTIVE TEXT ANALYSIS

Affective text analysis is located at the intersection of the fields of affective computing and text analysis (see Solka 2008 for a review of this field). Affective computing is currently one of the most active research topics in computer science and cognitive science. According to Tao and Tan (2005), affective computing as a scientific research field is the attempt to “assign computers the human-like capabilities of observation, interpretation and generation of affect features.” Although the affective computing field as a whole is more concerned with the development of new methodologies of human-computer interaction, an important component of affective computing capability is “affect understanding.” The artificial intelligence component at the human-computer interface must explore, detect, and extract the range of affective states of users in order to effectively interact with them across the full spectrum of communication. Affective text analysis is confined to the analysis of affect in text only, without benefit of multimedia or speech processing. In a particular sense, this makes the problem more challenging, but it also simplifies analysis by focusing on a single channel of communication.

Text analysis is an applied research field that is largely dependent on achievements in the academic discipline of natural language processing (NLP) or computational linguistics. NLP is a very large area of basic research, encompassing areas of linguistic study that are important requisites of the present work, including named entity recognition, word stemming, word tokenization, sentence detection, and part-of-speech processing, to name just a few (e.g., Jurafsky and Martin 2009).

Affective states are a superset of the basic emotions (Ekman 1999), which are a limited set of recognizable psychophysiological states. Affect in text belongs to a branch of semantics that focuses on the connotative meaning of text, in contrast to denotation. Denotative meaning lies in the traditional cognitive domain where words and phrases merely convey explicit declarative knowledge encoded in propositions. A word denotes its dictionary meaning and nothing more. Connotation invites associations of words to other words and to their evaluative aspects, or affective meaning (Allan 2007). The manner in which affective meaning is expressed in language is somewhat culture-dependent (Besnier 1990).

Textual affect can be either overtly or covertly communicated. The sentence “John loves Mary” is an explicit communication of affect. Compare this to “The economic stimulus plan is obviously socialist.” The second sentence represents a more subtle attempt to link the concept of the ‘stimulus plan’ to a negative evaluation, because in contemporary American politics the term “socialist” is pejorative. The possibility that overt or subtle communication of connotative meaning in text can be analyzed automatically opens the door to exciting opportunities to gain deep insight into the “deep structure” operated on by cognitive/affective processes that lie beneath the surface structure of language.

Specifically, affect expressed in a document by group members may be linked to group processes. Discovering links between affect and group process is a significant goal of this research effort.

1.1 MOTIVATION

Intelligence analysis is deeply concerned with social network analysis and identifying states and processes of groups. However, a significant proportion of intelligence data coming from groups and social networks, especially from internet sources such as blogs, email, position papers, forums, tweets, and chat, is textual. Political communications in the form of unstructured text convey rich connotative and affective meaning that can reveal underlying group social processes (e.g., importance of positive affect for group cohesion and negative affect for outgroup antipathy). Therefore, a critical element of deep social network analysis is automated extraction of affective meaning in text. This often depends on accurate reading of the 'subtext' that implicitly refers to commonsense/cultural knowledge, common ground and affect shared by group members.

In a global socio-political landscape where winning hearts and minds is paramount to foreign policy success, there is a pressing need for the development of an "affective radar" to assess group dynamics. Some examples of intelligence targets for which group affect is pivotal are identification of emerging threats and hate groups, assessment of viability of spontaneous foreign political movements, and identification of grassroots social responses to implementation of U.S. foreign policy decisions.

A critical element of deep social network analysis is the automated extraction and classification of affective responses toward various entities of interest, based on a theoretically-motivated affect taxonomy (for such definitions see Ortony and Turner 1990, and its references).

Affects are conveyed through keywords and phrases (n-grams) and their relative position in sentences. Simple sentences like "I am amused" are easy to parse. Sentences with negation, like "I am not amused," represent a more difficult challenge. Finally, sentences that contain words used in more complex contexts like "I went to the amusement park and broke my arm" are even more difficult to parse. In the last example, common knowledge suggests that "amusement park" would usually be associated with positive feelings, while the phrase "broke my arm" clearly suggests a negative feeling.

An important step in the process of affect detection is the extraction of key words and phrases that connote affects, feelings, and emotions (affect lexicon). Several research groups have endeavored to extract emotional content from text automatically (Liu et al. 2003, Strapparava and Valitutti 2004, Valitutti et al. 2004, Lu et al. 2006). Our approach is to construct an affect lexicon starting from seed lists and expanding the lexicon using the PageRank algorithm on graph structures defined by WordNet and other sources.

1.2 DEPTH OF AFFECTIVE TEXT ANALYSIS

Affect text analysis can be executed at multiple levels of analysis:

- *Traditional document-level sentiment analysis* assigns a positive or negative evaluation to a single (implied) entity, concept or event;
- *Extraction of multiple affect categories at a document level* – admire, reproach, love, hate, joy, distress, hope, fear, pride, shame, resentment, satisfaction, etc.;
- *Fine-grained affective analysis* propagates a multiplicity of affective meaning from affect terms to their intended sources and targets at sub-document levels.

Our approach is unique in that it provides coverage at all three levels.

2. BACKGROUND

There are two main divisions in semantic analysis of text, namely: denotation and connotation. Whereas denotation is the explicit meaning of text and carries the cognitive component, connotation refers to the affective meaning or associational aspect of text. The successful communication of connotative meaning often depends on an accurate reading of the ‘subtext’ which refers to implicit commonsense knowledge and affect shared by members of a cultural group. The most basic concept that relays connotative information is sentiment.

2.1 SENTIMENT ANALYSIS AND AFFECTIVE SENSING

Most early work in connotative analysis of text has been in the area of sentiment analysis or opinion mining, which is the assessment of polarity with respect to a topic or document (for a comprehensive review, see Pang and Lee 2008). Polarity is usually defined as the degree of positivity or negativity attached to an object, from the viewpoint of the user (reader). An important application of this research is opinion mining; which is the automated classification of customer feedback on products or movie reviews. Existing sentiment analysis systems tend to yield fair performance at best. However, polarity is not sufficiently fine-grained to provide a solid theoretical foundation for analysis of complex dynamics of groups or social networks. A more complete analysis of affective meaning is needed to fully understand group processes as embedded in text and subtext.

There have been a few previous attempts to perform the research required to automatically extract affective meaning from text using either an affect lexicon or supervised learning techniques. These efforts have mostly been directed at classification of documents or sentences into one of several affect or emotion categories. An important development was WordNet-Affect, which extends WordNet by defining a hierarchy for affective meaning (Strapparava and Valitutti 2004, Valitutti, et al. 2004). The Linguistic Inquiry and Word Count system or LIWC (Pennebaker et al. 2001) was a noteworthy attempt to provide in a software program a psycholinguistic summary of text characteristics at the document level. LIWC performs keyword spotting of affective processes that includes positive emotions in addition to negative emotions such as anxiety, anger, and sadness. Some investigators have specifically attempted to identify affect in text (Liu et al. 2003, Al Masum et al. 2007, Alm et al. 2005, Aman and Szpakowicz 2008, Abbasi et al. 2008) using either an affect lexicon or supervised learning techniques. These efforts have mostly been directed at document-level assessment of affect. Abbasi et al. (2008) studied the presence of violence- and hate-related affect in web forums operated by extremist political groups. This research shows the relevance of affect extraction techniques to the objectives of intelligence analysis.

Liu, Lieberman, and Selker (2003) recognized the significance of commonsense knowledge in extracting connotative/affective meaning from text. Their approach utilized affective knowledge contained in the Open Mind Common Sense (OMCS) database in the construction of their affect lexicon. In particular, they were interested in the kind of real-world knowledge that revealed common-place affective stances toward situations, things, people, organizations, concepts, and events. Sentences that contained affective meaning in OMCS were identified by spotting the “emotion ground” signified by affect-saturated keywords. These emotion grounds were used in models or templates to subsequently extract affect from documents. A similar approach grounded in an affect lexicon augmented with commonsense knowledge from OMCS was followed in He et al. (2004). However, the success of this approach was only partial since incorporation of commonsense knowledge was also burdened by the relatively unstructured sentence structures found in the OMCS database.

A recent paper focused on the graphical display of affect in text. Gobron et al. (2009) proposed a three-dimensional representation of affect. Emotions were extracted from text using both an affect lexicon and a language model classifier. Extracted emotion states were then rendered as facial expressions of virtual humans in a social networking animation.

Our approach is in the spirit of Liu et al. (2003), but instead of OMCS, we used ConceptNet - a successor to OMCS. The developers of ConceptNet (Havasi et al. 2007, Liu and Singh 2004) structured the sentences in OMCS into a more structured, machine-friendly format. The ConceptNet3 database currently contains over 700,000 assertions, and is still being maintained and improved. An assertion has the structure of a sentence or a clause; it joins two concepts (e.g., nouns or noun phrases) with one of 20 pre-defined verb-like relations. An important advantage of ConceptNet is that it is a semantic network that can be searched and manipulated like other graph structures.

We adopted a cognitive theory of emotion (affect) as the taxonomy for modeling affective meaning in text. According to appraisal theory (O'Rorke and Ortony 1994), affect is the response to cognitive evaluations made by individuals and groups to outcomes associated with self, agents, objects, and events. For example, admiration/reproach is a valenced reaction to an approved/disapproved action of another agent. Pride/shame result from a similar evaluation of an action focusing on the self as an agent. O'Rorke and Ortony's taxonomy considers 22 affects resulting from different types of appraisal, but of course this number is somewhat arbitrary and the set can be enlarged. Table 1 lists the 22 affect categories sorted by valence and binary/unary relation.

Table 1. Affect Categories From O'Rorke and Ortony (1991)

Unary		**Binary**	
Positive	**Negative**	**Positive**	**Negative**
Joy	*Distress*	*Happy-for*	*Resentment*
Pride	*Shame*	*Pity*	*Gloating*
Gratification	*Remorse*	*Admiration*	*Reproach*
Satisfaction	*Disappointment*	*Love*	*Hate*
Relief	*Fear-confirmed*	*Gratitude*	*Anger*
		Hope	*Fear*

2.2 AFFECT AND GROUP PROCESSES

A fundamental working hypothesis of this effort is that affective meanings extracted from text can serve as reliable indicators of underlying group processes during the entire group lifecycle. Evidence from the experimental literature confirms the existence of a close relationship between individual/group affect and group processes. For example, a well-known phenomenon studied in the social psychology literature is called Linguistic Intergroup Bias (LIB). The LIB is the "disposition to communicate positive in-group and negative out-group behaviors more abstractly than negative in-group and positive out-group behaviors" (Anolli 2006). To put it differently, people who behave in accordance with stereotypic expectations are described abstractly as typical members of the in-group or out-group; conversely, those who violate stereotypic expectations are described more concretely as individual or isolated cases that represent exceptions to group behavior.

Research on intergroup bias has demonstrated that instances of in-group favoritism and out-group derogation are common in populations (Hewstone, et al. 2002). This research has shown the interconnectedness of cognitive and affective components of this type of bias. For example, manipulating (increasing) positive affect in minimal group settings increases favorable bias toward group members. Positive affective relationships typical of stable groups are undermined and reversed during the process of group schism.

Sani (2005) found that schism is preceded by "the belief that the group identity has been subverted. This belief will prompt negative emotions (i.e., dejection and agitation) and decrease both group identification and perceived group entitativity (i.e., cohesion, oneness). In turn, low group entitativity will reduce the level of group identification. Finally, low group identification and high negative emotions will increase schismatic intentions." The tight coupling of schismatic intentions and emotion/affect demonstrates how effective affect can be as a barometer of group drift toward schism. Affect can both reflect and signal transition between developmental stages of groups (Barsade and Gibson 1998). The midpoint of a work team life is "characterized by considerable anxiety and anticipation about the capacity of the team to complete its goals" (p. 87). If affective terms can be captured and extracted from textual communications authored by group members, analysts may be able to gain valuable insights by treating these affective markers as leading indicators of the developmental stages of groups.

3. TECHNICAL APPROACH TO COMPUTER-MEDIATED AFFECT EXTRACTION

The affect extraction algorithm consists of one offline step and three online steps. The initial offline step of the algorithm is the *construction of an affect lexicon* for each affect category (i.e., identification of terms in English saturated with affect, feeling, and emotion). In order to extract affective meaning in a document three online steps are required. The first step is *extraction of entities and affect terms* in a document. Instead of pure sentiment analysis, which is just positive or negative, we explore nuances of affective meaning in 22 affect categories. The second step involves *determination of specific effects of affect terms in the local context* of a document. The final step employs a variant of the PageRank algorithm to *propagate affective meaning from affect terms to source and target entities*. The functions of the prototype are depicted in Fig. 1. The final step is shown in the box named "Determine affect-entity relationships."

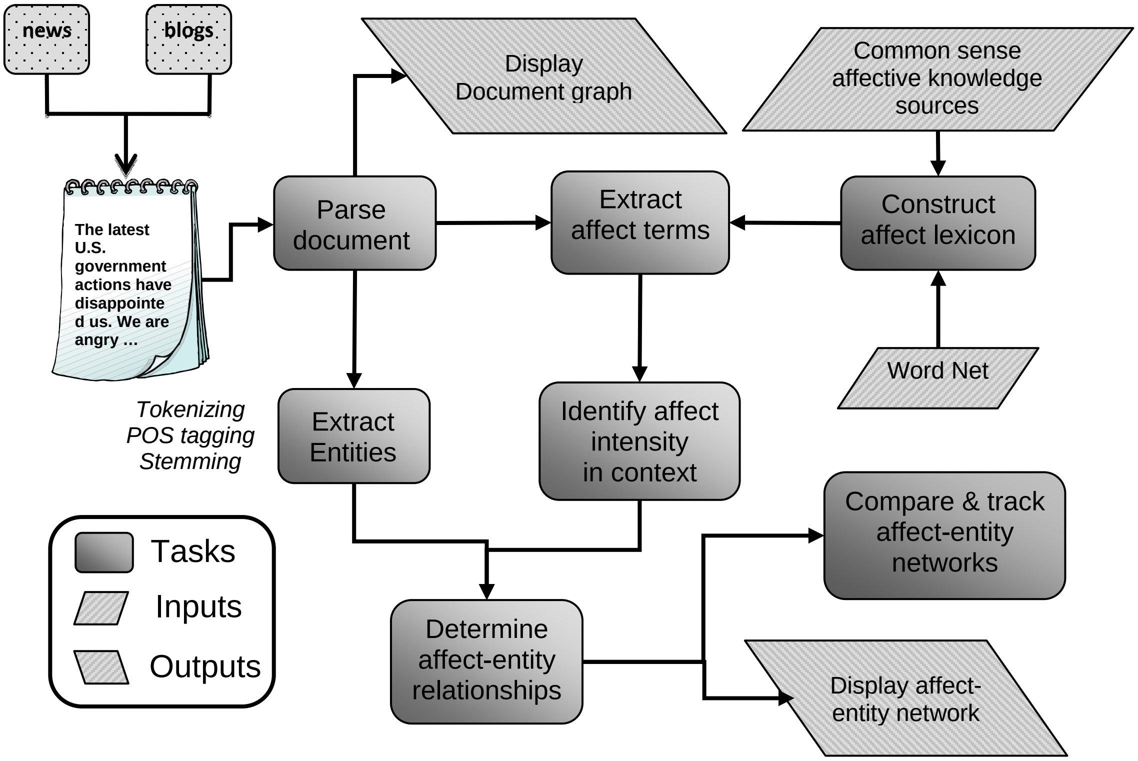

news
blogs
The latest U.S. government actions have disappointe d us. We are angry ...
Tokenizing
POS tagging
Stemming
Parse document
Display Document graph
Extract Entities
Extract affect terms
Identify affect intensity in context
Common sense affective knowledge sources
Construct affect lexicon
Word Net
Determine affect-entity relationships
Compare & track affect-entity networks
Display affect-entity network
Tasks
Inputs
Outputs

Fig. 1. TEAMSTER functional diagram.

The affect propagation algorithm automatically identifies extracts, quantifies, and displays affective relationships among entities in graphical form in our prototype.

The algorithm is endowed with a number of parameters that must be specified before running the algorithm on a text. Most of these parameters are associated with the propagation of affective meaning in a specific document. The parameter values can be manually tuned, but subsequent analysis demonstrated that manually-tuned parameters led to low fitness scores on a set of benchmark documents. Therefore, we performed an optimization study (see Sects. 5 and 6) to discover the set of parameter values that leads to the best fitness score.

3.1 SAMPLE TEXT

For purposes of illustration we work with a simple example of fictitious text explicating political ties in the Middle East from a U. S. analyst perspective. The text should help to expose various aspects of the affect propagation algorithm described in Sects. 3.3 – 3.8. The full text of the sample is given below:

> It has been a time of insecurity for Israel. The U.S. maintains strong support and liking for Israel, just as Israel values its friendship with the U.S. However, Israel harbors a deep animosity and fear of Iran, while Iran does not approve of Israel. On the other hand, Egypt has a moderate liking and respect for Israel.

The sample 57-word text is simple and explicit enough to demonstrate the execution of the affect algorithm while avoiding many of the difficult syntactical constructions and subtle meanings typical of unstructured political text.

3.2 SOFTWARE PROTOTYPE

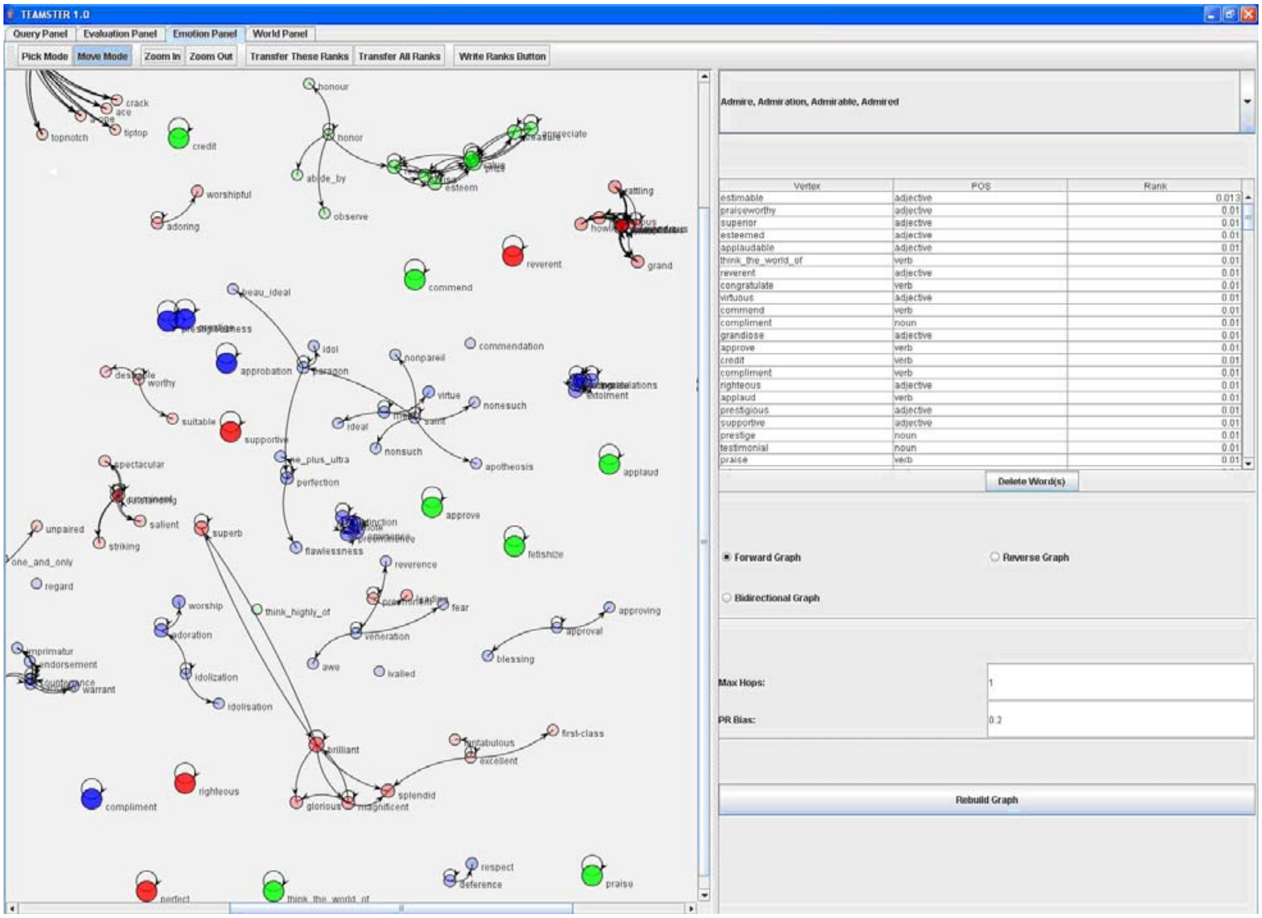
A software prototype was developed in Java to test the affect detection algorithms described in this report, and to provide to the eventual user a practical platform for connotative/affective analysis of text. The prototype, called The Extraction of Affective Meaning and Structure from TExt Resources (TEAMSTER) is designed to automatically extract affect from documents, whereby the results of each step are displayed in different panels at the graphical user interface. The three steps are:

- PageRank-directed construction of the affect lexicon using WordNet, ConceptNet, and other resources that structure knowledge as semantic networks or graphs (emotion panel);
- Sensing of affect terms, entities, and topics in a document (evaluation panel); and
- Propagation of affects between entities or topics and display of a network diagram of the affective relationships among entities in a document (affect analysis panel).

A fourth step, which is the summarization of affect at the document level, is displayed in the textual output. The following subsections describe and illustrate the contents of the output of TEAMSTER. The theoretical foundations underlying the prototype are detailed in Sects. 3.3 – 3.8 (graphical user interface) and Sect. 4 (text output).

3.2.1 Panel Content on the Graphical User Interface

The first step of the algorithm is implemented as an off-line process that produces a set of affect vectors that become the search terms for target documents. Figure 2 shows in the emotion panel that the affect lexicon is visualized as both a list of ranks (right side) and a graph (left side). PageRank operates on the WordNet database to populate an affect lexicon starting with seed lists for every affect category. The clusters on the graph tend to mirror WordNet synsets. The flow of affective meaning proceeds along the directed arcs of the WordNet graph as in normal PageRank; alternatively, the user may request reversed or bi-directional flow. The user also may restrict the maximum number of hops during the random walk, or modify the damping parameter. If the PageRank parameters are modified during a session for some affect category, the user may transfer the new ranks to the document analysis that is displayed in the evaluation pane. This feature allows the user to fine-tune the flow dynamics of affective meaning in order to improve the efficacy of the affect lexicon for the specific document being analyzed.



