SemanticEmo
– technical report –

Iulian GORIAC
Faculty of Computer Science, A.I.Cuza University of Iaşi,
Berthelot 16, Iaşi, Romania

iulian.goriac@info.uaic.ro[1]

Abstract. An enormous amount of data is stored in textual form. For thousands of years written documents were used to store the experience of a whole world. It is only recently that we began to use computers to try to extract that information and shape it in a form that will allow its automatic processing. There is also a tendency towards designing human-computer interfaces that are more and friendlier towards the former. Emotions, difficult as they may be to define and categorize, are a very important element when describing human communication and therefore it is essential to find a way to deal with them whether we want the machines to be able to help us use our heritage or to communicate with them in a natural way. **SemanticEmo** is a project that focuses on mining textual data related to a certain item of interest in order to provide the user with an emotional map of the context in which that item appears. Given the relatively large amount of data that is to be processed for each search, a very simple detection algorithm is used, based on the identification of some characteristic word patterns, algorithm proposed by Geneva Emotion Research Group. An attempt was made to map the affect labels into the emotion wheel slots in such a way that at the end of every search the user will be provided with a certain emotional map of the context of the item of interest. System analysis facilities like EmotionML markup and HTML visualization are also provided. This project is more like a proof of concept related to quick emotion detection a more 'deep semantic' approach being planned for the future.[2]

Keywords: semantic web, affective state, natural language processing (NLP), emotion detection, Geneva Affect Label Coder (GALC), Geneva Emotion Wheel (GEW), EmotionML

Contents

1 Alternative iulian.goriac@google.com

2 The project web page can be found at: http://code.google.com/p/semanticemo

Introduction

For thousands of years the experience of the previous generations was preserved and transmitted by using the written word. There is a huge amount of knowledge contained in books, magazines, articles, etc. and more is added in every second. Not only will it be a fun thing to do to be able to automatically process the 'deep meaning' of those documents, it will soon become a necessity as the human mind does not evolve as fast as it would be needed to adapt to the continuously growing informational ocean that we live in.

There is also a tendency towards designing computer user interfaces that are more and more friendly. Currently we are a long way from hardwiring machine code instructions, yet a significant learning effort is still required before one can actually interact with a computer. Probably the ideal interface will allow the user to communicate with a machine using any way of signaling (gestures, mimic, voice tone, natural language, brainwaves etc.) and, even if some achievements have already been made in any of the directions above, we still have to go a long way before all of them will be integrated in an elegant and unitary device that will be able to simplify the human-machine interaction to the point where virtually no specific learning will be required on the human side.

When we interact with the world around us we process the information at at least two levels: the cognitive level and the affective level. The first allows us to highly differentiate between the many situations we encounter; the later allow us to quickly synthesize a very wide range of data into only one bottom line: the emotion. What emotions are is still a well known debate topic that is still far from being exhausted yet it is also widely accepted that they influence our lives in the most intimate and significant ways. Given their importance in human life it is obvious that

This material will not try to deal with the emotion theory (see [1], [2], [4] and [6] for details) but will consider the developing an actual system capable to recognize emotions from textual data written by human users. Before designing the ultimate user interface some smaller steps need to be completed, steps like emotion detection, particularly in textual data provided by random users.

Idea

Develop a Web application that will allow the detection of the emotions related to a certain item (e.g. product, service, event, person) by analyzing the posts from different social platforms (e.g. Amazon, Facebook, Revyu, StumbleUpon, Twitter etc.). The tool should be able to offer the facility of (automatically) evaluate and classify different entities emotion based. Web links will be provided to exemplify every emotional category related to the specified item.

The original requirements (in Romanian can be found at http://profs.info.uaic.ro/~busaco/teach/courses/wade/web-projects.html see SemanticEmo).

Detecting emotions

Without getting into too much detail related to emotion theory, in this paper will only be presented some ideas relevant to the current implementation. The main theoretical resource for this chapter is [6].

Affective states

Affective states are personal experiences expressing the correspondence or discrepancy degree between an object or a situation and our tendencies [11].