Admire, Admiration, Admirable, Admired

Vertex	POS	Rank
estimable	adjective	0.013
praiseworthy	adjective	0.01
superior	adjective	0.01
esteemed	adjective	0.01
applaudable	adjective	0.01
think_the_world_of	verb	0.01
reverent	adjective	0.01
congratulate	verb	0.01
virtuous	adjective	0.01
commend	verb	0.01
compliment	noun	0.01
grandiose	adjective	0.01
approve	verb	0.01
credit	verb	0.01
compliment	verb	0.01
righteous	adjective	0.01
applaud	verb	0.01
prestigious	adjective	0.01
supportive	adjective	0.01
prestige	noun	0.01
testimonial	noun	0.01
praise	verb	0.01

● Forward Graph ○ Reverse Graph

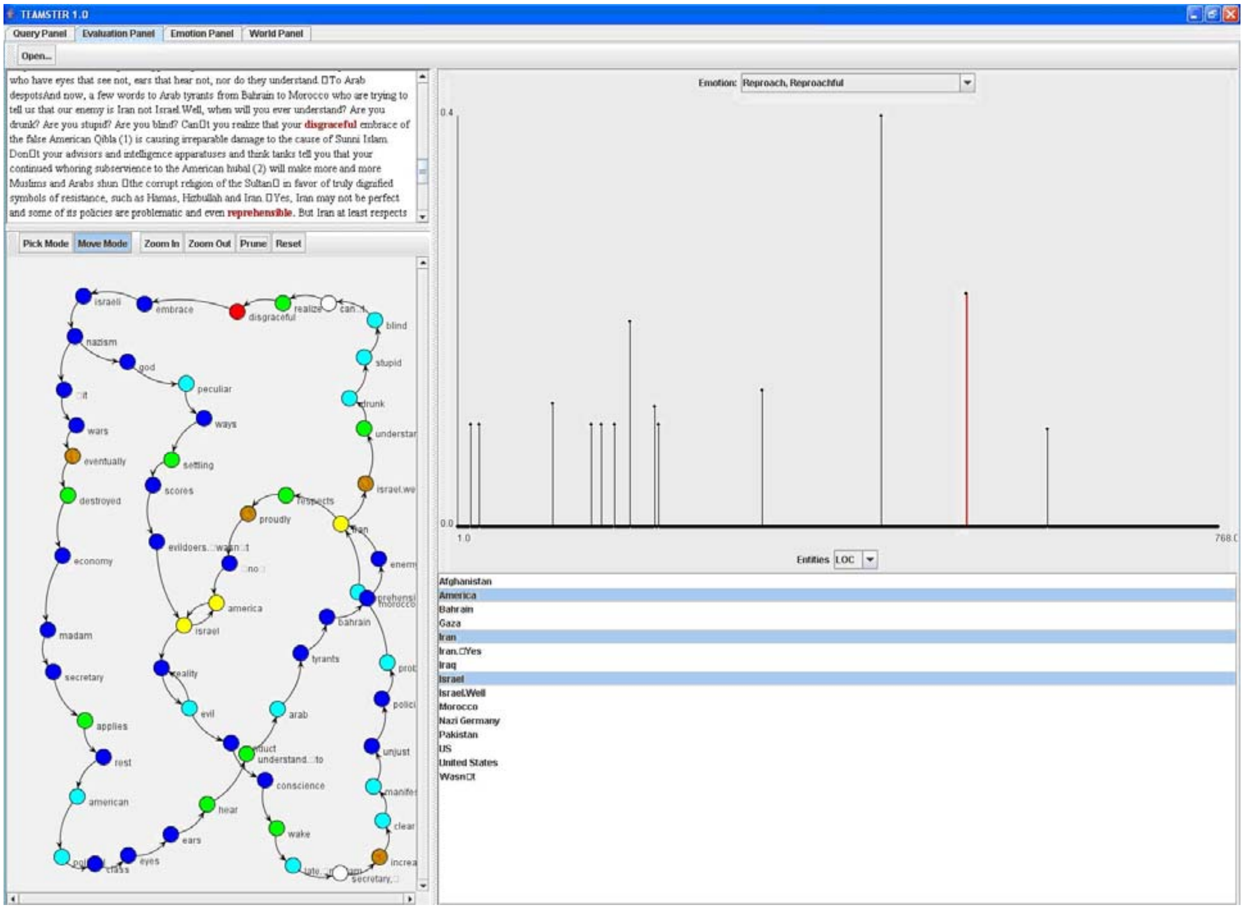
○ Bidirectional Graph

Max Hops: 1

PR Bias: 0.2

Fig. 2. The emotion panel displays the results of the construction of the affect lexicon.

The second step is an on-line process whereby the evaluation panel displays visualizations of affect in a document according to several different perspectives. The panel is partitioned into four quadrants as displayed in Fig. 3: (1) scrolling text window; (2) document word series chart; (3) extracted entity list; and (4) document graph. Words from the affect lexicon are highlighted in red in the scrolling text (upper left-hand corner), and displayed in sequential order in the word series chart. The length of the dropline in the word series chart (upper right-hand corner) is determined by its normalized PageRank value. The word series chart provides a sense of the flow of intensity with respect to a particular affect category throughout the document. Selecting a word in the scrolling text by double-clicking highlights that word in both the word series chart and document graphs. The document graph (lower left-hand corner) contains all words in the document with stop words removed and is rendered by Jung, an open-source Java drawing library. Each node is color-coded by its part-of-speech tag. Named entity extraction, the results of which are shown in the lower right-hand panel, is accomplished using the open-source LBJ named entity tagger from the University of Illinois.



who have eyes that see not, ears that hear not, nor do they understand. □To Arab despotsAnd now, a few words to Arab tyrants from Bahrain to Morocco who are trying to tell us that our enemy is Iran not Israel.Well, when will you ever understand? Are you drunk? Are you stupid? Are you blind? Can□t you realize that your **disgraceful** embrace of the false American Qibla (1) is causing irreparable damage to the cause of Sunni Islam. Don□t your advisors and intelligence apparatuses and think tanks tell you that your continued whoring subservience to the American hubal (2) will make more and more Muslims and Arabs shun □the corrupt religion of the Sultan□ in favor of truly dignified symbols of resistance, such as Hamas, Hizbullah and Iran. □Yes, Iran may not be perfect and some of its policies are problematic and even **reprehensible**. But Iran at least respects

Emotion: Reproach, Reproachful

Entities: LOC

Afghanistan
America
Bahrain
Gaza
Iran
Iran.□Yes
Iraq
Israel
Israel.Well
Morocco
Nazi Germany
Pakistan
US
United States
Wasn□t

Fig. 3. The evaluation panel visualizes affect in single documents.

An important function of the evaluation panel is to support development of analyst understanding of the multiplicity of affective meaning typical of real documents. The user can visualize affect-entity relationships without clutter by pruning the document graph to show only selected affect and extracted entities along with their shortest connecting paths. The shortest paths provide the user with a sense of local context by revealing direct connections between selected terms.

The third step executed by TEAMSTER is propagation of affect to entities and the display of affective relationships in the Affect Analysis Panel. Figure 4 shows a diagram of affective relationships extracted from an excerpt of a document downloaded from the official Hamas website, dated 4-13-2009. This document is saturated with affect in rather plain language relatively unmasked by diplomatic protocol.

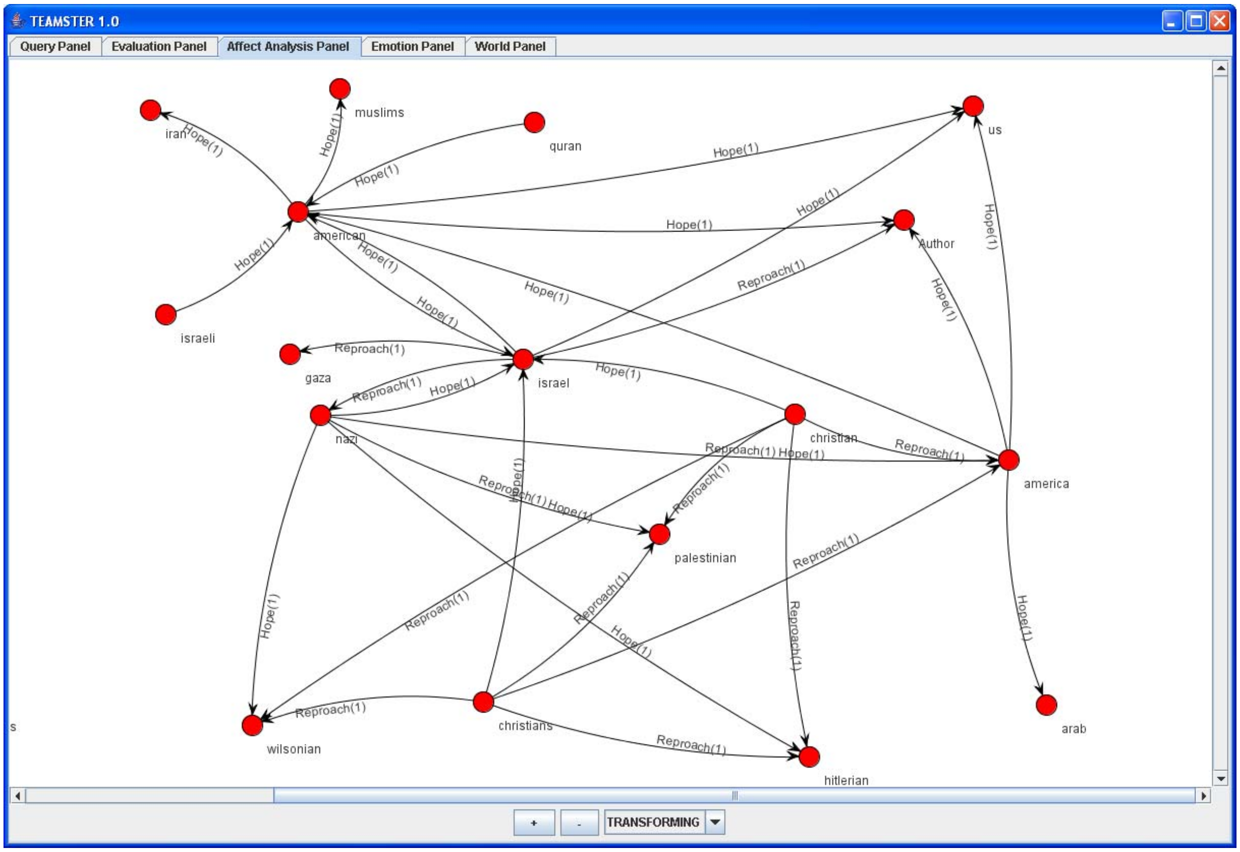


Fig. 4. An affect-entity relationship diagram from a Hamas document displayed in affect analysis panel.

3.2.2 Document-Level Summarization of Affective Content

The document-level summarization metrics are defined in Sect. 4. Briefly, these metrics accumulate information about each affective relationship in a document. The metrics are calculated by the affect extraction algorithm, but are not displayed on the graphical user interface. However, the metrics are presented for all entity pairs in the text output of the prototype application. For example, the direct affective relationship between America and Israel as characterized in the Hamas document is displayed as below (with corresponding definitions or equation numbers from Sect. 4 given in parentheses):

```
Entity 1 :: america
Entity 2 :: israel
rPositive :: 0.6529942057256105          (4.4a)
rNegative :: 0.0                          (4.4b)
rCombined :: 0.32649710286280526         (4.4c)
```

The textual output reveals an overall positive feeling on the part of America for Israel and a lack of American negative affect toward Israel (as described by Hamas). Therefore, the combined affect is on the positive side. The term "cosSum", taken from Eq. (4.1), is an intermediate result in the calculation of affective similarity. It refers to the cosine of a pair of (vector) relationships between two selected entities and a third entity that is subsequently summed over all other entities in the document. The affective similarity between America and Israel according to the sample document is shown in the output as:

```
Entity 1 :: america
Entity 2 :: israel
```

```
cosSum :: 25.819990956308914
numOfEntities :: 35.0
Affective Similarity :: 0.782423968372        (4.1)
```

which indicates a relatively high degree of similarity. Finally the text output reports two subjectivity metrics:

```
Subjectivity 1 :: 0.20588235294117646
Subjectivity 2 :: 0.23024785661018204
```

The subjectivity metrics assess the degree to which the author is engaged in non-zero affective relationships with other entities extracted from a document. Objective texts should possess relatively little evidence of affective relationships between the author and entities described in the text. The apparent lack of affective entanglement is typical of the preferred style of objective news reporting.

3.3 NOTATION USED IN DEVELOPMENT OF AFFECT PROPAGATION ALGORITHM

Let $A=\{A_1,\ldots,A_m\}$ be a set of affects. We adopted a well-established taxonomy from O'Rorke and Ortony (1994) that discriminates 22 affect categories, but in theory any taxonomy that associates affect terms with entities could be used. To each affect category, we may associate a collection of terms so that: $A_i=\{A_{i1},\ldots,A_{in}\}$ for $i=1,\ldots,n$. These are the terms that induce the affect in the chosen lexicon (English in this report). As sets, the affects are not mutually exclusive (i.e., we may have $i, j\in\{1,\ldots,m\}$ so that $A_i\cap A_j\neq\emptyset$). This occurs when an affect term is a member of multiple affect categories.

We partition *A* into binary and unary affects as: $A=A^b\cup A^u$. In our affect taxonomy there are 11 natural pairs of affects; each member of the pair contains a positive and negative member, e.g., love and hate. We define another partition of the affect category set *A* into positive affects A^+ and negative affects A^-. In our scheme, positive and negative affects are symmetric, thus $\left|A^+\right|=\left|A^-\right|=\frac{m}{2}$.

Let **A** be the set of all affect terms in a text. Further, define Also define **N** to be the set of terms indicating negation, **Q** the set of terms indicating quantification and **C** the set indicating conjunction. Lastly, let **E** be the set of entity terms, which is specified by an external recognition process. Unary affect tokens are always linked to a single source entity in **E.** The source is the entity experiencing the affect. Binary affect tokens are also linked to a source entity, but additionally, they are also joined to a target entity. The target entity is the recipient of the affect.

There always exists a distinguished entity (e_1) – author – in **E** that is referenced by personal pronouns {I, we, me, us}, or implicitly referenced (e.g., when there is no apparent explicit reference to either a source entity or a target entity belonging to a binary relation affect token.)

The sample text contains tokens from each word category described above. We have **E=**{Israel, U.S., Iran, Egypt}, **A=**{insecurity, support, liking, values, animosity, fear, approve, respect}, **N=**{not}, **Q** = {strong, deep, moderate} and **C=**{just as, and, while}. Some of these terms have multiple instances or tokens in the text.

We adopt the graph notation introduced by Cohen (2010) to define paths through a document. Let a (directed) graph $G=(V,E)$ of the word tokens in a document be a set of *vertices V* and a set of *weighted edges* $E=V\times V\times W$, where a weighted edge (v_i,v_j,w_{ij}) is a directed link from v_i to v_j with weight w_{ij}. We'll assume that edge weights lie in the closed interval [0,1]. Since *V* contains all (except some stop words) the word tokens in a document $V\supset\mathbf{E}\cup\mathbf{A}\cup\mathbf{Q}\cup\mathbf{N}\cup\mathbf{C}$. A path through a graph G is a sequence of triples $p = \langle (v_0, v_1, w_{01}), \ldots (v_{T-1}, v_T, w_{T-1,T}) \rangle$ such that every triple is an edge in *E*. The set of all paths from v_i to v_j is written $PATHS(G, v_i, v_j)$ or simply $PATHS(v_i, v_j)$when *G* is known. Also, let $\mathbf{G}$ be the adjacency matrix for graph *G* whose (i, j) is the weight of an edge from vertex *i* to vertex *j*.

Consider the following clause from the sample text "… Israel values its friendship with the U.S." This sentence is expressed in canonical "Noun Phrase-Verb Phrase" form, which allows the affect verb 'values' to follow the natural flow of the text to spread activation to and connect with the target entity 'U.S.' in the VP. The affect word 'values' must also propagate and link to the source entity 'Israel' in the NP, but the direction of activation is against the natural text ordering. The transpose of $\mathbf{G}$ (denoted $\mathbf{G}^T$) effectively reverses the directional flow of the words represented by graph *G*, which turns out to be useful in joining affect tokens to source/target entities that appear prior to the affect token in the text, and therefore against the flow induced by the natural ordering of the text. Forward and reverse directions of flow are required to join binary affect terms to entity pairs, but unary affect terms can be placed either prior to or following the modified entity in the natural text ordering. Since we have no a priori reason to suspect that either case applies, a propagation that is capable of proceeding in both forward and reverse directions is preferable when propagating activation from a unary affect to the associated entity. Using both matrices $\mathbf{G}$ and $\mathbf{G}^T$ in succession simulates a bidirectional flow of spreading activation from affect tokens to entities in the text graph. These three operations: forward flow, reverse flow, and bidirectional flow are the main methods required to spread the activation of flow from saturated affect tokens in a document to source and target entities.

We expect that local contextual modifications of affective meaning in a document occur within the boundaries of a local neighborhood. Let $L_H(x')$ be the *H*-hop neighborhood of vertex (term) $x' \in V$. That is, the set of all vertices *y* that can be reached from x' by following a path of at most *H* triples.

3.4 CONSTRUCTION OF THE AFFECT LEXICON

The initial offline step of the affect extraction algorithm is the *construction of an affect lexicon* for each affect category, (i.e., identification of terms in English saturated with affect, feeling, or emotion). This step is an off-line process that produces a set of affect vectors that become the search terms for documents. Figure 1 shows that the affect lexicon is visualized in the emotion panel as both a list of ranks (right side) and a graph (left side). PageRank is well-known as the theoretical foundation of many search engines (e.g., Google) in the internet webspace. A set of hyperlinked webpages can be formally represented as a graph where pages are nodes and hyperlinks are the directed arcs of the graph. PageRank is an algorithm that attaches a score denoting a degree of page authority to a website based only on the topology of web connectivity (Bianchini 2005).

PageRank is adapted here to the task of constructing an affect lexicon by capitalizing on the fact that lexical resources like WordNet and ConceptNet are graph structures. A random walk on these graphs can be performed in a manner similar to that described by Esuli and Sebastiani (2007). Starting with seed lists of words saturated with specific affects, a graph walk onto WordNet is performed to locate additional affect terms. Their intensities are ranked by PageRank while stepping through a word association graph (synsets) instead of a hyperlinked document graph. A similar graph walk can be performed on the ConceptNet database.

3.5 EXTRACTION OF ENTITIES AND AFFECT TERMS

The extraction of specific terms from a text depends upon an extensive preprocessing phase during which several natural language processing tasks (NLP) are carried out. These tasks include tokenization, word stemming, part-of-speech tagging, sentence detection, pruning and document graph construction. We rely on the Java OpenNLP Tools library (see References for URL address) to perform most of these tasks. After tokenization and other NLP tasks have completed, the affect and entity terms are identified by search within the graph structure of the underlying text. Entity extraction is exogenous to our affect extraction algorithm, and we are currently using the open-source Illinois LBJ named entity tagger to identify entities in documents (Ratinov and Roth 2009). This software tags entities in a text as a person, location, organization or miscellaneous category. In the future we plan to expand entity extraction to capture special classes of abstract nouns (e.g., socialism) and concrete nouns (e.g., steak), because these types of entities can serve as targets of affective meaning. We introduce a function, $f : \mathbf{A} \to U_+$, where $U_+ = [0,1]$ that returns the intensity or score of each affect token for affect category A_j obtained from an affect lexicon. We use a normalized PageRank function (*npr*) as the specific form of the affect intensity function *f* here.

3.6 CONTEXTUAL EFFECTS OF AFFECT TERMS

After the extraction of all affect and entity tokens, a number of common English words from a “stop list” must be removed or pruned from the document. These words do not add significantly to the affective meaning of a text. The graph transformation follows the natural ordering of the text, possibly including some aggregation of terms that repeat in multiple locations.

The meaning or intensity of affect terms are perturbed or modified by the other words appearing in the local context of the document graph. Hence it is important to adjust the affect intensities predicted by PageRank in the off-line step to account for negation, qualifiers, conditionals, and other subordinating conjunctions such as *although* and *while*. Conditionals are not addressed here as we initially focus on negation and qualification.

The proper treatment of negation is a basic topic in text analysis (e.g., Harabagiu et al. 2006). Let the partially ordered set of negation tokens $\{n_{ijk} : n_{ijk} \in L_H(a_{ij}) \cap \mathbf{N}\}$ of cardinality m_{ij} be the negation tokens (some of which may refer to identical negation words, e.g., *not*) qualifying affect token a_{ij}. Similarly, the partially ordered set of qualifier tokens $\{q_{ijk} : q_{ijk} \in L_H(a_{ij}) \cap \mathbf{Q}\}$ of cardinality p_{ij} are the qualifier words modifying a_{ij} in a document of interest. The token a_{ij} in a text corresponds to the affect term A_{ij} in the lexicon. The negation function processes the effects of nested negation tokens, i.e., double negation, triple negation, etc., and the qualifier function adjusts affect intensities for a series of qualifier words associated with an affect token. The unit ranges $U_+ = [0,1]$, $U_- = [-1,0]$ and $U_\pm = [-1,1]$ will be used as shorthand notations in what follows. We introduce a function $k : \mathbf{Q} \to U_\pm$ that returns a previously determined mapping from qualifier terms into a real-valued intensity (see Appendix B for details.) The negation and qualifier functions operate on affect token intensities and are defined below. We use recursion parameter *x* in the qualifier function to make the iterative or cumulative nature of the function more formal and explicit.

$$neg\left(f(a_{ij}), m_{ij}\right) = [-1]^{m_{ij}} \cdot f(a_{ij}) \tag{3.1}$$

$$qual\left(0, f(a_{ij}), k(q_{ij,0})\right) = f(a_{ij})$$

$$qual\left(x+1, f(a_{ij}), k(q_{ij,x+1})\right) = \begin{cases} 0 & f(a_{ij}) = 0 \\ qual\left(x, f(a_{ij}), k(q_{ij,x})\right) + k(q_{ij,x+1}) \cdot qual\left(x, f(a_{ij}), k(q_{ij,x})\right) & -1 \le \mathrm{k}(\mathrm{q}_{\mathrm{ij,x+1}}) < 0 \\ qual\left(x, f(a_{ij}), k(q_{ij,x})\right) + k(q_{ij,x+1}) \cdot \left[1 - qual\left(x, f(a_{ij}), k(q_{ij,x})\right)\right] & 0 \le \mathrm{k}(\mathrm{q}_{\mathrm{ij,x+1}}) \le 1 \end{cases} \tag{3.2}$$

The recursion parameter is $0 \le x \le p_{ij} - 1$. The qualifier function $qual : \mathbf{N}_0 \times U_+ \times U_\pm \to U_+$ is recursively defined whereas the negation function $neg : U_+ \times \mathbf{N}_0 \to U_\pm$ is not recursive. The output of *k* is a score that represents the strength of the qualifier when modifying an affect term. For $-1 < k(q) < 0$, *q* is a reducer (decreases affect intensity), and for $0 < k(q) \le 1$, *q* is an enhancer (increases affect intensity). The values assigned by *k* are pre-determined by an external ranking procedure (see Appendix B). It should be clear that affects a_{ij} are tokens and not just words or phrases in the affect lexicon. The important distinction is that there may be multiple tokens representing the same affect term in a given document.

The affect intensity, $I : U_+ \times \mathbf{N}_0 \times \mathbf{N}_0 \to U_\pm$, is the strength of a particular affect token in a local context of a document after accounting for negation and qualification.

$$I(a_{ij}, p_{ij}, m_{ij}) = neg(qual(p_{ij}, f(a_{ij})), m_{ij}) \tag{3.3}$$

where *f* refers to the normalized PageRank (*npr*) of affect term a_{ij} for category A_j from the affect lexicon. The ranges of both *qual*() and *neg*() are in the interval $\mathbf{U}_{-+}$, which guarantees that the range of *I* is in $U_\pm$. In case affect intensity is positive (i.e., $neg(f(a_{ij})) > 0$), then the range of *I* is restricted to U_+; otherwise the range of *I* is restricted to U_-. Substituting Eq. (3.1) into Eq. (3.3) we have:

$$I(a_{ij}, p_{ij}, m_{ij}) = \begin{cases} -qual(p_{ij}, f(a_{ij}), m_{ij}) & \text{if } m_{ij} \text{ is an odd integer} \\ qual(p_{ij}, f(a_{ij}), m_{ij}) & \text{otherwise} \end{cases} \tag{3.4}$$

3.7 PROPAGATION OF AFFECTIVE MEANING FROM AFFECT TERMS TO SOURCE AND TARGET ENTITIES

The extraction of affects and entities from a document is a prerequisite for the final step of single document analysis, namely the identification of non-zero affective relationships among entities. Many documents of potential interest to analysts – blogs, emails, political forum exchanges, and chat dialogues – have complex structures that often introduce multiple entities and topics. Affective relationships among some of those entities according to the author's perspective are usually encoded within the denotative or connotative semantics. The text mining task is to appropriately spread the activation of affective meaning from tokens in the affect lexicon to neighboring entities in a document. This on-line task is at least superficially similar to the type of spreading activation employed off-line to construct the affect lexicon, suggesting that a similar PageRank-like graph walking algorithm may be appropriate. PageRank applies here because we are working with graphical representations of documents. We selected a class of random walk with restart (*rwr*) algorithms that step through the graph space connecting affect terms to source and target entities.

We argue for a spreading activation model of the cognitive process of linking saturated affect terms to specific entities in the document. The concept of spreading activation has a long history in models of cognition (Anderson 1983), and has been implicated in more recent models of human text processing (Lemaire 2006, Crestani 1997, Graesser 1997). Our approach is founded on the notion that affective meaning in documents flows from saturated affect terms to significant neighbor nodes, which can be conceptualized as sinks. The spread of activation must be properly regulated in order for the spreading affect to reach their intended sources and targets. Flow must proceed in the correct direction and discouraged from following pathways in the document graph that are not productive.

In general flow proceeds in the same direction as the natural ordering of the text (forward) or in a reverse direction (reverse). In the case of unary affects, a bidirectional flow may be required to diffuse affective meaning to a linked entity. For example, “Mary was dejected” follows a reverse flow from affect term to source entity, and “The dejected Mary went home” depends on a forward affective flow to link affect and source entity. Each type of directional flow is associated with a specific adjacency matrix. There are three cases:

Case	Flow	Matrix
Binary (affect token to target entity)	forward	$\mathbf{G}$
Binary (affect token to source entity)	reverse	$\mathbf{G}^T$
Unary	forward, reverse	$\mathbf{G},\mathbf{G}^T$

3.7.1 Edge Weight Modification

Flow regulation is determined during a preliminary step that involves the adjustment of the edge weights in the adjacency matrices $\mathbf{G}$ and $\mathbf{G}^T$ to account for local context. A typical adjacency matrix contains only 0’s and 1’s indicating absence or presence of a directional link between two nodes in the document graph. However, each affect token induces a particular perspective on the graph semantics, and requires a unique approach to flow regulation in $\mathbf{G}$ and $\mathbf{G}^T$. Now edge weights are potentially allowed to have any value in U_+. The lower the value of the edge weight, the more flow of activation is inhibited from continuing in that direction. The flow algorithm searches for specific word types to impede flow, such as entities, affects and punctuation. First, we search for specific word types (affect, entities) in $\mathbf{G}$ and $\mathbf{G}^T$ represented by the set of edges E in G. The z_i [in Eqs. (3.5) – (3.6)] are constants or parameters in U_+ that modify edge weights according to word type.

$$w_{ij} = \mathbf{G}'[v_i, v_j] = \begin{cases} \mathbf{G}[v_i, v_j] \times z_a & if\, v_j \in V(G) \\ \mathbf{G}[v_i, v_j] \times z_e & if\, v_i \in V(G) \\ \mathbf{G}[v_i, v_j] & otherwise \end{cases} \tag{3.5}$$

In Eq. (3.5) $\mathbf{G}'$ is just the modified $\mathbf{G}$ matrix after accounting for affect and entity terms. Henceforth we refer to $\mathbf{G}'$ simply as $\mathbf{G}$. Next the algorithm accounts for the blocking effects of punctuation. However, the tokenizer omits punctuation from the document graph G, so we cannot find punctuation symbols in the nodes of $\mathbf{G}$. Instead, punctuation symbols are associated with the document graph edges. Let $P : E \rightarrow \mathbf{P}$ represent the mapping of an edge to its punctuation symbol where **P** = {period, ?, ! , comma, doublequote, empty}; if there is no mapping to an explicit punctuation symbol, the edge value is empty.

$$w_{ij} = \mathbf{G}'[v_i, v_j] = \begin{cases} \mathbf{G}[v_i, v_j] \times z_p & if\, P(d_{ij}) = "period"\, or\, "?"\, \text{or}\, "!" \\ \mathbf{G}[v_i, v_j] \times z_c & if\, P(d_{ij}) = "comma" \\ \mathbf{G}[v_i, v_j] \times z_q & if\, P(d_{ij}) = "doublequote" \\ \mathbf{G}[v_i, v_j] & otherwise \end{cases} \tag{3.6}$$

Equations (3.5) and (3.6) complete the modifications to the edge weights. The final step is to row-normalize $\mathbf{G}$ and $\mathbf{G}^T$ such that the transition probabilities from each node sum to unity. This step yields the normalized weight matrices $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{G}}^T$. Normalization is required as a pre-processing step for the Random Walk with Restart algorithm. This step is somewhat complex compared to normalizing a matrix with only 1’s and 0’s. Appendix A outlines the procedure we use to accomplish the normalization.

3.7.2 Online Affect Propagation

The main part of the affect propagation phase of the algorithm uses a variant of PageRank (random walk with restart) to accomplish spreading activation from affect tokens to entities. Random walk with restart (*rwr*) is executed twice centering on each affect term in the document, once for forward and once for reverse flow. The function $rwr : \mathbf{A} \times \mathbf{E} \times \left\{ \tilde{\mathbf{G}}, \tilde{\mathbf{G}}^{\mathsf{T}} \right\} \rightarrow U_{+}$ estimates the degree of association (relevance) of an entity with an affect token. The *rwr* algorithm can be considered as a special case of personalized PageRank. An energized start node, for example – an affect token, transmits activation to its neighborhood with probability proportional to the edge weights. At each step there exists probability α of returning to the start node, which propagates flow from an affect term. Thus, successive random walks through *G* are enacted from the start node. The (forward) relevance score of node *j* with respect to start node *i* (for binary affects) is defined as the steady state probability $rwr\left(v_i, v_j, \tilde{\mathbf{G}}\right)$ that node *j* will be visited during the random walk cycles using the forward normalized weight matrix $\tilde{\mathbf{G}}$. Consider the vector $\vec{w}_i = \left\langle rwr\left(v_i, v_1, \tilde{\mathbf{G}}\right), rwr\left(v_i, v_2, \tilde{\mathbf{G}}\right), \ldots rwr\left(v_i, v_l, \tilde{\mathbf{G}}\right) \right\rangle$ containing the forward *rwr* scores for all entities with respect to the i^{th} affect term. The vector $\vec{w}_i$ is determined by solving a linear system problem as shown in Eq. (3.7):

$$\vec{w}_i = \alpha \cdot \tilde{\mathbf{G}} \cdot \vec{w}_i + \left(1 - \alpha\right) \cdot \vec{h}_i \tag{3.7}$$

The symbol $\vec{h}_i$ in Eq. (3.7) is the $n \times 1$ start vector of zeros and ones. The start node in this vector (affect token) is assigned a scalar value of 1, and all other values in the vector are 0. Equation (3.7) permits *rwr* to be evaluated deterministically (see Tong et al. 2008). Where reverse flow is required instead of forward flow $\tilde{\mathbf{G}}^T$ is used in place of $\tilde{\mathbf{G}}$ in Eq. (3.7). The main difference between PageRank and *rwr* is that random jumps in the latter always return the random walk pointer to a subset of nodes in the document graph. In this case the unique start node is an affect token.

The "author" entity requires special processing because when used as a default entity it receives no output from *rwr*. Ordinarily, the *rwr* rank for author associated with affect a_{ij} will be calculated using a member of the set of personal pronouns **Z** = {I, we, me, us}. Where there are no local tokens from **Z** in the neighborhood of a_{ij} to represent the author, the "ghost" author rank associated with binary affect token a_{ij} is given as:

$$rwr\left(e_1, a_{ij}, \tilde{\mathbf{G}}\right) = \begin{cases} rwr\left(e_k, a_{ij}, \tilde{\mathbf{G}}\right) & \text{if } \exists \mathrm{e}_{\mathrm{k}} \in \mathbf{Z} \text{ s.t. } rwr\left(e_k, a_{ij}, \tilde{\mathbf{G}}\right) > t_a \\ \beta & \text{if } \neg\exists \mathrm{e}_{\mathrm{k}} \in \mathbf{E} \text{ s.t. } rwr\left(e_k, a_{ij}, \tilde{\mathbf{G}}\right) > t_a \\ 0 & \textit{otherwise} \end{cases} \tag{3.8}$$

where $0 \leq t_a < 1$. The directionality of text processing in *rwr* is either reverse or forward depending upon whether the author (e_1) is a source or target entity. Forward flow with $\tilde{\mathbf{G}}$ is used when the author is a target entity as shown in Eq. (3.8). However, when the author is the source entity $\tilde{\mathbf{G}}^T$ is substituted for $\widetilde{\mathbf{G}}$ in (3.8) to reverse affective flow. Author entity processing for unary affect terms requires the maximum rank obtained from forward and reverse propagation, or $\max\left(rwr\left(e_k, a_{ij}, \widetilde{\mathbf{G}}\right), rwr\left(e_k, a_{ij}, \widetilde{\mathbf{G}}^{\mathsf{T}}\right)\right)$.

The parameter β in Eq. (3.8), whose range is in U_+, is a constant rank assignment for so-called “ghost” entities when no other source or target entity achieves a threshold rank with respect to the value of t_a. In case no members of **Z** are activated and other entities are activated by an affect token, then a ghost entity will not be inserted. If any author pronouns are sufficiently activated as reflected in the *rwr* score, then that score is used instead of invoking a “ghost” author entity. Finally, if neither of the two above conditions is satisfied, an author entity is not associated with affect term a_{ij}.

We now return to the discussion of processing of affect terms in general. The affect-entity relationship function computes the overall strength of the relationship between an affect token and entity pair $EA_b : \mathbf{E} \times \mathbf{E} \times \mathbf{A} \rightarrow U_{\pm}$ or affect token and source entity $EA_u : \mathbf{E} \times \mathbf{A} \rightarrow U_{\pm}$ as shown in Eq. (3.9a) for binary affect categories and Eq. (3.9b) for unary affects.

$$EA_b\left(e_i, e_k, a_{jl}\right) = pown\left[I\left(a_{jl}, p_{jl}, m_{jl}\right) * \left(rwr\left(e_i, a_{jl}, \widetilde{\mathbf{G}}\right) \cdot rwr\left(e_k, a_{jl}, \widetilde{\mathbf{G}}^{\mathsf{T}}\right)\right)^{1/2}, \gamma\right] \tag{3.9a}$$

$$EA_u\left(e_i, a_{jl}\right) = pown\left[I\left(a_{jl}, p_{jl}, m_{jl}\right) * \max\left(rwr\left(e_i, a_{jl}, \widetilde{\mathbf{G}}\right), rwr\left(e_i, a_{jl}, \widetilde{\mathbf{G}}^{\mathsf{T}}\right)\right), \gamma\right] \tag{3.9b}$$

where $0 \leq \gamma < 1$ is a scaling parameter intended to help distribute combined ranks more evenly over the (0,1) range. The negative power function $pown(x, a) = \operatorname{sgn}(x)x^a$ is an extension of the standard power function to ensure proper treatment of negative numbers. Note from Eq. (3.9a) and Eq. (3.9b) that *rwr* is executed twice, once for each affect category (using $\widetilde{\mathbf{G}}$ for forward and $\widetilde{\mathbf{G}}^T$ for reverse). Equation (3.9a) captures the co-occurrence between two entities by taking the geometric mean as the combination function.

3.7.3 Evidence Aggregation from Multiple Affect Tokens

Many documents contain multiple affect tokens representing each affect category. Therefore, when considering the affective relationship between entities with respect to a category, we must aggregate the evidence from each affect token in that category. From our example, the tokens “values” and “liking” are both exemplars in the affect lexicon from the “love” category. We adopt a model of reasoning under uncertainty with belief functions based on certainty factors to assess the evidence from multiple tokens in a single document. Certainty factors are equivalent to a noisy-or gate in aggregating the effect of multiple bits of evidence on a belief (Lucas 2001). A belief function is similar to a probability but expresses a degree of certainty in a proposition in the face of empirical evidence.

Each entity pair $\left(e_i, e_k\right)$ has associated evidence for affect category A_j consisting of their joint ranks inherited from each member of the set of affect tokens. Some evidence may support the presence of an affective relationship with respect to A_j, whereas other evidence may refute the same. We define a set for each type of evidence: a set for (E_{ikj}^+) and a set against (E_{ikj}^-) the association of entity pair $\left(e_i, e_k\right)$ with binary affect category A_j. Similarly, E_{ij}^+ and E_{ij}^- are positive and negative evidence sets supporting an association between entity e_i and unary affect category A_j. Negative evidence is generated by the presence of negation in the local neighborhood of the affect token, but positive evidence is not influenced by negation. Evidence is only credible enough to be counted in case its *EA* rank is more extreme than threshold values $0 \leq t_b, t_u < 1$. Definitions for these evidence sets are given below:

$$E_{ikj}^{+} = \{a_{.j} : EA_b(e_i, e_k, a_{.j}) \geq t_b\} \qquad \text{(binary)} \qquad (3.10a)$$

$$E_{ikj}^{-} = \{a_{.j} : EA_b(e_i, e_k, a_{.j}) \leq -t_b\} \qquad \text{(binary)} \qquad (3.10b)$$

$$E_{ij}^{+} = \{a_{.j} : EA_u(e_i, a_{.j}) \geq t_u\} \qquad \text{(unary)} \qquad (3.10c)$$

$$E_{ij}^{-} = \{a_{.j} : EA_u(e_i, a_{.j}) \leq -t_u\} \qquad \text{(unary)} \qquad (3.10d)$$

The set of all binary affect evidence for the association of entity pair (e_i, e_k) with category A_j is E_{ikj}. E_{ikj}^{+} and E_{ikj}^{-} form a partition of E_{ikj}. Also, let cardinalities $c_{b+} = |E_{ikj}^{+}|$, $c_{b-} = |E_{ikj}^{-}|$, $c_{u+} = |E_{ij}^{+}|$, and $c_{u-} = |E_{ij}^{-}|$.

The expert system MYCIN (Tsadiras and Margaritis 1998, Qu and Shirai 2003) is an example of a model that uses belief/disbelief functions to aggregate evidence of the same type using certainty factors. In the case of binary affects, we are interested in quantifying the belief that entity pair (e_i, e_k) is associated with affect category A_j. The (dis-)belief functions are defined here using recursion parameters *x,y*. The belief function for positive evidence $B : \mathbf{N}_0 \times \mathbf{E} \times \mathbf{E} \times \mathbf{A} \times U_+$ and disbelief function for negative evidence $D : \mathbf{N}_0 \times \mathbf{E} \times \mathbf{E} \times \mathbf{A} \times U_-$ are defined as follows:

$$B_{ikj}(0, e_i, e_k, a_{j,0} | E_{ikj}^{+}) = 0 \qquad (3.11a)$$

$$D_{ikj}(0, e_i, e_k, a_{j,0} | E_{ikj}^{-}) = 0 \qquad (3.11b)$$

$$B_{ikj}(x+1, e_i, e_k, a_{j,x+1} | E_{ikj}^{+}) = B_{ikj}(x, e_i, e_k, a_{j,x} | E_{ikj}^{+}) + EA_b(e_i, e_k, a_{j,x+1}) \cdot [1 - B_{ikj}(x, e_i, e_k, a_{j,x} | E_{ikj}^{+})] \qquad (3.11c)$$

$$D_{ikj}(y+1, e_i, e_k, a_{j,y+1} | E_{ikj}^{-}) = D_{ikj}(y, e_i, e_k, a_{j,y} | E_{ikj}^{-}) + EA_b(e_i, e_k, a_{j,y+1}) \cdot [1 + D_{ikj}(y, e_i, e_k, a_{j,y} | E_{ikj}^{-})] \qquad (3.11d)$$

For $x = 0, \ldots, c_{b+} - 1$ and $y = 0, \ldots, c_{b-} - 1$, and where $a_{j,0}$ is a null argument. It is easy to verify that if the range of the EA_b function is $\mathbf{U}_+$, then the range of the belief function (*B*) is also $\mathbf{U}_+$. A similar argument can be formulated for the disbelief function (*D*). For all unary affect categories we use evidence sets E_{ij}^{+} and E_{ij}^{-}.

$$B_{ikj}(0, e_i, a_{j,0} | E_{ij}^{+}) = 0 \qquad (3.11e)$$

$$D_{ikj}(0, e_i, a_{j,0} | E_{ij}^{-}) = 0 \qquad (3.11f)$$

$$B_{ij}(x+1, e_i, a_{j,x+1} | E_{ij}^{+}) = B_{ij}(x, e_i, a_{j,x} | E_{ij}^{+}) + EA_u(e_i, a_{j,x+1}) \cdot [1 - B_{ij}(x, e_i, a_{j,x} | E_{ij}^{+})] \qquad (3.11g)$$

$$D_{ij}(y+1, e_i, a_{j,y+1} | E_{ij}^{-}) = D_{ij}(y, e_i, a_{j,y} | E_{ij}^{-}) + EA_u(e_i, a_{j,y+1}) \cdot [1 + D_{ij}(y, e_i, a_{j,y} | E_{ij}^{-})] \qquad (3.11h)$$

where $x = 0, \ldots, c_{u+} - 1$ and $y = 0, \ldots, c_{u-} - 1$. Intuitively, belief/disbelief increases/decreases proportionally with each independent piece of supporting evidence obtained from analyzing an affect token.

The combination of multiple evidence is obtained by chaining these rules in iterative fashion until all positive evidence is aggregated using *B*, and all negative evidence is aggregated using *D*. The order of considering the evidence does not matter, although taking tokens in their natural order in the document is a reasonable tactic. Finally, conflicting evidence is aggregated from the total evidence for (*B*) and against (*D*) the presence of an affective relationship for a given entity pair. Let $c_b = \left(c_{b+}, c_{b-}\right)$ denote the pair of cardinalities for the binary affect evidence sets, and let $A'_b = \left(a'_{b+}, a'_{b-}\right)$ denote the last pair of affect terms in both binary affect evidence sets. The affective relationship between entity pair $\left(e_i, e_k\right)$ and category A_j is represented by the function $r_{ikj} : \mathbf{E} \times \mathbf{E} \times \mathbf{N} \times \mathbf{N} \times E^{+}_{ikj} \times E^{-}_{ijk} \rightarrow U_{\pm}$ and defined as:

$$r_{ikj}\left(e_i, e_k, c_b, A'_b \middle| E^{+}_{ikj}, E^{-}_{ikj}\right) = \frac{B_{ikj}\left(c_{b+}, e_i, e_k, a'_{b+} \middle| E^{+}_{ikj}\right) - abs\left(D_{ikj}\left(c_{b-}, e_i, e_k, a'_{b-} \middle| E^{-}_{ikj}\right)\right)}{1 - \min\left(B_{ikj}\left(c_{b+}, e_i, e_k, a'_{b+} \middle| E^{+}_{ikj}\right), abs\left(D_{ikj}\left(c_{b-}, e_i, e_k, a'_{b-} \middle| E^{-}_{ikj}\right)\right)\right)} \tag{3.12a}$$

Now let $c_u = \left(c_{u+}, c_{u-}\right)$ denote the pair of cardinalities for the unary affect evidence sets, and let $A'_u = \left(a'_{u+}, a'_{u-}\right)$ be the last pair of tokens in both unary affect evidence sets. For entity e_i and category A_j, the unary affective relationship function $r_{ij} = \mathbf{N} \times \mathbf{N} \times E^{+}_{ij} \times E^{-}_{ij} \rightarrow U_{\pm}$ is defined as:

$$r_{ij}\left(e_i, c_u, A'_u \middle| E^{+}_{ij}, E^{-}_{ij}\right) = \frac{B_{ij}\left(c_{u+}, e_i, a'_{u+} \middle| E^{+}_{ij}\right) - abs\left(D_{ij}\left(c_{u-}, e_i, a'_{u-} \middle| E^{-}_{ij}\right)\right)}{1 - \min\left(B_{ij}\left(c_{u+}, e_i, a'_{u+} \middle| E^{+}_{ij}\right), abs\left(D_{ij}\left(c_{u-}, e_i, a'_{u-} \middle| E^{-}_{ij}\right)\right)\right)} \tag{3.12b}$$

which is equivalent to the rule for combining conflicting evidence in the EMYCIN expert system shell. The outputs of functions r_{ikj} and r_{ij} are actually single elements of a vector representing the total (binary/unary) affect relationship profile for an ordered entity pair where each component of the vector denotes the total strength of the evidence for a unique affect category. Please note that r_{ikj} is not symmetric (i.e., in general $r_{ikj} \neq r_{kij}$). This lack of affective symmetry is illustrated in the sample text, where Egypt is said to like Israel, but no reciprocal liking on the part of Israel is directly stated.

For the sake of simplicity we refer to the binary affect vector between two ordered entities $\left(e_i, e_k\right)$ as $\mathbf{r}_{ik} = \left\langle r_{ik1}, r_{ik2}, \ldots, r_{ikm}\right\rangle$. We can collect $\mathbf{r}_{\text{ik}}$ for all entity pairs $\left(e_i, e_k\right)$ to construct an affect-entity relationship matrix (for binary affect categories). This matrix contains all the (binary) affective information in a document. Similarly, the unary affect-entity relationship matrix is constructed using unary affect row vectors $\mathbf{r}_{\text{i}}$ for all entities e_{i}. Both matrices could be combined in a single matrix. Alternatively, we can write the collection of (binary) affective relationships as a vector set: $\mathbf{T}_b = \left\{\mathbf{r}_{12}, \mathbf{r}_{13}, \ldots, \mathbf{r}_{21}, \mathbf{r}_{23}, \ldots, \mathbf{r}_{(n-1)n}\right\}$. The collection of unary affective relationships is defined as $\mathbf{T}_u = \left\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_n\right\}$.

We may also consider the collection of affective relationships in a document as a network diagram or graph $AEDR = \left(\mathbf{E}, T\right)$ where the nodes are the set of named document entities **E**, and a labeled set of edges $T = \mathbf{E} \times \mathbf{E} \times W_{\mathbf{E}}$ where the weights, $W_{\mathbf{E}}$, are (affect) vectors rather than simple scalar values. Only those affect – entity relations or weights that have sufficient strength (i.e., above a threshold value) should be represented in the network diagram. Entities are represented as nodes, and affect categories are represented as labeled edges in the diagram. Unary affect relations are represented with self-referential loops to and from a single node. Figure 5 illustrates an affect-entity relationship diagram for the sample text.

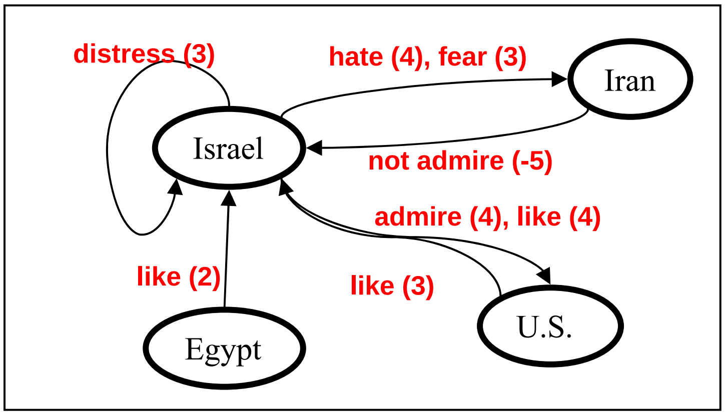


Fig. 5. Affect-entity relationship diagram for Middle Eastern political actors.

We think of the affect-entity relationship diagram as the underlying affect schema of the author of a document. This kind of visualization of affective relationships at the graphical user interface, which is implemented in the TEAMSTER prototype, provides users and analysts with a holistic apprehension of the affective or social perceptions of the document author.

4. SUMMARY METRICS FOR AFFECTIVE RELATIONSHIPS

The previous section described an algorithm to derive detailed affective meaning at sub-document levels from texts. While this detailed information is important to parsing connotative meaning, it is also important to be able to aggregate data in order to provide a high level view of affect in a document. Affective relationships among entities in a document can be aggregated to produce document-level scores related to the use of affect in the document. Ultimately this type of aggregation will allow us to connect affective meaning to underlying group processes.

4.1 AFFECTIVE SIMILARITY

One important metric is the extent to which entities (e_i, e_k) feel similarly about the other named entities extracted from the document. In other words, do two entities see and feel about the world in a similar way (from the perspective of the author of the document)? Walter and Bruch (2008) emphasized the importance of affective similarity in work groups for sustained efforts to attain goals set by the group. We define the affective similarity $s : (U_\pm)^{n(n-1)} \times N \rightarrow U_\pm$ or affinity between entity pair (e_i, e_k) as a sum of cosine similarities (referred to as cosSum in Sect. 3.2.2) with every other entity in the document as shown in Eq. (4.1):

$$s_{ik}(\mathbf{T}_b, n) = \frac{1}{n-2} \sum_{\substack{j \in \mathrm{E} \\ j \neq i,k}} \cos(\mathbf{r}_{ij}, \mathbf{r}_{kj}) = \frac{1}{n-2} \sum_{\substack{j \in \mathrm{E} \\ j \neq i,k}} \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{kj}}{\|\mathbf{r}_{ij}\|_2 \|\mathbf{r}_{kj}\|_2} \tag{4.1}$$

The fraction preceding the summation symbol is a normalization factor where $n = |\mathbf{E}|$ whereas the cosine sum is a cumulative distance quantity.