In [6] there are six affective states or phenomena described: preferences, attitudes, moods, affect dispositions, interpersonal stances and, of course, emotions. The distinction between those is based on an eight coordinate system containing:

- *event focus* – how connected is a certain affective state to the event that triggered it
- appraisal driven – the relevance to the major concerns of the organism
    - *intrinsic appraisal* – evaluates the feature of an object or person independently of the current needs and goals of the appraiser, based on genetic (e.g. sweet taste) or learned (e.g. bittersweet food) preferences
    - *transactional appraisal* – evaluates events and their consequences with respect to their conduciveness for salient needs, desires, or goals of the appraiser
- *synchronization* – the degree to which the affective state is capable of massive organism resource mobilization and coordination
- *rapidity of change* – how likely it is for an affective state to change according to the changes in the appraisal; very much like some sort of inertia
- *behavioral impact* – the level of impact on the motor expression and on communication
- *intensity* – the intensity of the response pattern and the corresponding subjective experience related to the affective state
- *duration* – how long the mobilization of the resources will last

Emotions

According to [6] an emotion is an episode of interrelated, synchronized changes in the states of all or most of the five organismic subsystems in response to the evaluation of an external or internal stimulus event as relevant to major concerns of the organism.

Organismic subsystem and major substrata	*Emotion function*	*Emotion component*
Information processing (CNS)	Evaluation of objects and events	Cognitive component (appraisal)
Support (CNS, NES, ANS)	System regulation	Neurophysiological component (bodily symptoms)
Executive (CNS)	Preparation and direction of action	Motivational component (action tendencies)
Action (SNS)	Communication of reaction and behavioral intention	Motor expression component (facial and vocal expression)
Monitor (CNS)	Monitoring of internal state and organismenvironment interaction	Subjective feeling component (emotional experience)

Table : Relationships between organismic subsystems and the functions and components of emotion ***(from [6])*** **(CNS - Central Nervous System; NES - neuro-endocrine system; ANS - autonomic nervous system; SNS - somatic nervous system)**

Relative to the affective state characterization Scherer in [6] distinguishes between utilitarian emotions and aesthetic emotions. Of our concern here are the utilitarian emotions. They have a very high event focus, a medium intrinsic appraisal and a very high transactional appraisal, very high synchronization, rapidity of change and behavioral impact, a high intensity and a low duration.

Things to consider when trying to detect emotions

By compiling the approaches described in the bibliographical references I present here a series of steps that an algorithm capable to identify the emotions related to a certain item might include:

1. gather the textual data related to the desired topic
2. split the text into sentences
3. split every sentence into words
4. identify the words that can hint to a certain emotion by using a dedicated dictionary; a simple tool to do that is Geneva Affect Label Coder (GALC) that can be found in [5]; more complex approaches [3], [9], [10] consider emotional

dimensions of every word as specified in WordNet-Affect[3], Whissell's Dictionary of Affect in Language[4], Affective Norms for English Words[5], etc.

5. same as 4 but add a suffix and a prefix analysis [10]
6. identify the specific expressions that people use to indicate a certain emotional state (e.g. feeling blue, happy as a funeral, broken heart, ...)
7. create a full parse of the sentence in order to find the real meaning of every word[6]:
    a. "as he heard the news he was suddenly flooded with relief"
    b. "he asked for troops for the relief of Atlanta"
8. identify the modifiers for the above mentioned constructions and the relations between them that can modulate or even alter the initial meaning; like:
    a. inverse modifiers: not happy -> sad
    b. positive modifiers: very angry -> full of rage
    c. negative modifiers: less fearful -> nervous / relieved
    d. self modifiers: self satisfied -> pride and not just satisfaction
    e. etc.
9. find the specific relations between the affective state hinting structures and the item of interest
10. perform anafora resolution and consider the whole context when deciding the affective states
11. report the findings preferably by using a dedicated markup language

The current implementation, further described, implements the steps 1, 2, 3, 4 (simple approach) and 11 (using EmotionML). Parsing support is included via Stanford Parser.