The function *s* quantifies the degree to which an entity pair is perceived to have similar feelings about the world. Note that *s* is symmetric, i.e., $s_{ik}(\mathbf{T}_b, n) = s_{ki}(\mathbf{T}_b, n)$.

4.2 DIRECT AFFECTIVE RELATIONSHIP

Next consider the direct affective relationship between two entities in a document. We define a decomposition of the elements of the vector r_{ik} into four additive vectors:

$$\mathbf{r}_{ik} = \mathbf{r}_{ik}^{++} + \mathbf{r}_{ik}^{+-} + \mathbf{r}_{ik}^{-+} + \mathbf{r}_{ik}^{--} \tag{4.2}$$

where, for example, $\mathbf{r}_{ik}^{+-} = \langle r_{ik1}^{+-}, r_{ik2}^{+-}, \ldots r_{ikm}^{+-} \rangle$ is the vector of negatively-valued binary affective relationships between $\mathrm{e_i}$ and $\mathrm{e_k}$ for positive affect categories. For the components of $\mathbf{r}_{ik}^{+-}$ only negative affect relationships (e.g., -0.5) for positive affect categories (e.g., hate) receive non-zero values; in other words, this vector compiles the evidence for lack of negative affective relationships. The constructions for all four additive vectors are defined in Eqs. (4.3a) to (4.3d).

$$r_{ikj}^{++} = \begin{cases} r_{ikj} & \text{if } \mathrm{r_{ikj}} > 0 \text{ and } \mathrm{A_j} \in \mathrm{A}^b \cap \mathrm{A}^+ \\ 0 & \text{otherwise} \end{cases} \tag{4.3a}$$

$$r_{ikj}^{+-} = \begin{cases} r_{ikj} & \text{if } \mathrm{r}_{\mathrm{ikj}} > 0 \text{ and } \mathrm{A}_{\mathrm{j}} \in \mathrm{A}^{b} \cap \mathrm{A}^{-} \\ 0 & \text{otherwise} \end{cases} \tag{4.3b}$$

$$r_{ikj}^{-+} = \begin{cases} r_{ikj} & \text{if } \mathrm{r}_{\mathrm{ikj}} < 0 \text{ and } \mathrm{A}_{\mathrm{j}} \in \mathrm{A}^{b} \cap \mathrm{A}^{+} \\ 0 & \text{otherwise} \end{cases} \tag{4.3c}$$

$$r_{ikj}^{--} = \begin{cases} r_{ikj} & \text{if } \mathrm{r}_{\mathrm{ikj}} < 0 \text{ and } \mathrm{A}_{\mathrm{j}} \in \mathrm{A}^{b} \cap \mathrm{A}^{-} \\ 0 & \text{otherwise} \end{cases} \tag{4.3d}$$

Several measures can be defined to express direct affective relationships for (e_i, e_k). The function $R^+ : (U_+)^m \times (U_+)^m \times \mathbf{N} \times \mathbf{N} \rightarrow U_\pm$ assesses the strength of the evidence for the presence of positive affects between two entities, and $R^- : (U_+)^m \times (U_-)^m \times \mathbf{N} \times \mathbf{N} \rightarrow U_\pm$ does the same for negative affects. Finally, the function $R : U_\pm \times U_\pm \times \mathbf{N} \rightarrow U_\pm$ denotes the strength of the evidence for an overall positive affective relationship between two entities by taking into account both positive and negative affect.

$$R_{ik}^{+}\left(\mathbf{r}_{ik}^{++}, \mathbf{r}_{ik}^{-+}, m_b; \lambda\right) = pown\left[\left(\left\|\mathbf{r}_{ik}^{++}\right\|_2 - \left\|\mathbf{r}_{ik}^{-+}\right\|_2\right)\Big/\sqrt{m_b/2}, 1/\lambda\right] \tag{4.4a}$$

$$R_{ik}^{-}\left(\mathbf{r}_{ik}^{+-}, \mathbf{r}_{ik}^{--}, m_b; \lambda\right) = pown\left[\left(\left\|\mathbf{r}_{ik}^{+-}\right\|_2 - \left\|\mathbf{r}_{ik}^{--}\right\|_2\right)\Big/\sqrt{m_b/2}, 1/\lambda\right] \tag{4.4b}$$

$$R_{ik}\left(R_{ik}^{+}, R_{ik}^{-}; \lambda\right) = \left[R_{ik}^{+}\left(\mathbf{r}_{ik}^{++}, \mathbf{r}_{ik}^{-+}, m_b; \lambda\right) - R_{ik}^{-}\left(\mathbf{r}_{ik}^{+-}, \mathbf{r}_{ik}^{--}, m_b; \lambda\right)\right]\Big/2 \tag{4.4c}$$

Here m_b denotes the number of binary affect categories, and λ is a user-selected parameter (positive real number) that attempts to evenly distribute the function output through the entire range by means of a root transformation. None of the functions defined in Eqs. (4.4a), (4.4b) or (4.4c) are symmetric.

Observing the conjoint space formed by affective similarity and direct affective relationship reveals something about the nature of group processes when the group memberships of entities in a document are known. For example, consider the following partition of the affective space in Table 2:

Table 2. Examples of Social Meaning in a 2D Affective Index Space

Case	*Possible social meaning*
$s_{ik}(\mathbf{T}_b, n) > 0$ & $R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda), R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda) > 0$	Strong group cohesion
$s_{ik}(\mathbf{T}_b, n) > 0$ & $R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda), R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda) \leq 0$	In-group rivalry or competition, intergroup competition
$s_{ik}(\mathbf{T}_b, n) \leq 0$ & $R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda), R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda) > 0$	In-group heterogeneity
$s_{ik}(\mathbf{T}_b, n) \leq 0$ & $R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda), R_{ik}(R_{ik}^{+}, R_{ik}^{-}; \lambda) \leq 0$	Out-group antipathy or in-group factionalization

We can visualize the elements of this table in a 2-dimensional space as in Fig. 6:

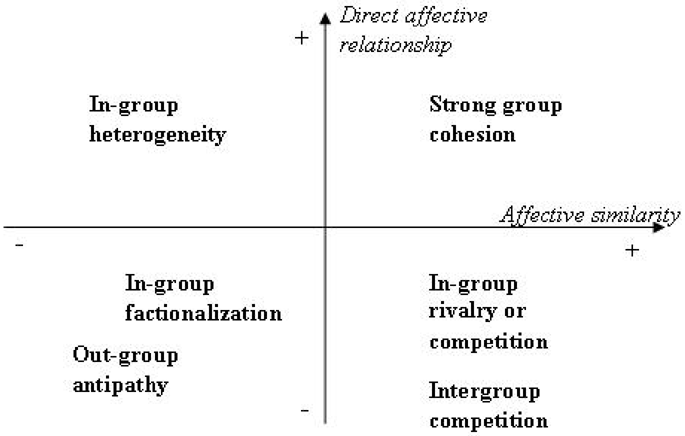


Fig. 6. Examples of social meaning portrayed in a 2D affect space.

4.3 SUBJECTIVITY

Subjectivity (with respect to the author) of text has previously been studied by NLP researchers (e.g., Weibe et al. 2004). News reporting is expected to convey minimal subjective content; conversely, op-ed pieces are typically saturated with subjectivity. We hypothesize that a large degree of the affective component of bias or subjectivity expressed in a document should manifest itself in the form of direct affective relationships between the author and named entities in the document. Let $AR = \{\mathbf{r}_{1i} : i = 2, \ldots, n\}$ be the set of affective relationships with author as subject. The degree of subjectivity in a document can be estimated by summarizing the extent of author relationships with other entities. For example, Eqs. (4.5a) and (4.5b) respectively show the relative frequency of author relationship with other entities using $J : (U_{\pm})^{n-1} \rightarrow U_{+}$ and the average author relationship intensity, $J^{*} : (U_{\pm})^{n-1} \rightarrow \mathbf{R}^{+}$. We define the Boolean variable $\phi_i = \begin{cases} 1 & if\ \mathbf{r}_{1i} \neq 0 \\ 0 & otherwise \end{cases}$

$$J(AR, n) = \frac{1}{n-1} \sum_{i=2,n} \phi_i \tag{4.5a}$$

$$J^{*}(AR, n) = \frac{1}{n-1} \sum_{i=2,n} \left\| \mathbf{r}_{1i} \right\|_2 \tag{4.5b}$$

where *n* is the number of entities in a document.

Affective similarity, direct affective relationship and subjectivity metrics for all entity pairs are computed and reported in the textual output of TEAMSTER.

5. VALIDATION OF AFFECT PROPAGATION ALGORITHM

The performance of the affect algorithm was evaluated using a subset of news reporting documents from the Multi-Perspective Question Answering (MPQA) corpus. The content of news articles is often rather dry with the author intent of maintaining at least a veneer of objectivity. This assessment obviously does not apply to op-ed articles, which tend to be richer in affective content. We elected to search for an appropriate subset of the larger corpus that was relatively charged with affective content. A 50-document subset was selected according to the following procedure. Documents were screened for affective content using the initial affect seed lists for all affect categories. The documents containing the highest absolute number of affect terms between 400 and 3600 words in length were selected.

Eleven parameters of the affect propagation model (Table 3) were optimized using a combination of random search, hill-climbing and simulated annealing. A range of parameter sets were further analyzed qualitatively to evaluate the effectiveness of the affect propagation algorithm on real unformatted documents.

Table 3. Optimization Parameters For The Affect Propagation Algorithm

Parameter Name	Symbol	Description
affectWeightage	z_a	Edge weight constant for affect tokens
entityWeightage	z_e	Edge weight constant for entity tokens
sentenceEndWeightage	z_p	Edge weight constant for sentence termination symbols {. ? !}
commaWeightage	z_c	Edge weight constant for commas
quoteWeightage	z_q	Edge weight constant for quotes
alpha	α	Restart probability for RWR algorithm
tBinary	t_b	Threshold intensity detection score for binary affective relationships
tUnary	t_u	Threshold intensity detection score for unary affective relationships
tAuthor	t_a	Threshold intensity detection score for affective relationships involving the author entity
rootTransformationConstant	γ	Scaling parameter applied after combining ranks of source entity and target entity for an affect token
ghostAuthorRank	β	Default “ghost” author rank

5.1 MPQA CORPUS

The MPQA corpus contains news articles and op-ed pieces manually annotated using a specially-crafted annotation scheme for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations). The corpus was initiated during the summer 2002 NRRC Workshop on Multi-Perspective Question Answering (MPQA) (Wiebe et al. 2005) sponsored by ARDA. The current release of the corpus contains 692 documents, a total of 15802 sentences. There are 5 distinct subsets of documents:

- MPQA original subset
- OpQA (Opinion Question Answering) subset
- XBank
- ULA (Unified Linguistic Annotation)
- ULA-LU (Language Understanding subcorpus)

These articles are from 187 different foreign and U.S. news sources dated from 2001–2002. They were identified by human searches and by an information retrieval system. The majority of the articles are on 10 different topics, but a number of additional articles were randomly selected (more or less) from a larger corpus of 270,000 documents. The 10 topics are:

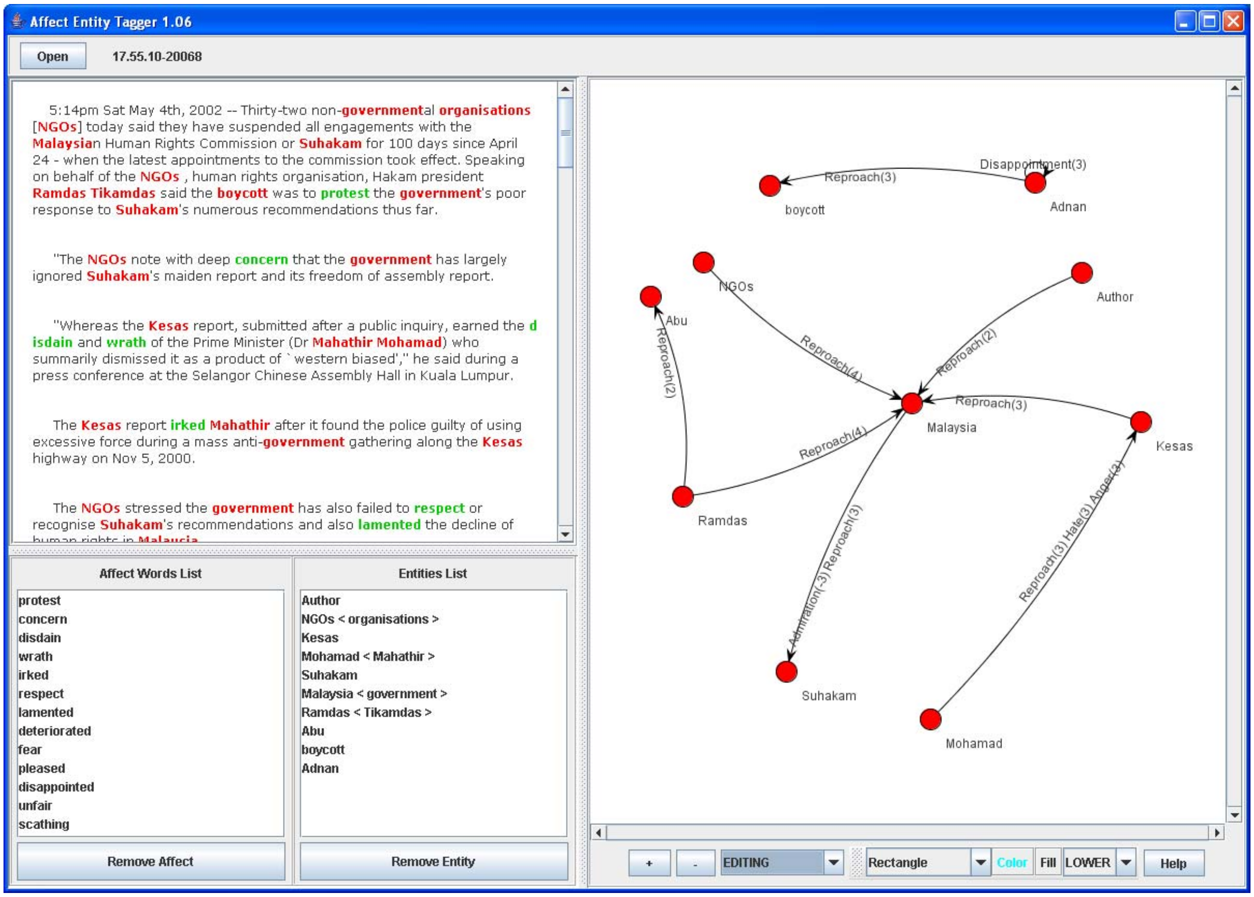
- argentina: economic collapse in Argentina
- axisofevil: reaction to President Bush's 2002 State of the Union Address
- guantanamo: U.S. holding prisoners in Guantanamo Bay
- humanrights: reaction to U.S. State Department report on human rights
- kyoto: ratification of Kyoto Protocol
- mugabe: 2002 presidential election in Zimbabwe
- settlements: Israeli settlements in Gaza and West Bank
- spacestation: space missions of various countries
- taiwan: relations between Taiwan and China
- venezuela: presidential coup in Venezuela

Most of the document subset we used for validation and optimization focused on one of the 10 topics listed above. The MPQA Corpus can be retrieved at http://www.cs.pitt.edu/mpqa/.

5.2 AFFECT-ENTITY ANNOTATION

Unfortunately the careful annotation scheme devised for the MPQA corpus was not adequately coded to identify affect-entity relations as they are conceptualized in this study. Therefore, we devised our own annotation scheme. The documents were tagged for affect terms and entities, and these tags served as a benchmark for validation of the affect propagation algorithm. One of the authors served as annotator with the aid of an affect-entity tagger application that was designed specially for this purpose and implemented as a Java application. A screenshot of the graphical user interface to the tagger is shown in Fig. 7.

The panel in the upper left contains the raw text with affect terms highlighted in green and entities in red. The affect word list and entities list are shown at lower left. The entity list displays co-referenced entities in angle-brackets. The application requires the annotator to double-click terms in the document, and add them to either the affect or entity list. Affect terms are also classified by category. Entities appear as red-colored nodes on the diagram on the right side of the window. The annotators' task is to connect and label entities with the appropriate affect label(s). The strength of the affect relation was designated by the user positioning a slider between the values of -5 and +5. More extreme values refer to more intense expressions of affective relationship. Negative values refer to a negation of the affect, as in "Iran does *not approve* of Israel." Note that affect negation is not equated to the opposite form of the affect term (i.e., the semantic meaning is not identical to "Iran does *disapprove* of Israel"). A score of zero indicates either a balanced ambivalent relationship or the lack of an affective relationship (most probably the latter.) A zero intensity score is the default for all possible relationships not explicitly scored by the annotator. An affect vector for a specific affective relationship between two entities comprises the intensity scores for all 22 affect dimensions taken together.



5:14pm Sat May 4th, 2002 -- Thirty-two non-governmental organisations [NGOs] today said they have suspended all engagements with the Malaysian Human Rights Commission or Suhakam for 100 days since April 24 - when the latest appointments to the commission took effect. Speaking on behalf of the NGOs , human rights organisation, Hakam president Ramdas Tikamdas said the boycott was to protest the government's poor response to Suhakam's numerous recommendations thus far.

"The NGOs note with deep concern that the government has largely ignored Suhakam's maiden report and its freedom of assembly report.

"Whereas the Kesas report, submitted after a public inquiry, earned the d isdain and wrath of the Prime Minister (Dr Mahathir Mohamad) who summarily dismissed it as a product of `western biased'," he said during a press conference at the Selangor Chinese Assembly Hall in Kuala Lumpur.

The Kesas report irked Mahathir after it found the police guilty of using excessive force during a mass anti-government gathering along the Kesas highway on Nov 5, 2000.

The NGOs stressed the government has also failed to respect or recognise Suhakam's recommendations and also lamented the decline of human rights in Malaysia

Affect Words List

protest
concern
disdain
wrath
irked
respect
lamented
deteriorated
fear
pleased
disappointed
unfair
scathing

Entities List

Author
NGOs < organisations >
Kesas
Mohamad < Mahathir >
Suhakam
Malaysia < government >
Ramdas < Tikamdas >
Abu
boycott
Adnan

Fig. 7. Graphical user interface to the affect-entity tagger application.

5.3 ERROR FUNCTION

The raw affect vectors could be used as direct input into the calculation of the error function, but the affect vectors have a special property or hidden bias that emerges while measuring distances between affect intensity values. The bias overemphasizes real differences at the extremes of the scale and minimizes real differences at the center of the scale. Therefore, affect vectors undergo a sign-preserving square-root transformation to emphasize real distances between low-valued intensities, and minimize real distances between high-valued intensities. For example, the distance between intensities 0 and +1 should exceed the distance between intensities +4 and +5. The justification for this nonlinear transformation of scores is that the former represents a distinction between relationship and non-relationship, whereas the latter signifies only a slight difference between two very positive scores. The transformed affect intensity y_i is calculated from the original affect intensity x_i as follows:

$$y_i = \operatorname{sgn}(x_i)\sqrt{|x_i|} \tag{5.1}$$

This transformation contracts distances between intensities at the extremes while allowing distances between intensities at the center of the range to remain large.

The error function adopted for the optimization problem is the average Euclidean error of the transformed intensity, and is defined as:

$$dist(\vec{y},\vec{z}) = \|\vec{y}-\vec{z}\| = \sqrt{\sum_i (y_i - z_i)^2} \tag{5.2}$$

where **y, z** are affect vectors respectively obtained from the affect propagation algorithm output and benchmark results. The distance function was calculated for the relationship between every possible entity pair (including unary relations). The final cumulative error score for the entire document set was just the average Euclidean error over all possible affective relationships and documents. The cumulative error became the basis of the fitness score for a parameterized affect propagation algorithm against the benchmark set of documents.

5.4 FITNESS SCORE

A fitness function based on average Euclidean error alone biases optimized solutions toward as few discovered affective relationships as possible. This property is due to the fact that most potential entity relationships (i.e., |**E**|^2) are not realized. An improved fitness score was derived that is actually a ratio of the expected error of a random guesser to the actual error generated by the affect propagation algorithm. The fitness ratio has a semantic interpretation, in that fitness reflects the extent to which the algorithm can improve upon an appropriate smart random guesser.

The performance of the random guesser (RG) is modeled as a sequence of independent Bernoulli trials for each potential affective relationship where the choice objects are in the set {*R, NR*}. The symbol *R* = “affective relation present” and *NR* = “affective relation not present”. The value of *RG* on a single Bernoulli trial follows the expression in Eq. (5.3):

$$RG = \begin{cases} R & \text{with probability } p \\ NR & \text{with probability } 1-p \end{cases} \tag{5.3}$$

The parameter *p* is just the probability of guessing that an affective relationship exists between any two entities. Let *q* represent the actual number of non-zero affective relationships in a document relative to the number of potential relationships (which is equal to |**E**|^2 where |**E**| is the average number of entities in a document.) An appropriate random guesser is one that matches the affect propagation algorithm’s propensity to find a relation between an arbitrary entity pair. If *r* is the total number of non-zero relations found by an algorithm, then:

$$\hat{p} = r/E^2 \tag{5.4}$$

There exist four possible outcomes of random guessing, and the expected values of these probabilities are given in Table 4. The probabilities off the main diagonal in Table 4 represent error outcomes.

Table 4. Expected Probabilities Arising From The Random Guess Model

		Ground Truth (baseline)	
		R	**NR**
Random Guess	**R**	pq	p(1-q)
	NR	(1-p)q	(1-p)(1-q)

We obtained an exact value for the average error using the baseline corpus in the case where a parameterized model predicts non-relationships between all entity pairs in the text (z = .0262477). The quantity z is therefore a constant with respect to the benchmark corpus taken as a whole. We call solutions from these models zero-vector solutions in the fitness space. Apparently a large number of parameter combinations generate zero-vector solutions for the affect algorithm. A small proportion (*q*) of possible affective relations will actually be explicitly coded into most real texts, but most (*1-q*) will be non-relations. For the benchmark corpus *q* = .10114. We can use Table 4 to decompose the error components of the constant z. There are two possible outcomes of a model generating zero-vector solutions: (1) a non-relationship (*NR*) is selected when ground truth indicates there exists a non-zero affective relationship with probability $(1-p)q$; and (2) a non-relationship (*NR*) is selected when ground truth indicates there is no affective relationship with probability $(1-p)(1-q)$. Condition (1) is a miss (error) and condition (2) is a correct rejection. A model generating a zero-vector solution will not miss any non-relations, because it never posits an affective relationship. Let ε equal the expected random guess error when a non-zero relationship is missed by the random guesser. The average error is $z=(1-p)q\varepsilon+(1-p)(1-q)\cdot 0$. But since for the zero vector solution $p=0$, we simplify the expression such that $z=q\varepsilon$. Therefore $\varepsilon = z/q$. Now let us also assume that the miss error ε can also serve as an approximate estimate of a false alarm, i.e., the detection of an affective relationship when ground truth indicates that none exists. Further, let $0<d<1$ equal the expected proportional error experienced when an affective relationship is correctly predicted but is still quantitatively off the mark. The expected random guess error for a single relationship taking into account all four possible outcomes from Table 4 is:

$$
\begin{aligned}
E(RG) &= \left[0\hat{p}q+\hat{p}(1-q)+q(1-\hat{p})+d\hat{p}q\right]*\varepsilon = \frac{\left[\hat{p}(1-q)+q(1-\hat{p})+d\hat{p}q\right]*z}{q} \\
&= \frac{\left[\hat{p}+q+(d-2)q\hat{p}\right]*z}{q} = \left\{1+\left[\frac{1+(d-2)q}{q}\right]*\hat{p}\right\}*z = \left\{1+\left[\frac{1+(d-2)q}{q}\right]*\left(\frac{r}{E^2}\right)\right\}*z
\end{aligned}
\tag{5.5}
$$

We designate the old fitness score (average Euclidean error) of the affect propagation algorithm by E(Alg). The fitness ratio is defined below.

$$
F=\frac{E(RG)}{E(A\lg)} \tag{5.6}
$$

where F ≥ 0. When F >1 the algorithm predicts more accurately than a similar random guesser. The value of F can be infinite, but never attains this value unless the algorithm achieves a perfect match with the benchmark data (zero error). The improved fitness function eliminates the bias against finding relations. Instead, it attempts to balance the total numbers of false alarms and misses, leading to an optimum number of discovered relations. We note that F is just a linear transformation of the ratio $\hat{p}/E(A\lg)$. In order to compute the fitness ratio we still require an estimate of the parameter *d*. We are using the value of *d* = 0.44.

5.5 OPTIMIZATION

The optimization study addressed only the affect propagation algorithm. We did not desire to allow the performance of the algorithm to be affected by the performance of the entity tagger and other basic natural language processing algorithms as these aspects of the software were not central to the function of the new algorithm for affect propagation. Accordingly, for the purposes of the optimization study, the benchmark entities and affect terms were supplied directly to the affect propagation algorithm. The algorithm automatically processed all 50 documents from the test corpus using a single parameter set. The output of this processing was a single fitness score that could be compared to fitness scores resulting from other parameter sets. A new parameter set was generated and subsequently evaluated for its fitness. This process iterated until the optimization algorithm terminated. The parameter set with the highest fitness was taken as an indicator of the performance of the optimization algorithm.

First, a series of 5000 random searches over the entire parameter space was conducted in order to establish a performance benchmark for the hill-climbing and simulated annealing algorithms. The summary statistics for the improved fitness score are: (1) maximum = 1.402; (2) minimum = 0.805; (3) mean = 0.996; (4) standard deviation = 0.067. The best fitness obtained by random search was therefore 1.402, representing a 40% improvement over intelligent random guessers.

5.5.1 Stochastic Hill-Climbing Optimization

The initial optimization algorithm employed was a simple stochastic hill-climber over a continuous space initialized with a random parameter set. The current best parameter set is randomly perturbed on all dimensions and then evaluated. Only new parameter sets with a better fitness score are accepted and therefore become the new standard. This class of hill-climbing algorithms is vulnerable to entrapment in local minima over the solution space, but if the initial solution is close to a global minimum, then hill-climbing can be an effective method of finding the local minimum for that solution. Parameter modifications are executed with a fixed step size of .001, with each parameter value having an equal chance of being increased, decreased or no change. The pseudo-code for the hill-climbing algorithm is show below:

```
Method :: executeHillClimbingOptimization()
NO_OF_EVALUATIONS = 50;
NO_OF_PARAMETERS = 10;
MAX_NO_OF_PARAMETER_COMBINATIONS = 100;
for i = 1 to NO_OF_EVALUATIONS do
        newParameterList = generateRandomParameterList();
        currentFitness = calculateFitness(newParameterList, documentSet);
        do
               log(currentFitness);
               oldParameterList = newParameterList;
               previousFitness = currentFitness;
               newParameterList = stochasticFirstChoice(oldParameterList, currentFitness, 1);
               currentFitness = calculateFitness(newParameterList, documentSet);
        while (currentFitness > previousFitness)
end for

Method :: stochasticFirstChoice(oldParameterList, previousFitness,
                parameterSelector)
while parameterSelector <= MAX_NO_OF_PARAMETER_COMBINATIONS do
       for i = 0 to NO_OF_PARAMETERS do
               V[i] = generateRandomIntegerInRange(0,2);
        end for
        resultList = generateNextParameterList(V, oldParameterList);
        currentFitness = calculateFitness(resultList, documentSet);
        if (currentFitness > previousFitness) then
                return resultList;
        else
               parameterSelector++;
               return (stochasticFirstChoice(oldParameterList, previousFitness,
                         parameterSelector));
        end if
end while
```

```
return previousFitness;

Method :: generateNextParameterList(V, parameterList)
// This if else is repeated for all 11 parameters (for 11 elements inside V vector) with
different stepsizes and ranges
if (V[0]==1) then
        if(parameter1< 0.999) then
           newParameter1 = parameter1 + 0.001; // stepsize is 0.001 and range is (0,1)
        else
          newParameter1 = parameter1 + 0.001 - 1;
else if (V[0]==2) then
        if(parameter1< 0.999) then
           newParameter1 = parameter1 - 0.001;
        else
          newParameter1 = parameter1 - 0.001 + 1;
end if

Method :: calculateFitness(parameterList, documentSet)
cumulativeError = evaluateAEPropAlgorithm(parameterList, documentSet);
fitness = cumulativeError / (size(documentSet));
return fitness;
```

5.5.2 Simulated Annealing

Simulated annealing (Aarts et al. 1997, Romeo and Sangiovanni-Vincentelli 1991) represents an advancement over stochastic hill-climbing in that it is capable of avoiding entrapment in local minima and therefore has a better chance of approaching the global minimum. The basic idea for improvement is that solutions with an inferior fitness score have a non-zero probability of acceptance. This feature allows the algorithm to probabilistically jump outside troughs that are potential local minima. The acceptance probability decreases as the algorithm progresses, thus simulating a physical annealing process as the sequence settles into a low-energy state. The actual probability of accepting a less fit solution is $\exp(deltaE/(T * previousFitness))$ where $deltaE = currentFitness - previousFitness$ and T is the iteration number. The acceptance probability decreases as the ratio $deltaE/previousFitness$ becomes more extreme (i.e., larger relative decrements in fitness have a lower chance of acceptance).

We adopted an enhancement over simple simulated annealing to allow the optimization process to adapt to local fitness landscape conditions. When the fitness score is relatively flat we permit a wide range of search parameters to prevail in order to stimulate a new and productive direction of search. Conversely, when great improvements in fitness are observed the scope of parameter search is narrowed under the assumption that the algorithm is “on the right track.” Accordingly, Wang and Chen (1997) utilize a symmetric Beta distribution to model the variable step size for the modification of parameter values as follows:

$$\alpha = \beta = \begin{cases} \gamma * (previousFitness/currentFitness)^{\theta} & \text{if previousFitness} \geq \text{currentFitness} \\ \gamma * (currentFitness/previousFitness)^{\theta} & \text{if currentFitness} > \text{previousFitness} \end{cases} \tag{5.7}$$

The effect of Eq. (5.7) is that a greater disparity between the current and previous fitness scores will generate a less variable and more peaked Beta distribution. Every parameter has a 0.5 probability of modification after each iteration. The pseudo-code for adaptive simulated annealing is shown below:

Method :: executeAdaptiveSimulatedAnnealingOptimization()

```
NO_OF_EVALUATIONS = 1;
NO_OF_PARAMETERS = 11;
Theta = 5;      // exponent for adaptive step size selection
Gamma = 6;  // multiplicative constant for adaptive step size selection
Decay = 0.98;   // temperature decay factor
Qprob = 0.5;     // probability of mutating a single dimension in the ParameterList
Jmin = 100;  // minimum number of iterations
Jmax = 1500;  // maximum number of iterations
Jc = 100        // Sequence length for stopping criterion
K = .0005;     // change threshold for stopping criterion
for i = 1 to NO_OF_EVALUATIONS do
        oldParameterList = generateRandomParameterList();
        currentFitness = calculateFitness(oldParameterList, documentSet);
        T = 5;    // initial temperature
        previousFitness = currentFitness;
        iter = 1;
        Converge = FALSE;
        totUnder = 0;
        do
                Alpha = (maximum(previousFitness/currentFitness,
                        currentFitness/previousFitness)$^{Theta}$;
                newParameterList = randomNextParameterList(oldParameterList, Alpha);
                currentFitness = calculateFitness(newParameterList, documentSet);
                log(currentFitness, previousFitness, T, iter)    // save to file

                deltaE = currentFitness – previousFitness;
                randomNum = generateRandomRealNumberInRange(0, 1);

                if (deltaE<0 OR randomNum <= $e^{deltaE/(T*previousFitness)}$) then
                        log(currentFitness);
                        previousFitness = currentFitness;
                        oldParameterList = newParameterList;
                end if
                if (ABS(deltaE) <= K) then totUnder++;
                else totUnder = 0;
                endif
                if (iter > Jmin AND totUnder >= Jc) then Converge == TRUE;
                else Converge = FALSE;
                endif

                T = Decay*T;
                iter++;
        while (iter <= Jmax AND Converge == FALSE)
end for
```

Method :: randomNextParameterList(oldParameterList, Alpha)

```
for i = 0 to NO_OF_PARAMETERS do
        V[i] = Random();    // get a random value in the interval (0,1)
end for
resultList = generateNextParameterList(V, oldParameterList, Alpha);
return resultList;
```

```
Method :: generateNextParameterList(V, parameterList, Alpha)
// This if else is repeated for all 11 parameters (for 11 elements inside V vector) with
different stepsizes and ranges
if (V[0] <= Qprob) then
    Brand = B(Alpha, Alpha);        // random draw from Beta distribution
    if (Brand <= 0.5) then
        newParameter1 = 2*Brand*parameter1;
    else
        newParameter1 = parameter1 + (2*Brand-1)*(1-parameter1);
    endif
endif

Method :: calculateFitness(parameterList, documentSet)
cumulativeError = evaluateAEPropAlgorithm(parameterList, documentSet);
fitness = cumulativeError / (size(documentSet));
return fitness;
```

The stopping criterion for simulated annealing included a minimum (100) and maximum (1500) total number of iterations as well as a convergence check. The iterative sequence was considered to have converged if there was no improvement/modification of the accepted parameter list greater (less) than a fitness value of .0005 for 100 iterations. All simulated annealing parameters were fine-tuned during the testing phase.

We scheduled each simulated annealing run to be followed by a modified stochastic hill-climbing algorithm to search for optimum parameter sets that would fine-tune the affect propagation algorithm. Instead of initializing a random parameter set, the stochastic hill-climber began with the best solution from simulated annealing, and then proceeded through a series of 3 step size changes: .001, .01 and .1. The parameter list with the highest fitness score, as well as a number of other qualitatively interesting solutions, were extracted from the sequence generated during the final optimization run.

6. RESULTS OF THE OPTIMIZATION STUDY

Fitness score apparently varied as a function of total number of non-zero affective relationships found by the affect propagation algorithm. Figure 8 shows fitness score as a function of number of relationships from a sample of random searches. Figures 9 and 10 show the same function generated from the output of two separate simulated annealing optimization runs. All three plots display the same general pattern of fitness increasing from an initial value of 1.0 (obtained by predicting zero affective relations). Fitness then increases monotonically with the number of affective relations found, peaking at around 600–1600 relations found (random search), 1300 to 1500 affective relations found (first simulated annealing run) and 1550–1650 affective relations found. Fitness trails off monotonically for higher numbers of discovered relations. The total number of affective relations marked non-zero in the benchmark set is 487. Peak fitness scores occurred for solutions predicting roughly three times this number of discovered relations. In Figs. 9 and 10, the fitness for solutions predicting about 487 relationships was roughly in the neighborhood of 1.425.

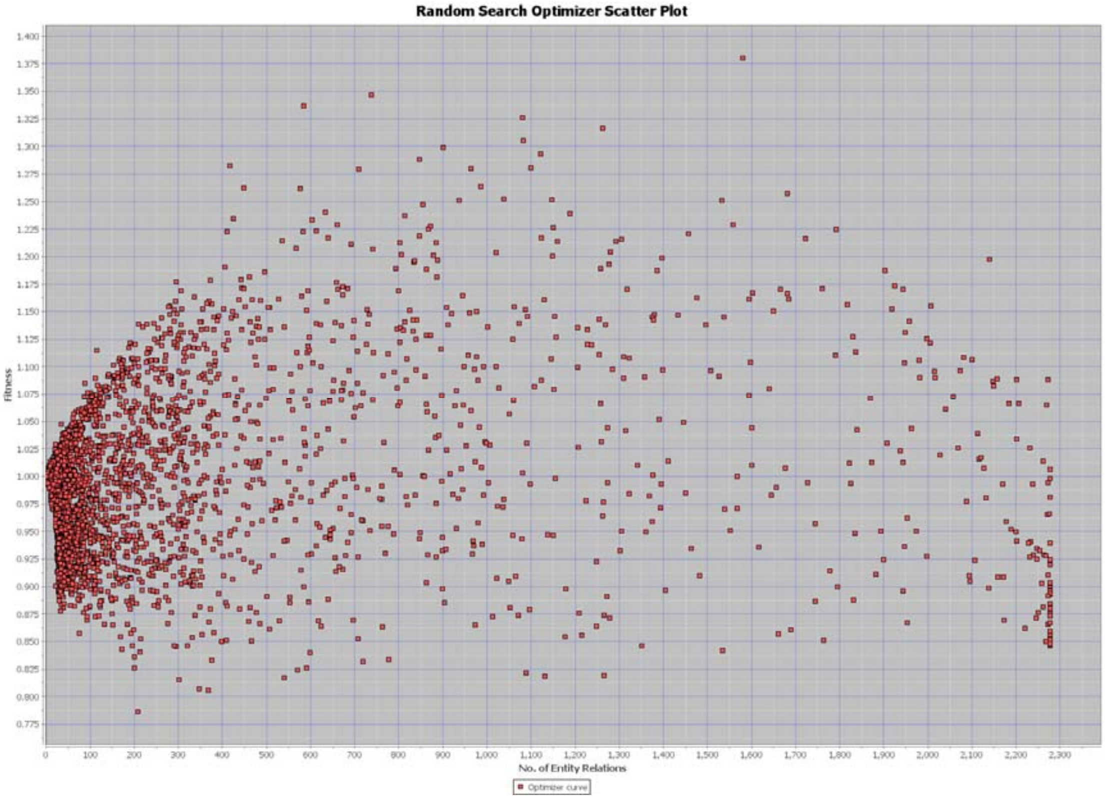


Fig. 8. Fitness as a function of total number of affective relations from random search.

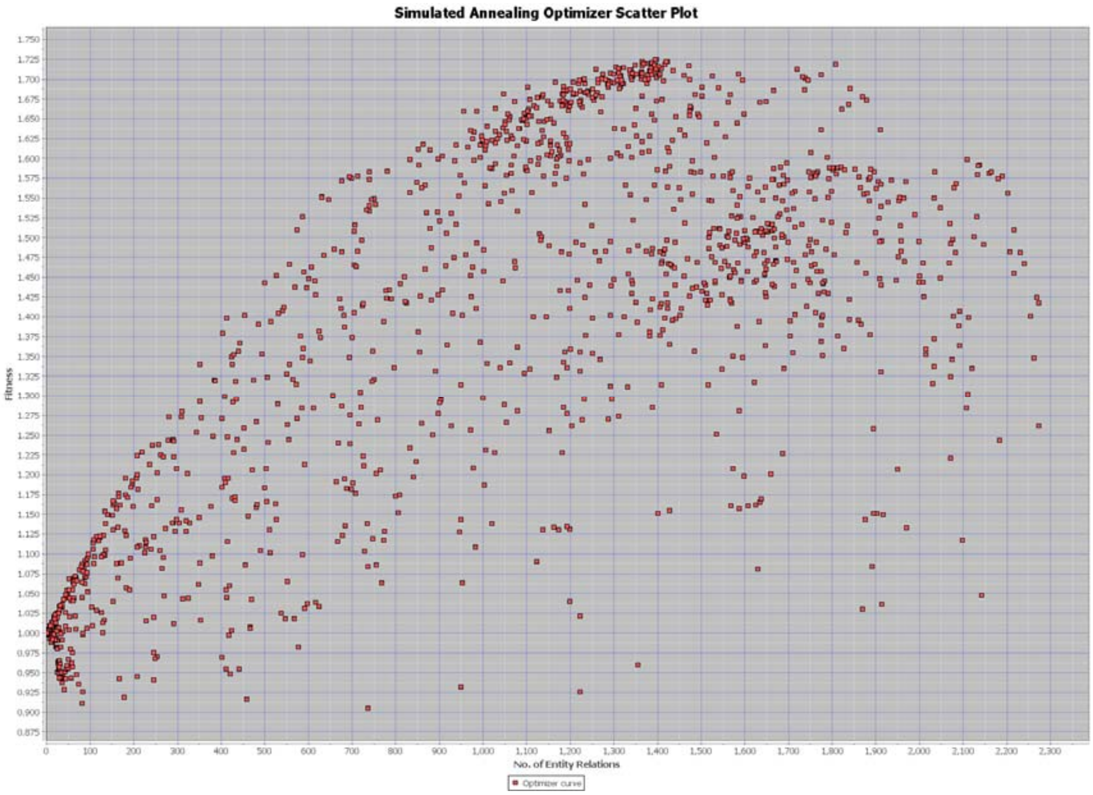


Fig. 9. Fitness as a function of total number of affective relations from simulated annealing optimization (1).

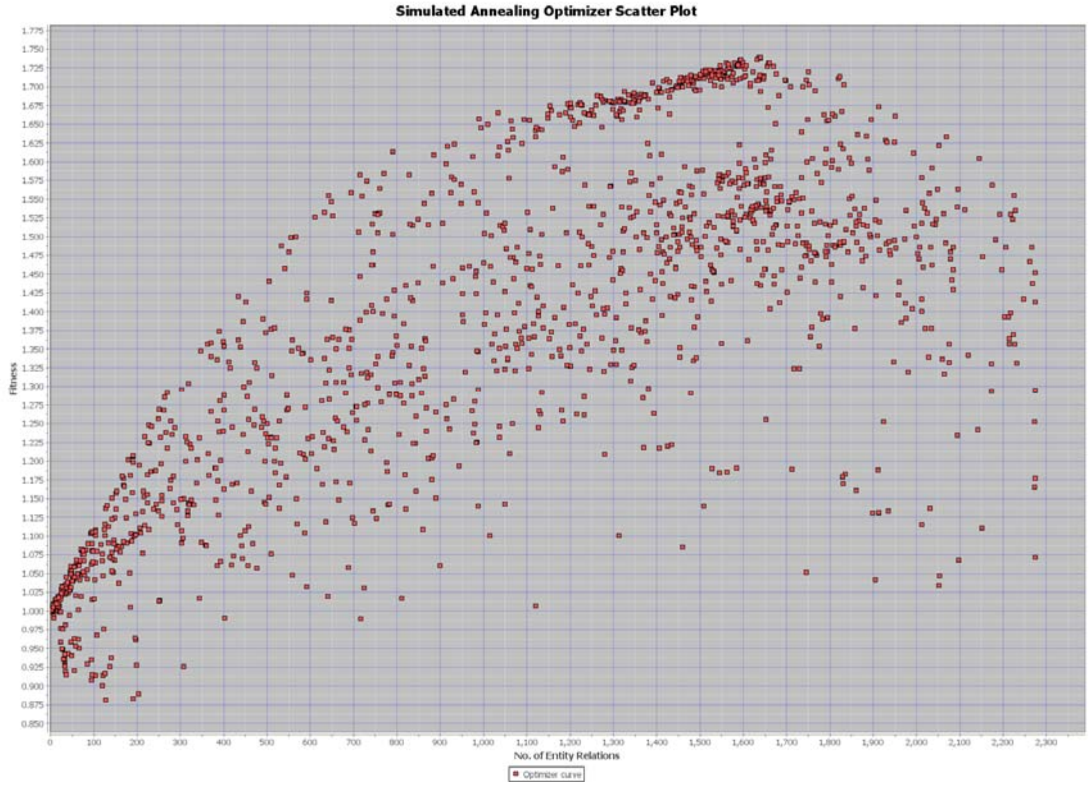


Fig. 10. Fitness as a function of total number of affective relations from simulated annealing optimization (2).

Simulated annealing successfully discovered many parameter solutions with greater fitness than the 5000-trial random search, and did so in less than a third the number of iterations. Figure 11 shows the performance of the first simulated annealing run. Following an initial phase of seemingly random search (where temperature was very high), the algorithm settled into a phase of picking solutions that discovered very few relations, resulting in fitness scores at or below 1.0. At about 225 iterations the algorithm entered a fast learning phase while simultaneously entertaining solutions that discovered many more affective relations. From about 300–1400 iterations the algorithm entered a slow learning phase. The second simulated annealing run (Fig. 12) revealed a similar pattern, but skipped the zero-relationship solution phase. After 750 iterations the algorithm accomplished a step-increase in fitness where it peaked for the remainder of the 1500-iteration run.

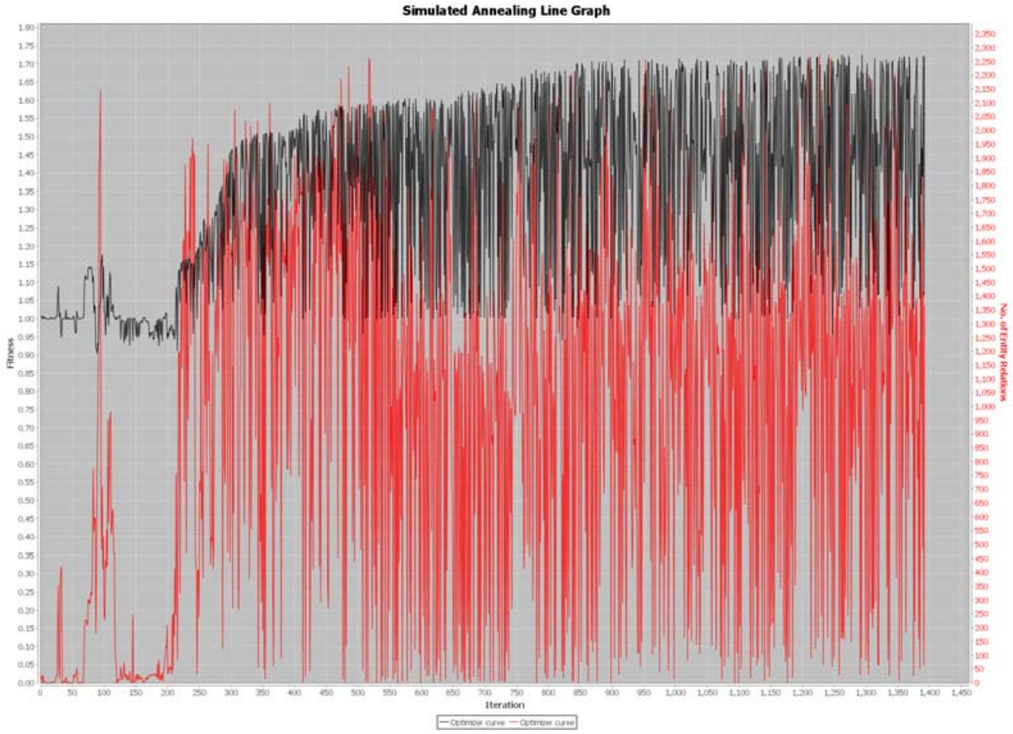


Fig. 11. Fitness results from an initial simulated annealing optimization of the affect propagation algorithm using 50 selected MPQA document annotations.