3 http://wndomains.fbk.eu/wnaffect.html

4 http://www.hdcus.com/manuals/wdalman.pdf

5 http://csea.phhp.ufl.edu/media/anewmessage.html

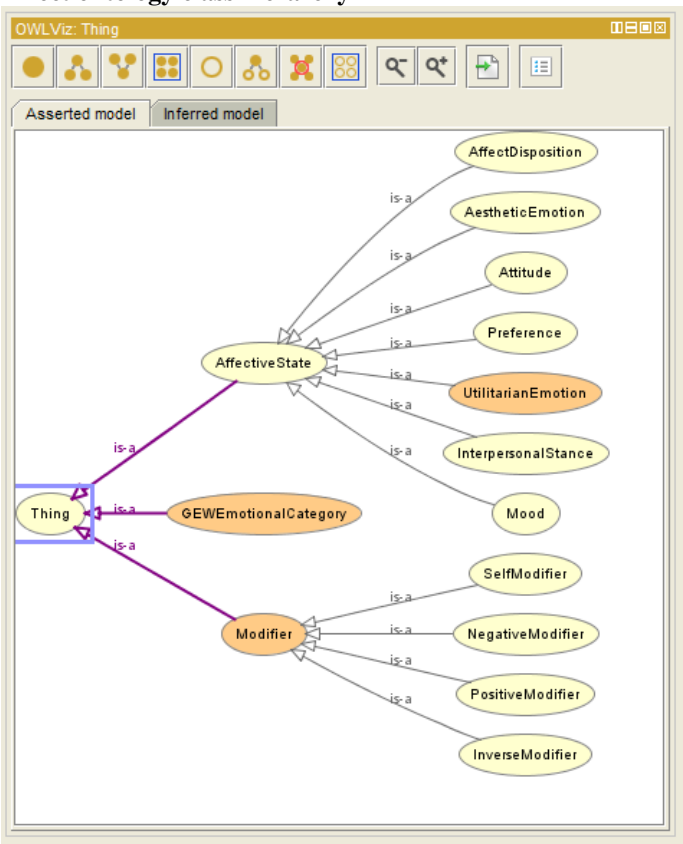
6 Examples are from WordNet definitions (http://wordnet.princeton.edu)

System implementation

Emotion ontology

There is no standard commonly accepted way of classifying emotions. Therefore any representation of knowledge related to this subject should be as flexible as possible and allow for easy refactoring and extension. OWL is an ontology specification language that offers a fair amount of flexibility for relation description and is widely supported by software tools and developers community.

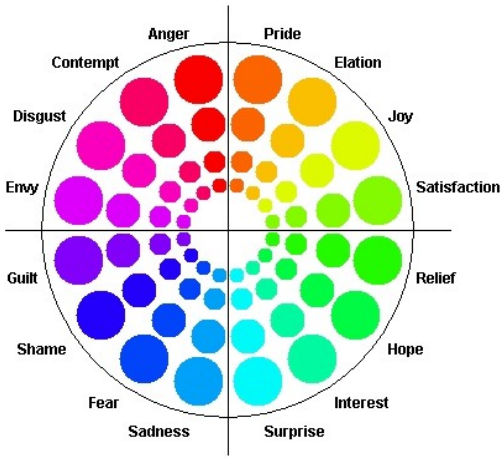
Figure : **Affect ontology class hierarchy**



For our system we used Protégé[7] to develop a very simple emotion ontology based on the work presented in [6] and [7].

7 http://protege.stanford.edu

The base class for describing affective states is **AffectiveState** a direct subclass of **Thing**. Among those, what we are interested in finding are the utilitarian emotions which are divided, according to the Geneva Emotion Wheel, in 16 categories:



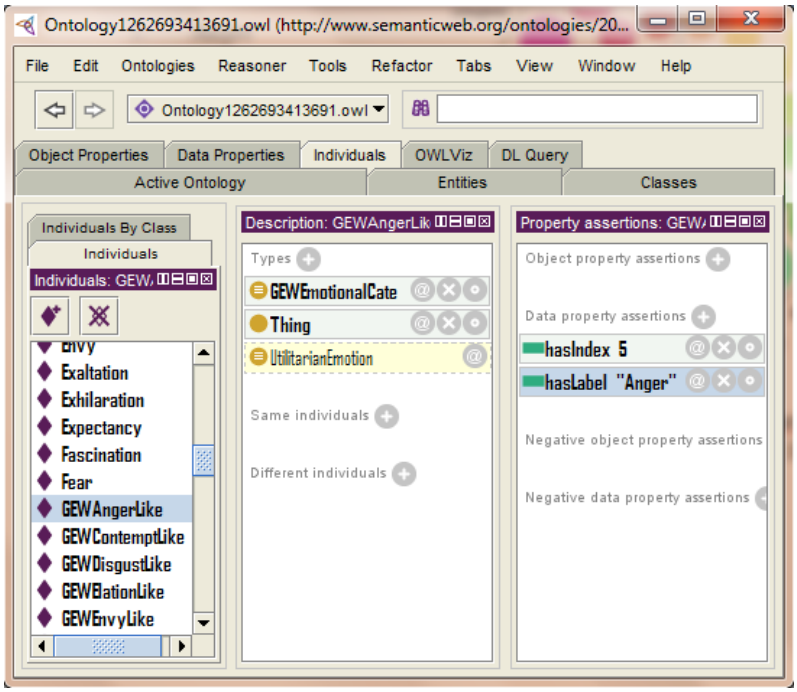
12. Satisfaction
13. Joy
14. Elation
15. Pride
16. Anger
17. Contempt
18. Disgust
19. Envy
20. Guilt
21. Shame
22. Fear
23. Sadness
24. Surprise
25. Interest

Figure : **Geneva Emotion Wheel**

Hope

26. Relief
27.
28.

Figure : **Description and properties of an emotional category**



Each type is represented by an individual of both **UtilitarianEmotion** and **GEWEmotionalCategory** and is name accordingly to the following pattern GEW+CategoryName+Like.

29.

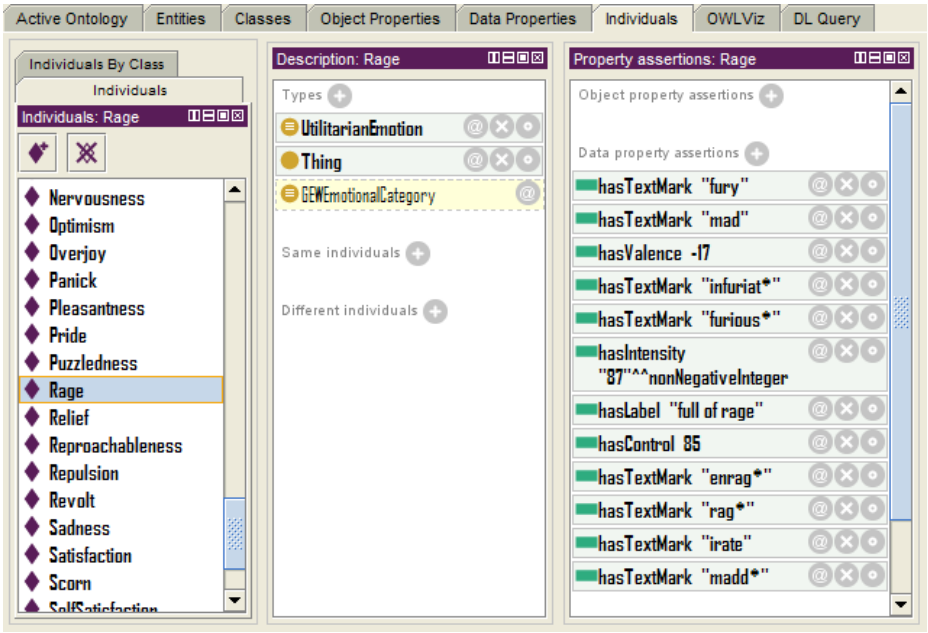
30. As one can see every emotion category has two properties, its integer index used for identification in the application routines and a string label used for the user interface.