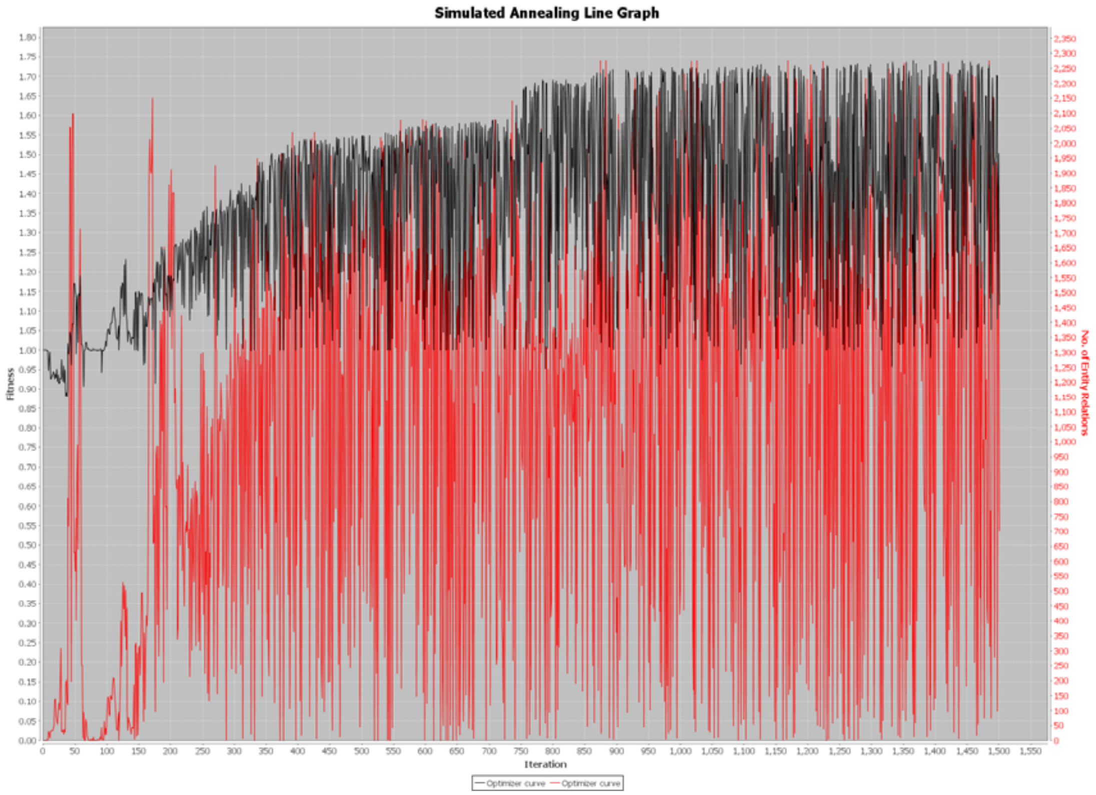


Fig. 12. Fitness results from a second simulated annealing optimization of the affect propagation algorithm using 50 selected MPQA document annotations.

The best solution obtained from simulated annealing + stochastic hill-climbing achieved a fitness score of 1.742; values of the parameter list are in Table 5. Four of five edge weight reduction parameters were awarded high values, suggesting that they were not highly effective in regulating spreading activation of flow from affect terms to entities. The exception was 'quote'– apparently affective flow did not naturally penetrate a quotation block from either the inside or outside. The alpha parameter was very low – this would have the effect of minimizing random jumps back to the affect token and equalizing the PageRank scores of all entities close to the start node. Thresholds for binary and unary affects were very low; very few affect-entities pairs were dropped due to low associations as measured by the PageRank scores. However, the author entity threshold was quite high, effectively removing most affective relationships involving the author from consideration. Finally, the root transform constant for distributing scores evenly along the (0,1) interval was established at approximately the value for a cube root.

Table 5. Highest Fitness Parameter Values For Affect Algorithm

Parameter Name	Parameter Value
affectWeightage	0.80841
entityWeightage	0.99590
sentenceEndWeightage	0.98719
commaWeightage	0.75694
quoteWeightage	0.09206
alpha	0.00476
tBinary	0.00939
tUnary	0.05224
tAuthor	0.80871
rootTransformationConstant	0.35432
ghostAuthorRank	0.00010

The solution with the highest fitness score generated 1355 non-zero affective relationships. The affective structure produced by the best fitness solution was qualitatively compared to the affective structures generated by two other solutions with high fitness scores. However, these two solutions were qualitatively superior in the sense that the generated numbers of affective relationships that were comparable to the benchmark set of documents.

Figures 13–16 show the affect-entity relationship diagrams for the benchmark and all three selected parameter solutions with respect to MPQA document 20.24.49-8480, an op-ed piece containing some fairly direct affective statements. The best fitting parameter solution (Fig. 14) generated an affect-entity diagram cluttered with relationships. Some affective relations from the benchmark diagram were captured by the affect algorithm generated diagrams. For example, admiration (Fig. 14) and reproachfulness (Figs. 14–16) of the author toward 'country' (Korea) were represented in the algorithm diagrams, as was the reproachful attitude of Korea toward the United States (Figs. 14–16), the reproachful attitude of author toward the United States (Figs. 14–16), reproach (Figs. 14–16) and hate (Fig. 14) of the author for President Bush, and reproach of the international community toward President Bush (Figs. 14–16). The algorithm-generated solutions appeared to spread activation of certain affects beyond intended boundaries to additional, but unintended, entity sources and targets. For example, the reproachful attitudes of the author toward the United States, President Bush and country also spread to the notion of reunification (Figs. 14 and 16), the government (Figs. 14 and 16) and the international community (Figs. 14–16). There was no evidence of these affective relationships in the document; indeed, they appear to contradict other evidence. There was also evidence for an over-abundance of reciprocal affective relationships beyond that which was expressed in the benchmark diagram. For example, President Bush and the United States reciprocated the reproachful attitude of the author toward them (Figs. 14–16). Some of these additional affective relationships may be discoverable through affective inference, but that topic is beyond the scope of this report.

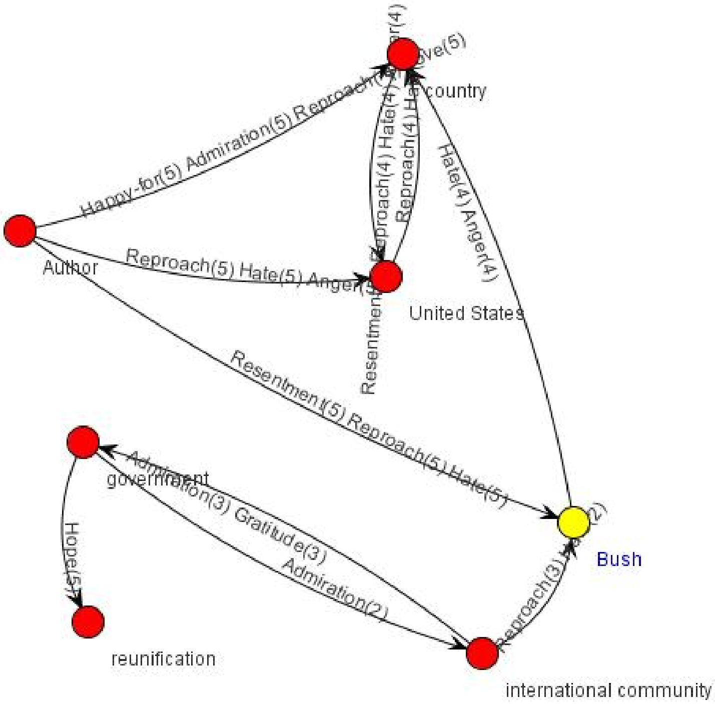


Fig. 13. Benchmarked affect-entity diagram for document 20.24.49-8480.

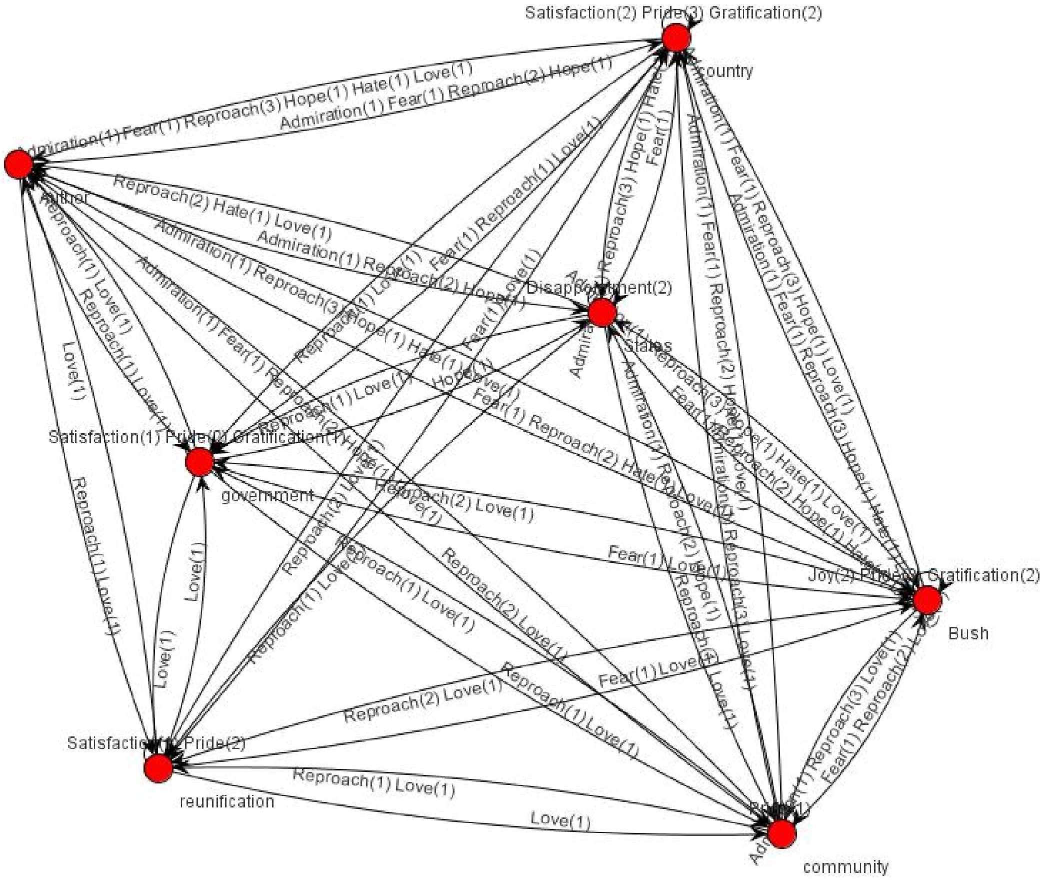

Satisfaction(2) Pride(3) Gratification(2)
country
Admiration(1) Fear(1) Reproach(3) Hope(1) Hate(1) Love(1)
Admiration(1) Fear(1) Reproach(2) Hope(1)
Author
Reproach(2) Hate(1) Love(1)
Admiration(1) Reproach(2) Hope(1)
Admiration(1) Reproach(3) Hope(1) Hate(1)
Disappointment(2)
States
Satisfaction(1) Pride(2) Gratification(1)
government
Reproach(2) Love(1)
Fear(1) Reproach(2) Hate(1) Love(1)
Joy(2) Pride(3) Gratification(2)
Bush
Reproach(1) Love(1)
Reproach(2) Love(1)
Fear(1) Love(1)
Reproach(2) Love(1)
Love(1)
Satisfaction(1) Pride(2)
reunification
Reproach(1) Love(1)
Love(1)
community

Fig. 14. Affect algorithm (highest fitness - 1379 relations) generated affect-entity diagram for document 20.24.49-8480.

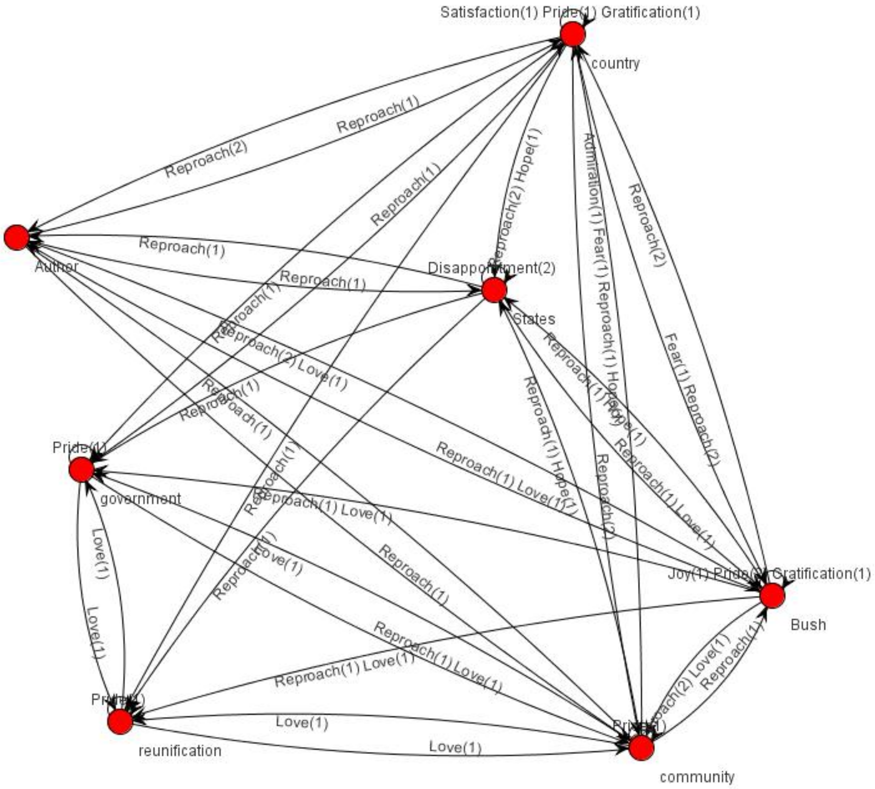


Fig. 15. Affect algorithm (746 relations) generated affect-entity diagram for document 20.24.49-8480.

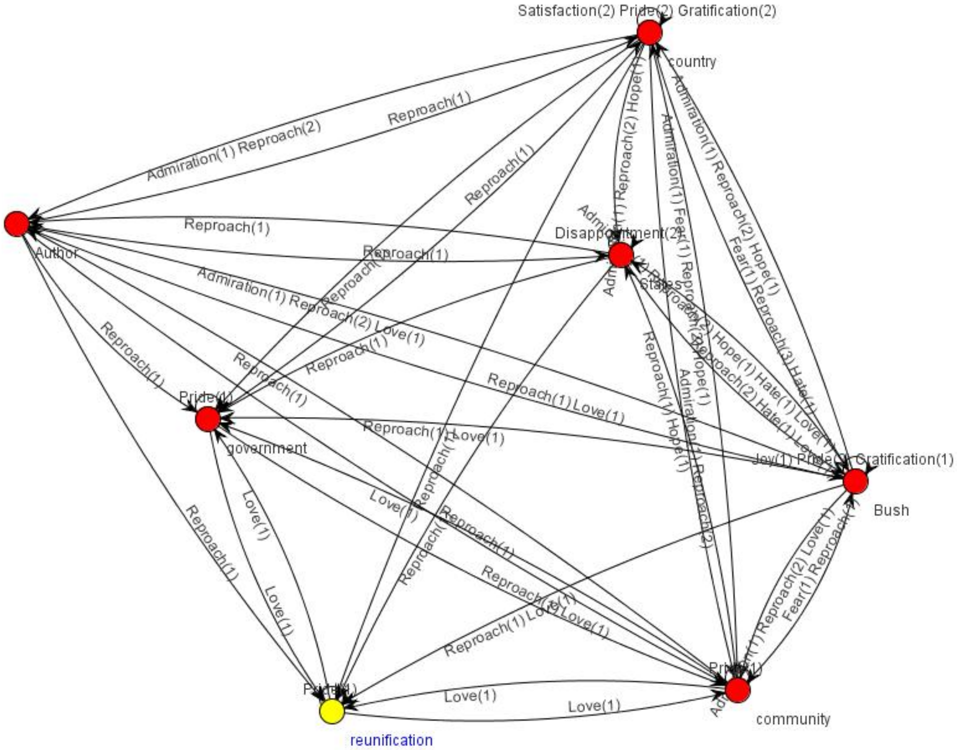


Fig. 16. Affect algorithm (581 relations) generated affect-entity diagram for document 20.24.49-8480.

7. FUTURE DIRECTIONS

This research represents only the initial stages of the effort to extract detailed affective meaning from text. Many tasks and challenges remain before affective computing can claim to be a mature research area. The performance of the affect extraction algorithms could be improved significantly by a more thorough accounting of context effects in documents. Different sentence constructions such as active vs. passive voice ought to exert different influences on the mechanism of affect propagation. Word sense disambiguation can also be quite important. For example, the key word "support" functioning as a verb could be a synonym for the "admiration" affect category, or denote a structural element functioning as a noun. The seed lists currently are annotated by word sense, but the specific word sense is not used during the search for affect tokens in documents. The reason for this deficiency is that TEAMSTER does not currently have a word disambiguation algorithm, although many techniques have been described in the NLP literature. Co-reference resolution among entities, including pronoun references is an important feature for improved entity recognition. Effects of conditionals and counterfactual statements should be parsed appropriately. Suppose an author writes, "If you do this thing, I will hate you for it." The author is threatening hatred toward a target entity, but it doesn't have the same force as the present tense "I hate you." Nested sources comprise another area of contextual complexity that is especially prevalent in news reporting. An example of a nested source is found in "The White House spokesperson indicated that the President feels that this is a bad bill and intends to veto it if passed by Congress." This text reveals the President's dislike of the bill, but says nothing directly about the opinion of the spokesperson or the author.

A second priority for the improvement of TEAMSTER functionality is to enable full connectivity of the seed lists, WordNet and ConceptNet. Currently TEAMSTER searches synsets from WordNet, but other graph relations may also be valuable conduits for the diffusion of affective meaning, such as the "similar to" relation. We also hope to construct and/or integrate political affective lexicons to leverage current political usage of affectively-saturated terms. For example, during the 2008 U. S. Presidential election the word "socialist" was frequently employed as a pejorative term in political discourse among American conservatives. However, this term would not likely appear in any conventional affect lexicon, as the affective tinge is "acquired" by association.

A final limitation of the affect lexicon is that it is constructed along the lines of the principle of centrality or representativeness of concepts with respect to a specific affect category. This is due to the application of the PageRank algorithm. The words "adore" and "like" are highly representative of the affect category of "love", and therefore receive high PageRank scores. However, "adore" connotes a more intense feeling or depth of affect as compared to "like." There is a need to find a means to incorporate intensity into the word score of terms in the affect lexicon.

An important goal of this research is to extend the analysis from single to multiple documents. The evolution of affective relationships can be tracked by performing change detection on the structure of the affect-entity relation network over time-ordered multiple texts from a single source. Entities can also be clustered on the basis of affective similarity. A dynamic analysis of the affect-induced entity space displayed in an evolution panel could provide insightful indications of formation and/or dissolution of social/political groups with shared affect.

Finally, an exciting area of future research is affective inference. Inferring affect from text implies going beyond the denotative and connotative meaning of a text. Affective inference reads between lines of text to discover what additional statements could be made about affective relationships among entities discussed in a text. For example, the sample text specifies that "The U.S. maintains strong support and liking for Israel" and later states that "Israel harbors a deep animosity and fear of Iran…" A reader might conclude solely from these two propositions in evidence that "The U. S. harbors animosity of fear of Iran." We are not aware of any substantial experimental or theoretical literature on the topic of affective inference, but we predict it could become an important area of research in the future.

8. CONCLUSION

In this report we introduced a new method for extracting specific affect words at the sentence and paragraph level and for attributing affective meaning to specific documents. We extended affective meaning to include affective relationships between entities extracted from texts. We have also developed a technique for building large collections of affect words and phrases using seed word lists and the common sense built into such open source resources as WordNet and ConceptNet. Finally, we incorporated these algorithms into a software prototype called TEAMSTER.

The affect propagation algorithm was validated using a high-affect 50-document subset of the MPQA corpus. These documents were manually annotated for affects, entities and affective relationships. Eleven free parameters in the algorithm were optimized using a simulated annealing cooling schedule. The optimized affect propagation algorithm was found to perform 74% better than comparable random guess models.

The results demonstrate that our affect propagation algorithm represents a good start to computer-mediated extraction of affective meaning from text, but significant progress is still needed before it can approach the accuracy of human judgment.

9. REFERENCES

Aarts, E. H. L., J. H. M. Korst, and P. J. M. van Laarhoven. 1997. “Simulated Annealing.” Chapter 4 in Local Search in Combinatorial Optimization, pp. 91–120.

Abbasi, A., H. Chen, S. Thoms, and T. Fu. 2008. “Affect Analysis of Web Forums and Blogs Using Correlation Ensembles.” IEEE Transactions on Knowledge and Data Engineering, 20(9): 1168–1180.

Abbasi, A. and H. Chen. 2007. “Affect Intensity Analysis of Dark Web Forums.” Proceedings of IEEE International Conference on Intelligence and Security Informatics (ISI 2007), New Brunswick, NJ, pp. 282–288.

Allan, K. 2007. “The Pragmatics of Connotation.” Journal of Pragmatics, 39: 1047–1057.

Alm, C. O., D. Roth, and R. Sproat. 2005. “Emotions from Text: Machine Learning for Text-Based Emotion Prediction.” Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, October.

Al Masum, S. M., H. Prendinger, and M. Ishizuka. 2007. “Emotion Sensitive News Agent: An Approach towards User Centric Emotion Sensing from the News.” Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 614–620.

Aman, S. and S. Szpakowicz. 2008. “Using Roget’s Thesaurus for Fine-Grained Emotion Recognition.” Proceedings of the Third International Joint Conference on Natural Language Processing.

Anderson, J. R. 1983. “The Architecture of Cognition.” Cambridge, MA: Harvard Univ. Press

Anolli, L., V. Zurloni, and G. Riva. 2006. “Linguistic Intergroup Bias in Political Communication.” The Journal of General Psychology, 133(3): 237–255.

Barside, S. G. and D. E. Gibson. 1998. “Group Emotion: A View from Top and Bottom.” Book Chapter in Research on Managing Groups and Teams, Vol. 1, JAI Press, pp. 81–102.

Besnier, N. 1990. “Language and Affect.” Annual Review of Anthropology, 19: 419–451.

Bianchini, M., M. Gori, and F. Scarselli. 2005. “Inside PageRank.” ACM Transactions on Internet Technology, 5(1): 92–128.

Brin, S. and L. Page. 1998. “The Anatomy of a Large-Scale Hypertextual Web Search Engine.” Proceedings of Seventh International World-Wide Web Conference (WWW 1998) [Online]. Available: http://ilpubs.stanford.edu:8090/361/

Cohen, W.W. 2010. “Graph Walks and Graphical Models.” School of Computer Science, Carnegie Mellon University: CMU-ML-10-102.

Crestani, F. 1997. “Application of Spreading Activation Techniques in Information Retrieval.” Artificial Intelligence Review, 11: 453–482.

Ekman, P. 1999. “Basic Emotions.” In T. Dalgleish and M. Power (Eds.) Handbook of Cognition and Emotion. Sussex, U.K.: John Wiley and Sons, Ltd.

Esuli, A. and F. Sebastiani. 2007. “Random-Walk Models of Term Semantics: An Application To Opinion-Related Properties.” Proceedings of LTC’07, pp. 221–225.

Gobron, S., J. Ahn, G. Paltoglou, M. Thelwall, and D. Thalmann. 2010. “From Sentence to Emotion: a Real-Time Three-Dimensional Graphics Metaphor of Emotions Extracted from Text.” The Visual Computer, 26: 505–519.

Graesser, A.C., K.K. Millis, and R.A. Zwaan. 1997. “Discourse Comprehension.” Annual Review of Psychology, 48: 163–189.

Harabagiu, S., A. Hickl, and F. Lacatusu. 2006. “Negation, Contrast and Contradiction in Text Processing.” Proceedings of the Twenty-first National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, Boston, MA., July 16–20.

Havasi, C., R. Speer, and J. B. Alonso. 2007. “ConceptNet 3: A Flexible, Multilingual Semantic Network for Common Sense Knowledge.” Proceedings of Recent Advances in Natural Language Processing, Borovets, Bulgaria, September.

He, S., D. Ballard, and D. Gildea. 2004. “Building and Tagging with an Affect Lexicon.” Technical Report 833, Rochester, NY: The University of Rochester, Computer Science Department.

Hewstone, M., M. Rubin, and H. Willis. 2002. “Intergroup Bias.” Annual Review of Psychology, 53: 575–604.

Jurafsky, D. and J. H. Martin. 2009. “Speech and Language Processing.” Upper Saddle River, NJ: Pearson Prentice Hall.

Lemaire, B., G. Denhière, C. Bellissens, and S. Jhean-Larose. 2006. “A Computational Model for Simulating Text Comprehension.” Behavior Research Methods, 38(4): 628–637.

Liu, H., H. Lieberman, and T. Selker. 2003. “A Model of Textual Affect Sensing Using Real-World Knowledge.” Proceedings of IUI, Miami, FL, January, pp. 12–15.

Liu, H. and P. Singh. 2004. “ConceptNet: a Practical Commonsense Reasoning Tool-Kit.” BT Technology Journal, 22(4): 211–2226.

Lu, C., J. S. Hong, and S. Cruz-Lara. 2006. “Emotion Detection in Textual Information by Semantic Role Labeling and Web Mining Techniques.” Proceedings of the Third Taiwanese-French Conference on Information Technology, Nancy, France.

Lucas, P. J. F. 2001. “Certainty-Factor-Like Structures in Bayesian Belief Networks.” Knowledge-Based Systems, 14: 327–335.

MIT Media Laboratory, 2009. “ConceptNet Website.” http://web.media.mit.edu/ hugo/conceptnet/, Retrieved April.

MIT Media Laboratory, 2009. “Montylingua Website.” http://web.media.mit.edu/ hugo/montylingua/index.html, Retrieved April.

“OpenNLP Tools Homepage.” 2010. http://opennlp.sourceforge.net/

O’Rorke, P. and A. Ortony. 1994. “Explaining Emotions.” Cognitive Science, 18: 283–323.

Ortony, A. and T. J. Turner. 1990. “What’s Basic about Basic Emotions.” Psychological Review, 97(3): 315–331.

Pang, B. and L. Lee. 2008. “Opinion Mining and Sentiment Analysis.” Foundations and Trends in Information Retrieval, 2(1-2): 1–135.

Pennebaker, J.W., M.E. Francis, and R. J. Booth. 2001. “Linguistic Inquiry and Word Count (LIWC): LIWC2001.” Mahwah: Lawrence Erlbaum Associates.

Princeton Computer Science Laboratory, 2009. “Wordnet Website.” http://wordnet.princeton.edu/, Retrieved April.

Qu, W. and K. Shirai. 2003. “Belief Learning in Certainty Factor Model and Its Application to Text Categorization.” ICICS-PCM 2003, Singapore, 15-18 December, pp. 1192–1196.

Ratinov, L. and D. Roth. 2009. “Design Challenges and Misconceptions in Named Entity Recognition.” Proceedings of CoNLL 2009. (Illinois Named Entity Tagger retrieved at http://cogcomp.cs.illinois.edu/page/software_view/4)

Romeo, F. and A. Sangiovanni-Vincentelli. 1991. “A Theoretical Framework For Simulated Annealing.” Algorithmica, 6: 302–345.

Sani, F. 2005. “When Subgroups Secede: Extending and Refining the Social Psychological Model of Schism in Groups.” Personality and Social Psychology Bulletin, 31(8): 1074–1086.

Solka, J. L. 2008. “Text Data Mining: Theory and Methods.” Statistical Surveys, 2: 94–112.

Strapparava, C. and R. Mihalcea. 2008. “Learning to Identify Emotions in Text.” Proceedings of SAC’08, Fortaleza, Brazil: ACM, March 16–20.

Strapparava, C. and A. Valitutti. 2004. “Wordnet-Affect: an Affective Extension of Wordnet.” Proceedings of LREC, Vol. 4.

Tao, J. and T. Tan. 2005. “Affective Computing.” ACII 2005, LNCS 3784, Springer–Verlag Berlin Heidelberg, pp. 981–995.

Tong, H., C. Faloutsos, and J.-Y. Pan. 2008. “Random Walk with Restart: Fast Solutions and Applications.” Knowl Inf Syst, 14:327-346. DOI 10.1007/s1015-007-0094-2.

Tsadiras, A. K. and K. G. Margaritas. 1998. “The MYCIN Certainty Factor Handling Function as Uninorm Operator and its Use as a Threshold Function in Artificial Neurons.” Fuzzy Sets and Systems, 93: 263–274.

Valitutti, A., C. Strapparava, and O. Stock. 2004. “Developing Affective Lexical Resources.” PsychNology Journal, 2(1): 61–83.

Walter, F. and H. Bruch. 2008. “The Positive Group Affect Spiral: a Dynamic Model of the Emergence of Positive Affective Similarity in Work Groups.” Journal of Organizational Behavior, 29: 239–261.

Wang, P. P. and D.-S. Chen. 1996. “Continuous Optimization by a Variant of Simulated Annealing.” Computational Optimization and Applications, 6: 59–71.

Wiebe, J., T. Wilson, R. Bruce, M. Bell, and M. Martin. 2004. “Learning Subjective Language.” Computational Linguistics, 30(3): 277–308.

Wiebe, J., T. Wilson, and C. Cardie. 2005. “Annotating Expressions of Opinions and Emotions in Language.” Language Resources and Evaluation (formerly Computers and the Humanities), 39(2–3): 165–210.

ACKNOWLEDGMENTS

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U. S. Department of Energy under Contract No. DE-AC05-00OR22725.

APPENDIX A. MATRIX NORMALIZATION OF G AND G^T TO ACCOUNT FOR FRACTIONAL OUTLET EDGE WEIGHTS

APPENDIX A. MATRIX NORMALIZATION OF G AND G^{T} TO ACCOUNT FOR FRACTIONAL OUTLET EDGE WEIGHTS

The adjacency matrices must be row-normalized for the random walk with restart algorithm. Typically row values sum to unity after normalization, except in case (1) where there are no outlinks from a node. For the case where all edges have a value of 0 or 1, normalization is a simple procedure where each weight is divided by the sum of the row. However, owing to modifications to edge weights due to affect terms, entity terms, and punctuation, some adjacency matrix values are fractions in the interval [0,1]. Fractional values increase the complexity of the normalization process. There are four cases we must consider during normalization.

Case 1: *The values of the entire row **i** are zero and node **i** is a terminal node*. All normalized values are set to zero. The *rwr* function recognizes terminal nodes and handles them appropriately. [First condition in Eq. (A.1).]

Case 2: *An edge weight is less than 1*. Fractional weights are set to a value equal to the normalized value that would be obtained if all row entries greater than zero were 1, followed by a multiplication of the result by the fractional value itself. This procedure is a rescaling of the traditional normalization. [Third condition in Eq. (A.1).]

Case 3: *Unity edge weights*. Unity weights are incremented beyond their traditional normalized values to absorb the probability mass freed up by downsizing fractional weights in the row. [Fourth condition in Fig. (A.1).]

Case 4: *Zero self-referential weight when only fractional weights exist in the row*. A positive value in the self-referential loop absorbs the loss of probability mass obtained by setting fractional weights values. [Second condition in Eq. (A.1).]

The following algorithm was developed to complete normalization of the adjacency matrix while accounting for all four cases described above.

ADJACENCY MATRIX NORMALIZATION

Do for each row (i) for $\mathbf{G}$ and $\mathbf{G}^T$:

1. Calculate the sum of all weights in the row $p_i = \sum_j w_{ij}$
2. Calculate the number of non-zero weights in the row $q_i = \sum_j \{w_{ij} > 0\}$. If $q_i = 0$, then node *i* is a terminal node.
3. Calculate the number of weights in the row equal to one $r_i = \sum_j \{w_{ij} = 1\}$
4. Normalize as follows:

$$w'_{ij} = \begin{cases} 0 & if\ q_i = 0 \\ (q_i - p_i)/q_i & if\ w_{ij} = 0\ \&\ r_i = 0\ \&\ j = i \\ w_{ij}/q_i & if\ w_{ij} < 1 \\ \left[w_{ij} + \dfrac{(q_i - p_i)}{r_i} \right] \Big/ q_i & if\ w_{ij} = 1 \end{cases} \tag{A.1}$$

APPENDIX B.
PSYCHOMETRIC SCALING OF QUALIFIER TERMS

APPENDIX B. PSYCHOMETRIC SCALING OF QUALIFIER TERMS

Qualifiers modify the intensity of an affect term in the local context of a document. A list of 142 common qualifier terms was assembled and scaled to specific intensity values in the range (-1, 1) using the function *k*. Reducers were assigned negative qualifier values by *k*, whereas enhancers had positive qualifier values. The scaling procedure is described as follows. First, we summarized qualifiers into six categories: (1) quantity; (2) temporal; (3) emphasis; (4) sureness; (5) size; and (6) intent. The qualifier terms in each category were sorted into reducers and enhancers. For example, the quantity category contained the terms *abundant, all, extra, few, less, little, many, minimal, more, much, only, scarcely* and *some*. The enhancers were identified as *abundant, all, extra, many, more, much* and *only*. The reducer set was *few, less, little, minimal, scarcely* and *some*. Next both reducers and enhancers from each category were rank-ordered by intensity from neutral to extreme; ties were permitted. The qualifier term values were evenly distributed across the range (-1,0) for reducers and (0,1) for enhancers. The first step from zero is a half-step such that whole steps are equal to $s^{+} = 2/(2n+1)$ where *n* is the total number of enhancers in a category, and $s^{-} = 2/(2m+1)$ where *m* equals the number of reducers in a category. Thus, the set of enhancers is $\left\{\frac{1}{2n+1}, \frac{3}{2n+1}, \ldots, \frac{2n-1}{2n+1}\right\}$; similarly, the reducer set is $\left\{-\frac{1}{2m+1}, -\frac{3}{2m+1}, \ldots, -\frac{2m-1}{2m+1}\right\}$. Figure B1 illustrates the distribution of intensity scores over the entire (-1,1) range.

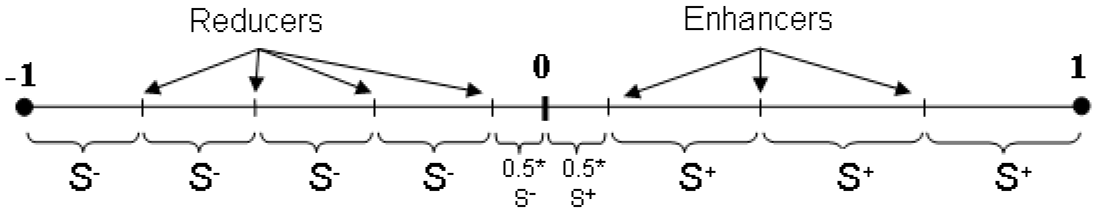


Fig. B1. Qualifier terms scaled in the (-1,1) interval.

APPENDIX C.
AFFECT-ENTITY RELATIONSHIP DIAGRAMS

APPENDIX C. AFFECT-ENTITY RELATIONSHIP DIAGRAMS

This appendix contains diagrammatic evidence for affective relationships from the algorithm-generated solutions along with the benchmarks. A complete set of diagrams for MPQA document 20.24.49-8480 is shown in Figs. 13a–13d in Sect. 6. Full diagrammatic contents are reproduced here for three additional MPQA documents: 21.07.24-24231 (Figs. C1–C4), 21.50.57-15245 (Figs. C5–C8) and 23.18.15-25073 (Figs. C9–C12). These examples display properties similar to those discussed in Section 5: (1) many benchmarked affective relationships discovered; (2) many affective relationships discovered not documented in the benchmark diagram; (3) symmetric affective relationships; and (4) over-diffusive spreading of affect through the documents.

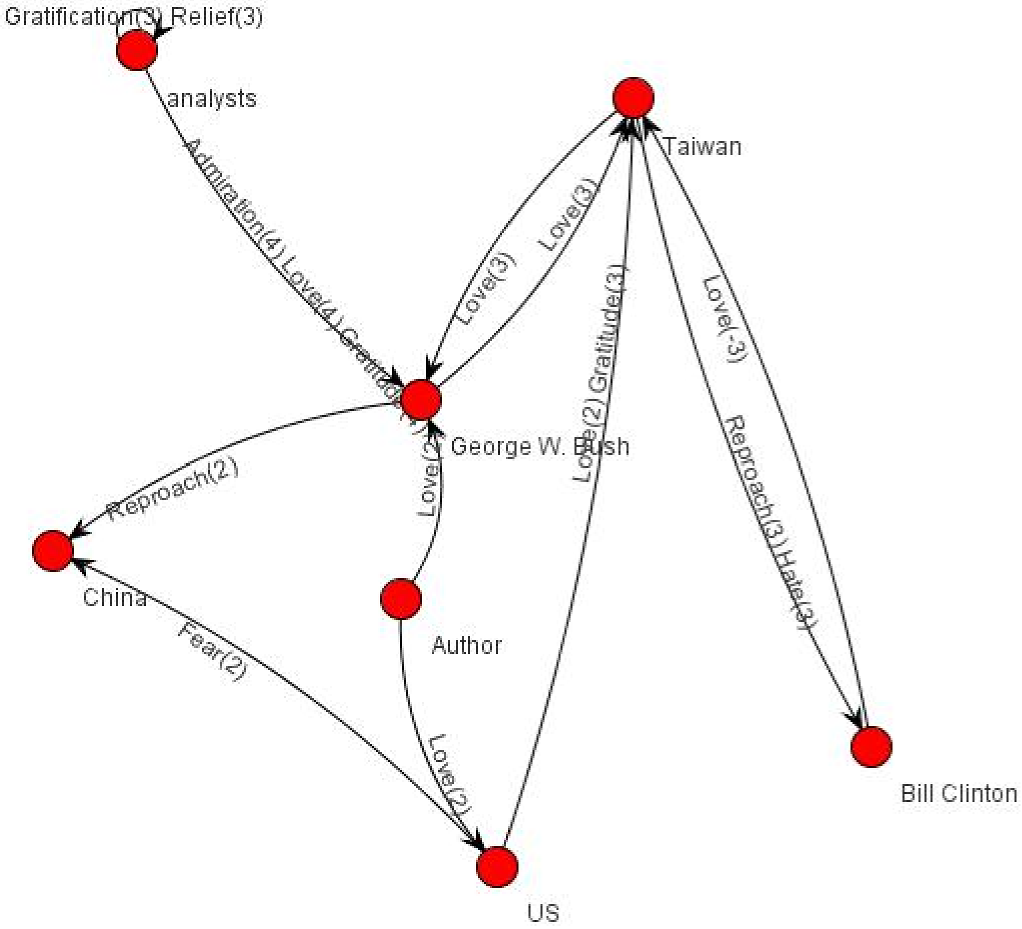


Fig. C1. Benchmark affect-entity diagram for document 21.07.24-24231.

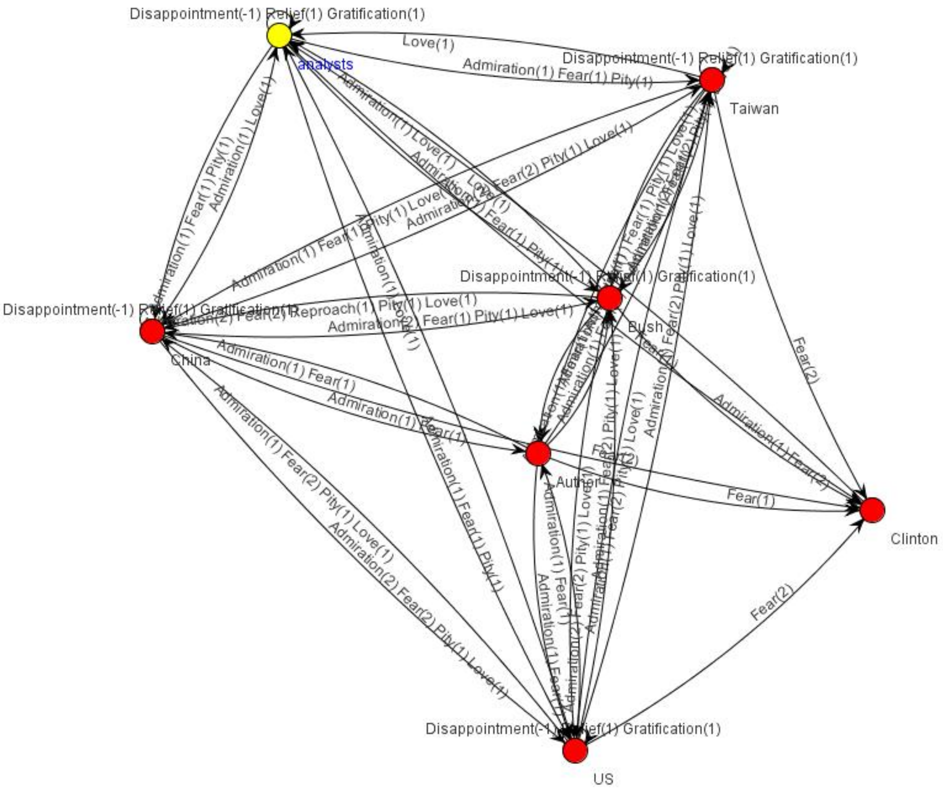

Disappointment(-1) Relief(1) Gratification(1)
analysts
Love(1)
Admiration(1) Fear(1) Pity(1)
Disappointment(-1) Relief(1) Gratification(1)
Taiwan
Admiration(1) Love(1)
Admiration(1) Fear(1) Pity(1)
Admiration(1) Love(1)
Fear(2) Pity(1) Love(1)
Admiration(1) Fear(1)
Disappointment(-1) Relief(1) Gratification(1)
Bush
Disappointment(-1) Relief(1) Gratification(1)
Admiration(2) Fear(2) Reproach(1) Pity(1) Love(1)
Admiration(2) Fear(1) Pity(1) Love(1)
China
Admiration(1) Fear(1)
Admiration(1) Fear(1)
Fear(2)
Admiration(1) Fear(2)
Auther
Fear(1)
Clinton
Admiration(1) Fear(2) Pity(1) Love(1)
Admiration(2) Fear(2) Pity(1) Love(1)
Admiration(1) Fear(1) Pity(1)
Admiration(1) Fear(1)
Fear(2) Pity(1) Love(1)
Fear(2)
Disappointment(-1) Relief(1) Gratification(1)
US

Fig. C2. Affect-entity diagram generated by highest-fitness affect propagation algorithm (1379 relations) for document 21.07.24-24231.

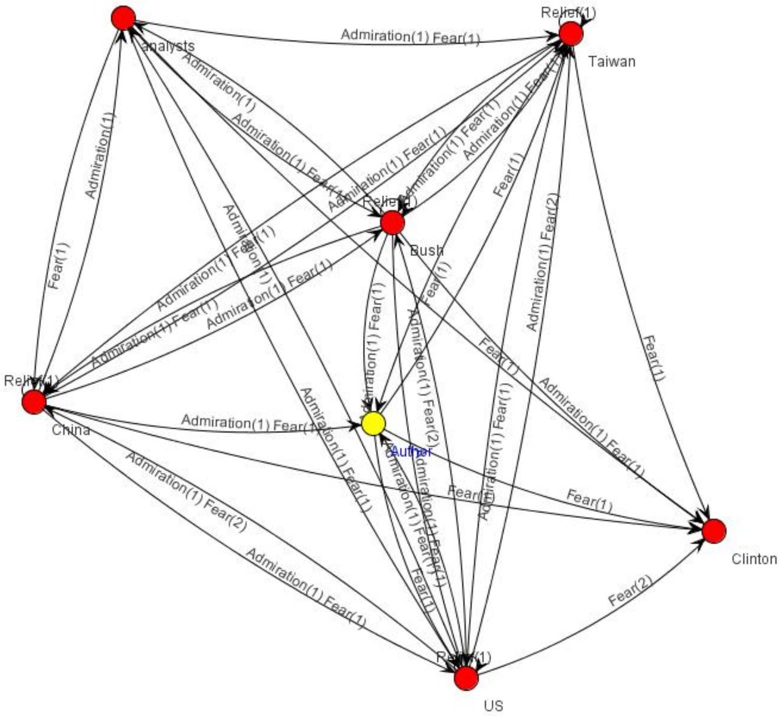


Fig. C3. Affect-entity diagram generated by affect propagation algorithm (746 relations) for document 21.07.24-24231.

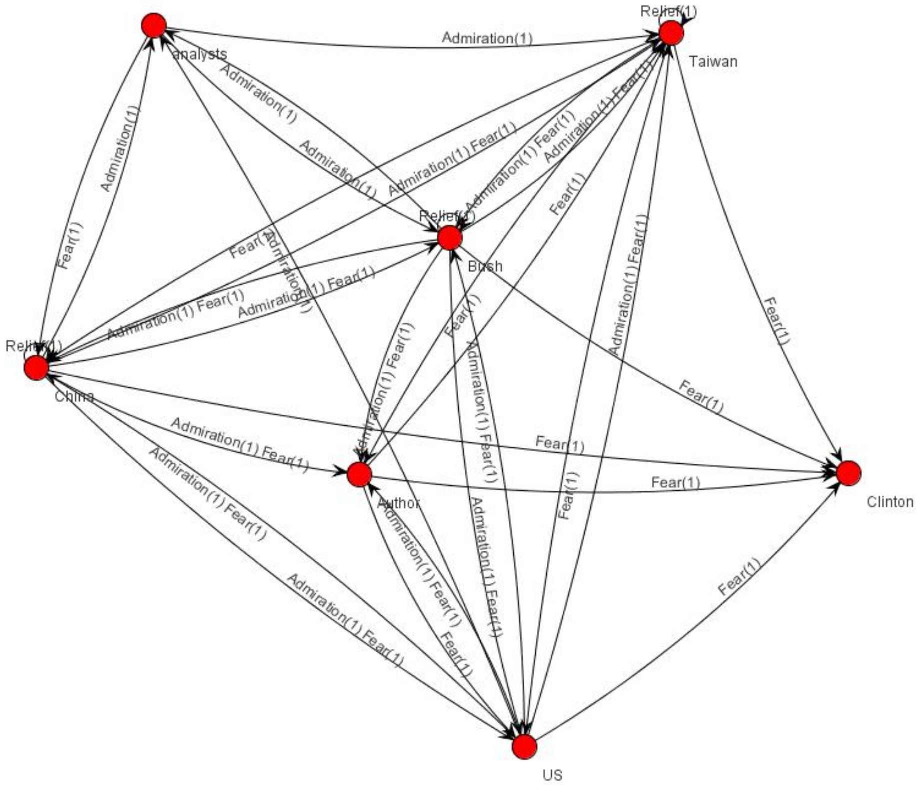


Fig. C4. Affect-entity diagram generated by affect propagation algorithm (581 relations) for document 21.07.24-24231.

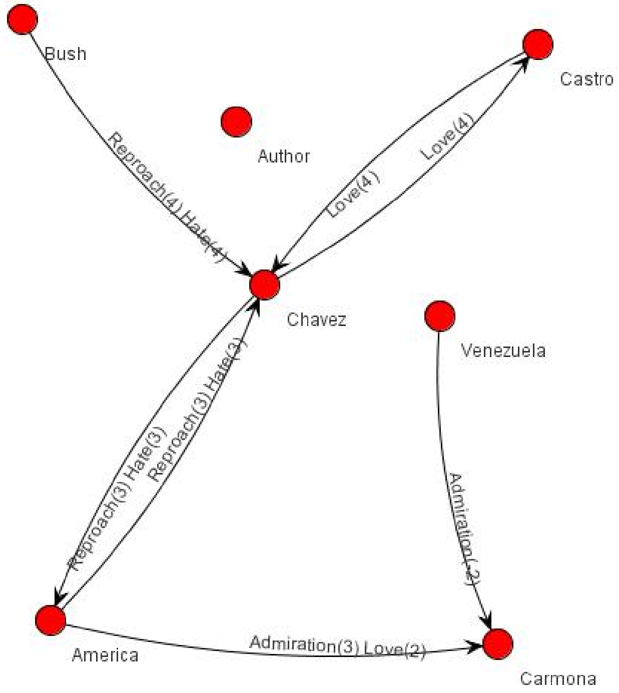


Fig. C5. Benchmark affect-entity diagram for document 21.50.57-15245.

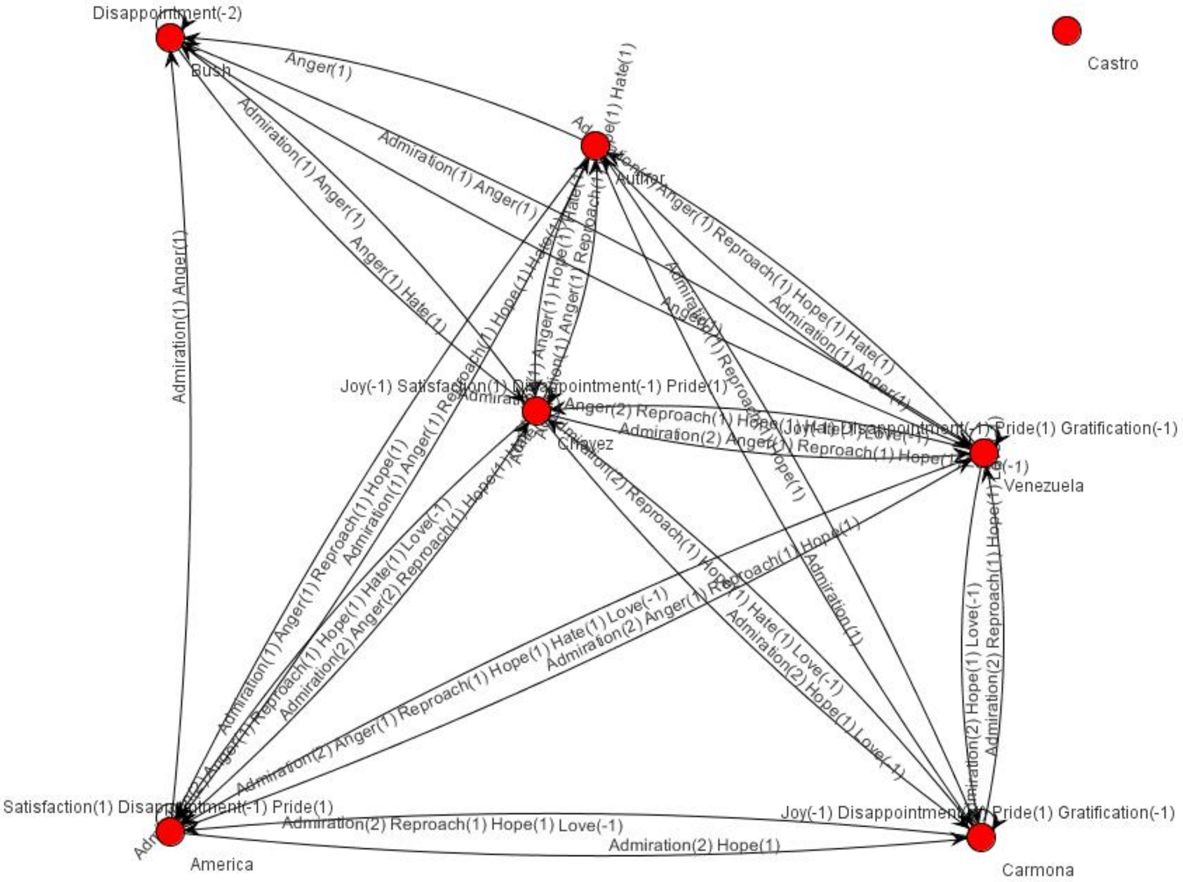

Disappointment(-2)
Bush
Castro
Author
Chavez
Venezuela
America
Carmona
Anger(1)
Admiration(1) Anger(1)
Admiration(1) Anger(1)
Anger(1) Hate(1)
Admiration(1) Anger(1)
Hope(1) Hate(1)
Anger(1) Reproach(1) Hope(1) Hate(1)
Admiration(1) Anger(1)
Admiration(1)
Anger(1) Reproach(1) Hope(1)
Anger(1) Hope(1) Hate(1)
Admiration(1) Anger(1) Reproach(1)
Joy(-1) Satisfaction(1) Disappointment(-1) Pride(1)
Anger(2) Reproach(1) Hope(1) Hate(1) Love(-1)
Admiration(2) Anger(1) Reproach(1) Hope(1)
Joy(-1) Disappointment(-1) Pride(1) Gratification(-1)
Love(-1)
Admiration(1) Anger(1) Reproach(1) Hope(1)
Admiration(1) Anger(1) Reproach(1) Hope(1) Hate(1)
Anger(1) Reproach(1) Hope(1) Hate(1) Love(-1)
Admiration(2) Anger(2) Reproach(1) Hope(1)
Admiration(2) Anger(1) Reproach(1) Hope(1) Hate(1) Love(-1)
Admiration(2) Anger(1) Reproach(1) Hope(1)
Admiration(2) Reproach(1) Hope(1) Hate(1) Love(-1)
Admiration(2) Hope(1) Love(-1)
Admiration(1)
Admiration(2) Hope(1) Love(-1)
Admiration(2) Reproach(1) Hope(1) Love(-1)
Satisfaction(1) Disappointment(-1) Pride(1)
Admiration(2) Reproach(1) Hope(1) Love(-1)
Admiration(2) Hope(1)
Joy(-1) Disappointment(-1) Pride(1) Gratification(-1)

Fig. C6. Affect-entity diagram generated by highest-fitness affect propagation algorithm (1379 relations) for document 21.50.57-15245.

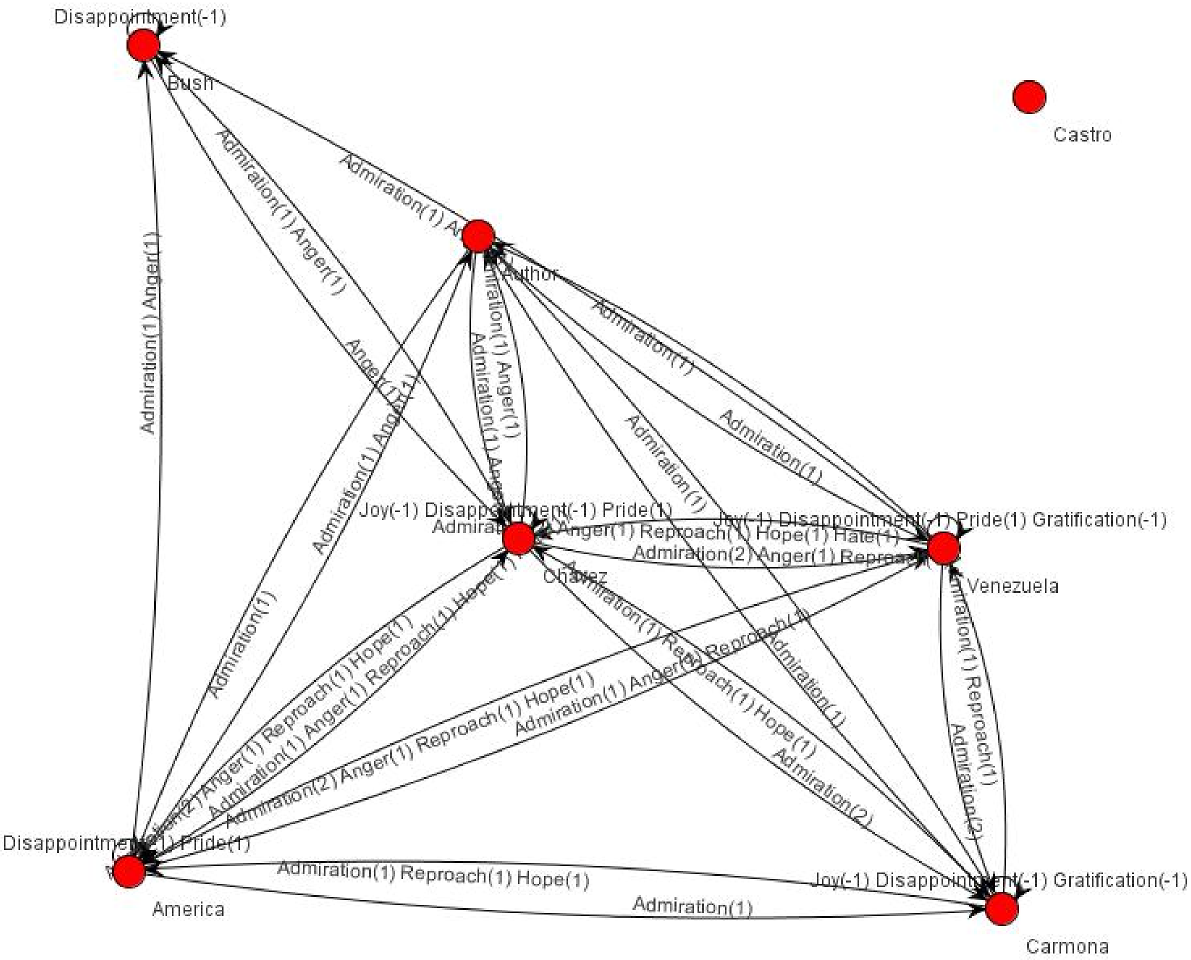

Disappointment(-1)
Bush
Castro
Admiration(1) Anger(1)
Admiration(1) Anger(1)
Admiration(1) A
Author
Admiration(1) Anger(1)
Admiration(1) Anger(1)
Anger(1)
Admiration(1) Anger(1)
Admiration(1)
Admiration(1)
Admiration(1)
Joy(-1) Disappointment(-1) Pride(1)
Admira
Chavez
Anger(1) Reproach(1) Hope(1) Hate(1)
Admiration(2) Anger(1) Reproach
Joy(-1) Disappointment(-1) Pride(1) Gratification(-1)
Venezuela
Admiration(1)
Admiration(1) Anger(1) Reproach(1) Hope(1)
Admiration(1) Anger(1) Reproach(1) Hope(1)
Admiration(2) Anger(1) Reproach(1) Hope(1)
Admiration(1) Anger(1) Reproach(1)
Admiration(1) Reproach(1) Hope(1)
Admiration(1)
Admiration(2)
Admiration(1) Reproach(1)
Admiration(2)
Disappointment(-1) Pride(1)
America
Admiration(1) Reproach(1) Hope(1)
Admiration(1)
Joy(-1) Disappointment(-1) Gratification(-1)
Carmona

Fig. C7. Affect-entity diagram generated by affect propagation algorithm (746 relations) for document 21.50.57-15245.

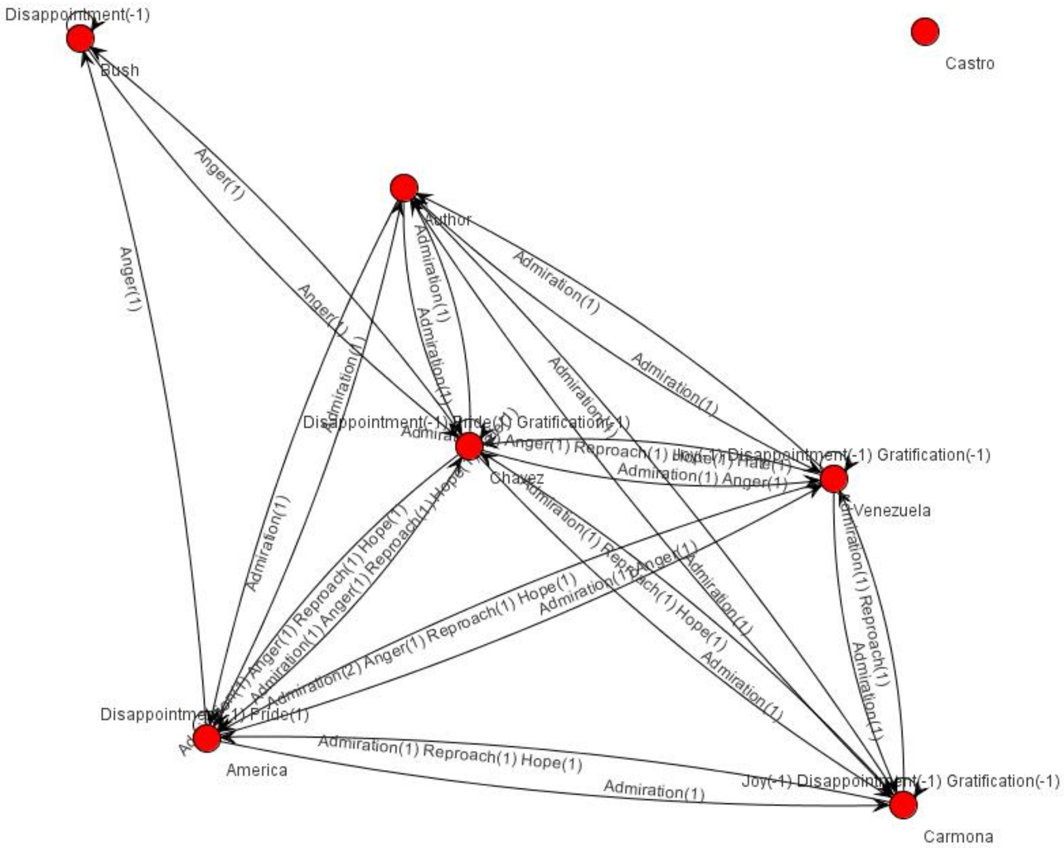

Disappointment(-1)
Bush
Castro
Anger(1)
Anger(1)
Author
Admiration(1)
Admiration(1)
Admiration(1)
Admiration(1)
Admiration(1)
Anger(1)
Admiration(1)
Disappointment(-1) Pride(1) Gratification(-1)
Admira
Anger(1) Reproach(1) Joy(-1) Disappointment(-1) Gratification(-1)
Hope(1) Hate(1)
Admiration(1) Anger(1)
Chavez
Venezuela
Admiration(1)
Anger(1) Reproach(1) Hope(1)
Admiration(1) Anger(1) Reproach(1) Hope(1)
Admiration(1) Reproach(1) Hope(1)
Anger(1)
Admiration(1)
Admiration(2) Anger(1) Reproach(1) Hope(1)
Admiration(1)
Admiration(1) Reproach(1)
Admiration(1)
Disappointment(-1) Pride(1)
America
Admiration(1) Reproach(1) Hope(1)
Admiration(1)
Joy(-1) Disappointment(-1) Gratification(-1)
Carmona

Fig. C8. Affect-entity diagram generated by affect propagation algorithm (581 relations) for document 21.50.57-15245.

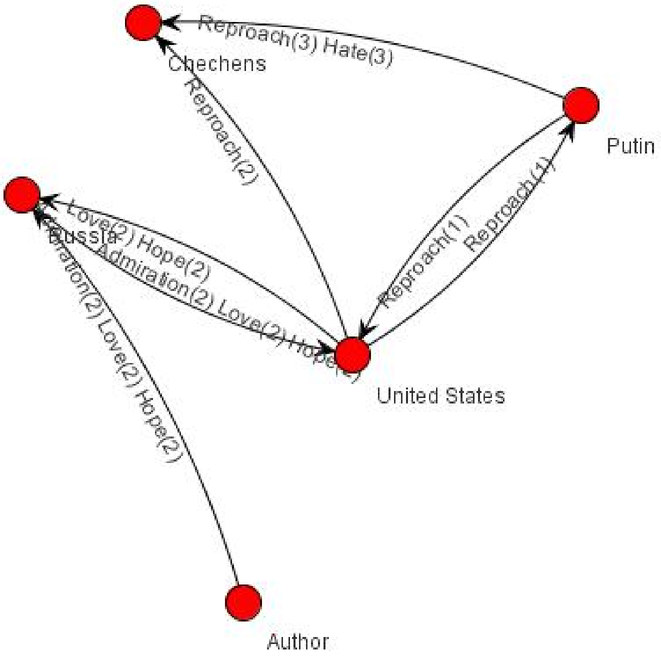


Fig. C9. Benchmark affect-entity diagram for document 23.18.15-25073.

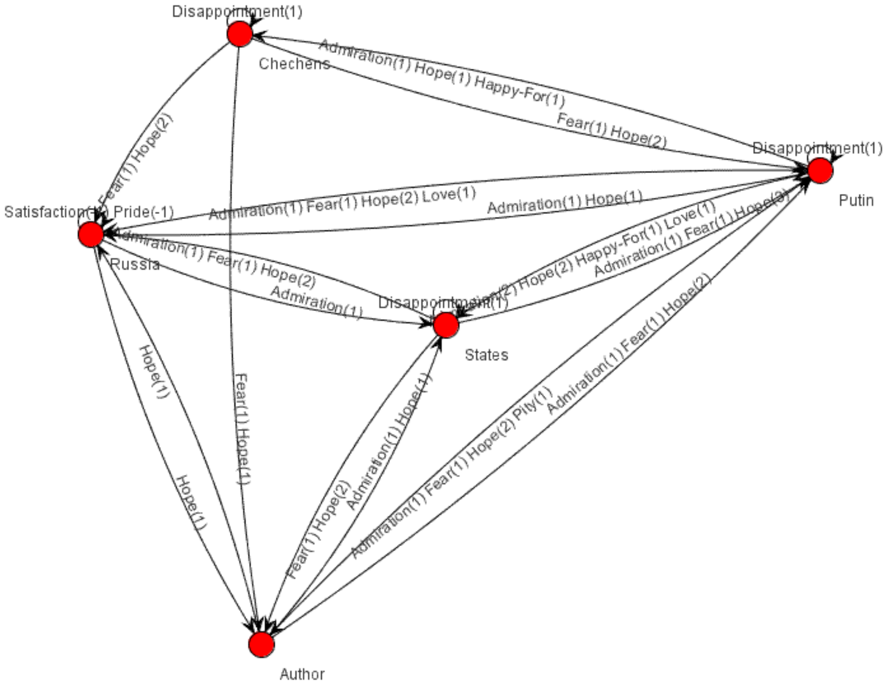

Disappointment(1)
Chechens
Admiration(1) Hope(1) Happy-For(1)
Fear(1) Hope(2)
Disappointment(1)
Putin
Fear(1) Hope(2)
Satisfaction(1) Pride(-1)
Admiration(1) Fear(1) Hope(2) Love(1)
Admiration(1) Hope(1)
Russia
Admiration(1) Fear(1) Hope(2)
Admiration(1)
Hope(2) Happy-For(1) Love(1)
Admiration(1) Fear(1) Hope(3)
Disappointment(1)
States
Hope(1)
Hope(1)
Fear(1) Hope(1)
Fear(1) Hope(2)
Admiration(1) Hope(1)
Admiration(1) Fear(1) Hope(2) Pity(1)
Admiration(1) Fear(1) Hope(2)
Author

Fig. C10. Affect-entity diagram generated by highest-fitness affect propagation algorithm (1379 relations) for document 23.18.15-25073.

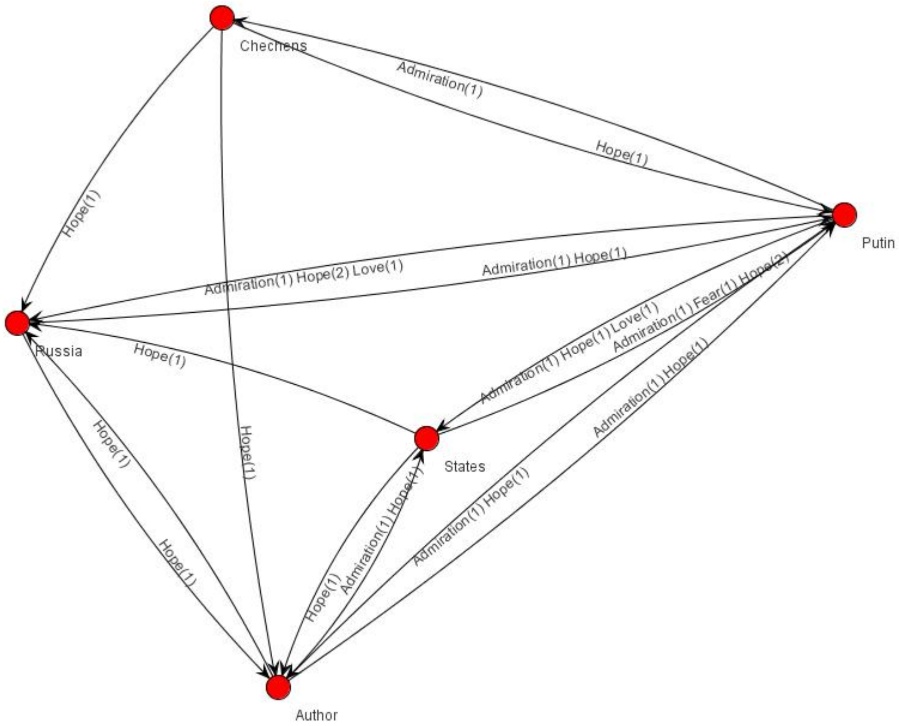


Fig. C11. Affect-entity diagram generated by affect propagation algorithm (746 relations) for document 23.18.15-25073.

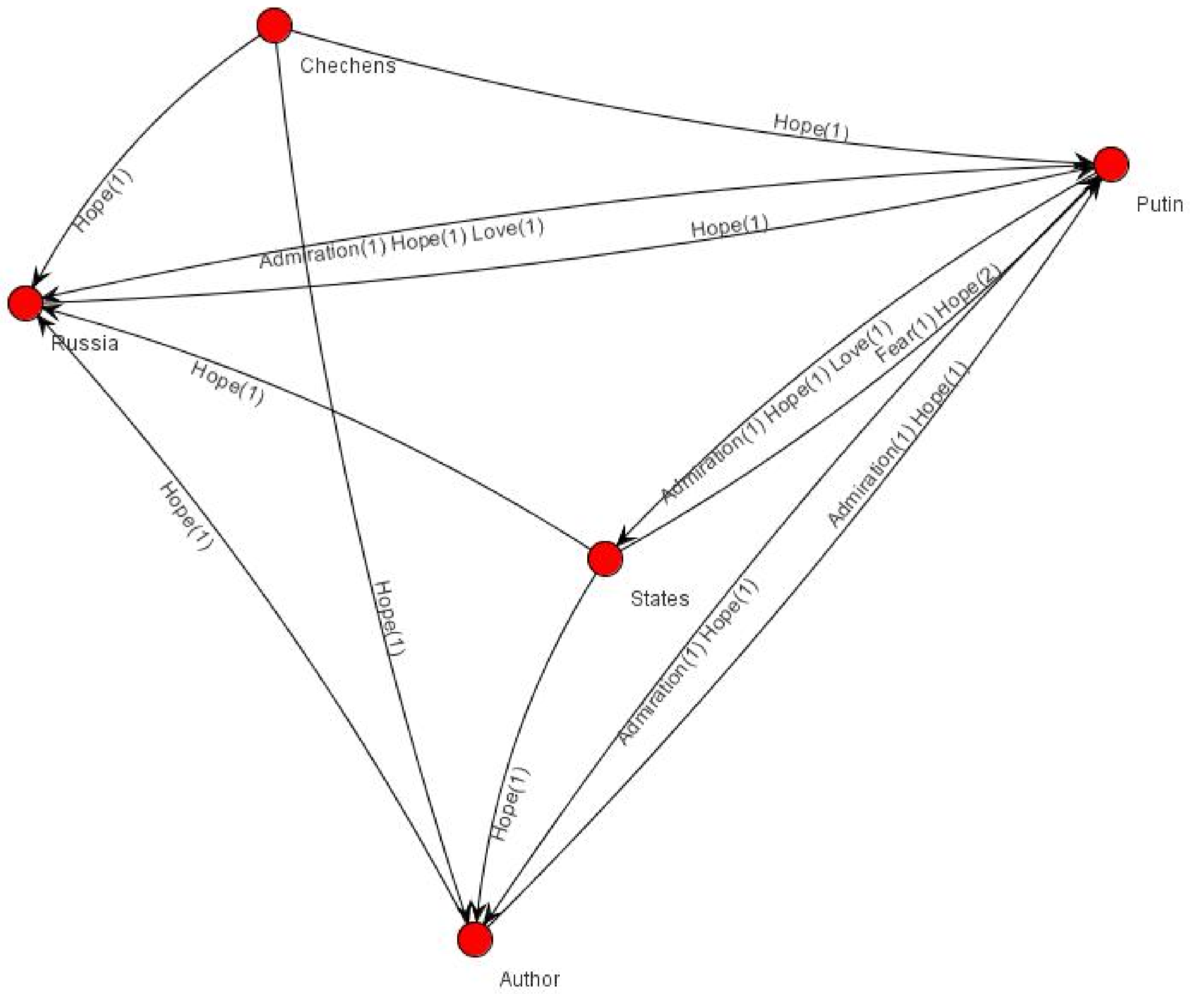


Fig. C12. Affect-entity diagram generated by affect propagation algorithm (581 relations) for document 23.18.15-25073.