31. Every particular emotion is described based on two main attributes [6]:

- the level of perceived control in the situation that generates the emotion (vertical dimension on GEW) (named the control)
- the positive/negative (pleasant/unpleasant) quality of the situation and of the resulting feeling (horizontal dimension on GEW) (named the valence)

Every category has four levels of intensity considered, each of them identified by an emotion name (the names for the 64 emotions were taken from [7]). Every emotion is has certain properties:

Figure : **The properties of an emotion**



the label: a string holding the name of the emotion

— the intensity: an integer ranging from 0 (no emotion) to 100 (maximum intensity)

— the valence: an integer ranging from -100 (very negative emotion) to 100 (very positive emotion)

— the control: an integer ranging from -100 (the participant feels completely out of control) to 100 (the participant feels completely empowered)

— a set of text patterns used to spot the emotions in the text. The patterns are very simple: either complete words (‘rage’) or words starting with a certain string (‘furious*’)

The ontology knowledge is made accessible to the application code via instances of the **EmoOntology** class. For reading the OWL format specified data the Jena[8] semantic web framework is used.

Data acquisition

One way to find the peoples’ opinion about a certain topic like a product, a service, an event or a certain person is to read online posts related to that item. There are various places where one can find these posts: blogs, sites, forums, mailing lists, social networking sites etc. There are different ways to search the sources but usually the presentation sites offer specific means for that.

The SemanticEmo application provides a common interface that can be used in order to acquire data related to a specific topic:

8 http://jena.sourceforge.net/index.html

```
public interface Harvester {
        // returns a list of articles related to the topic
        public List<Article> harvest(String topic);

        // returns the id of the harvester
        public String getId();

        // returns the name of the harvester (that will be
        // used for the user interface)
        public String getName();

        // used to activate and deactivate the harvester
        public boolean isActive();
        public void setActive(boolean active);
}
```

The **Harvester** interface is implemented for every desired source of data. For instance, the **TwitterHarvester** provides access (via Twitter4J[9] library) to the Twitter search web services but different implementations can be provided for sites like Facebook, Revyu, Amazon or even Bing searches.

All the harvesters are registered in a **EmoSearchEngine** class instance that also contains an **EmoOntology** and an **EmoDetector** (but more on that later).

```
public class EmoSearchEngine {
        List<Harvester> harvesters;
        EmoOntology emoOnt;
        EmoDetector emoDetector;
}
```

For every search all the active harvesters are interrogated and all their responses are compiled in one big list of articles. Every article is processed by the **EmoDetector** instance and the result is compiled into the various responses that will be provided to the user.

The **Article** class is very important for our discussion and therefore the most relevant part of its implementation will be presented here:

```
public class Article implements Comparable<Article> {

        // text labels for emotion intensities
        static String[] intensities = new String[]{
                                "*", "**", "***", "****"};

        // every article is identified by a UUID assigned at
        // the construction of the Article object. The UUID // is used by the web
    interface in order to process // requests related to a certain article (like
        // rendering its emotional markup into HTML
        String id;
```

9 http://twitter4j.org/en/index.html

```
        // the Web address of the article
        String url = null;

        // the search topic (specified by the user)
        String topic;

        // the content of the article; a simple text string
        // that contains only textual information (NO markup)
        String content = "";

        // the XML document containing the emotional markup
        // in EmotionML
        Document emotionMarkup = null;

       // this function is used when performing word based // emotion detection;
    every call adds a new <emotion>
       // XML element
        public void addEmo(
                            Word w, Emotion e, EmoOntology ont);

        // returns an XML representation of the emotional
        // markup in EmotionML
        public String toXmlString();

        // returns an HTML representation of the emotional
        // markup
        public String toHtml();
}
```

Emotion detection

After reading the emotion ontology (see page 7) all the text markers are collected in a tree-like structure that maps every pattern to the corresponding emotion. A one to one between the marker and the emotion is used. The tree-like structure has the property of checking the matching between every word and all the patterns in virtually constant time no matter how many patterns are used. It does that by using a RETE-like algorithm [12]. The word is considered a match if it has exactly the same character sequence as the mark (in the case of a non-star pattern) or if the beginning of the word matches the non-star character sequence of the star patterns. Here is a small part of the matching structure:

Figure : **Tree structure for emotion words pattern matching**[10]

Once collected, every article is processed by identifying the separate words contained in the article. Tokenization is performed by using the algorithms provided

10 The picture was created by using FreeMind
http://freemind.sourceforge.net/wiki/index.php/Main_Page

by the StanfordParser[11] library. Every word is checked against the pattern tree and the appropriate **Emotion** class instance is returned. For every detection, a specific array of 64 sets that stores references to the articles each emotion has been found in, is updated. In addition, every word that hints to a certain emotion is marked by using the EmotionML Markup Language[8]. The markers are not introduced directly into the text because a further extension of the detection algorithm might need to use overlapping markers. Instead it was opted for an external markup that only specifies the emotion attributes and the start and end string positions delimiting the substring expressing the emotion.

For example the following article[12] has three emotions detected:

@********** Sounds fun! Wish I could be there to see you! Enjoy yourslef and I hope you get a big win!

The underlined words are marked by the following markings:

```
<emotionml xmlns="http://www.w3.org/2009/10/emotionml">
  <emotion>
    <category name="Joy" set="gew"/>
    <intensity value="*"/>
    <dimensions set="valenceControl">
      <valence value="0.1"/>
      <control value="0.06"/>
    </dimensions>
    <metadata>
      <word begin="23" end="26" value="fun"/>
    </metadata>
  </emotion>
  <emotion>
    <category name="Joy" set="gew"/>
    <intensity value="***"/>
    <dimensions set="valenceControl">
      <valence value="0.51"/>
      <control value="0.34"/>
    </dimensions>
    <metadata>
      <word begin="62" end="67" value="Enjoy"/>
    </metadata>
  </emotion>
  <emotion>
    <category name="Hope" set="gew"/>
    <intensity value="**"/>
    <dimensions set="valenceControl">
      <valence value="0.31"/>
      <control value="-0.2"/>
    </dimensions>
```

11 http://nlp.stanford.edu/software/lex-parser.shtml

12 A Twitter post (the name of the author has been removed for privacy reasons)

```
      <metadata>
        <word begin="83" end="87" value="hope"/>
      </metadata>
    </emotion>
  </emotionml>
```

Every detected emotion has a dedicated `<emotion>` element. The `<category>` attribute *name* holds the name of the emotion category (as specified at page 8). The value of the set attribute stand for Geneva Emotion Wheel. Every category has 4 `<intensity>` levels marked with 1, 2, 3 or 4 stars, 1 is the lowest 4 is the highest. There are two `<dimensions>` defined for each emotion `<valence>` and `<control>`. Each can take real values from -1 to 1. When creating the markup the values stored in the Emotion instance are mapped to [-1, 1] interval. The `<word>` subelement of `<metadata>` specifies the value and the position of the substring that was identified to hint to a certain emotion.

System architecture and implementation

SemanticEmo was written entirely in Java by using NetBeans[13] IDE. The Emotion Ontology was partly written in Protégé and partly automatically generated (definitions for the 64 emotions). It uses, notably, the following Java libraries:

- Jena 2.6.2 – for reading the emotion ontology
- Twitter4J 2.1.0– for accessing Twitter web services and extract the articles related to a certain topic
- Stanford Parser 1.6.1 – for English text tokenization
- Junit 4.5 – for unit testing

For application development the GlassFish[14] application server was used.

The source code and detailed references can be found on the project's page at http://code.google.com/p/semanticemo .

Modules

The system has two main parts the Wonderland library used by the Semanticemo web application.

Wonderland

Wonderland is the author's first attempt to develop an application capable to sustain a conversation in a natural language with a human user. At this point in time, relevant for the SemanticEmo project are the following functionalities:

— providing an easy read only specific interface to the emotion ontology

13 http://netbeans.org

14 https://glassfish.dev.java.net

— implementation and management of the textual data acquisition modules (like the Twitter module)
— simple emotion detection based essentially on a rather statistic detection method
— external EmotionML markup of the simple textual data
— provides graphical maps based on all the emotions detected for a certain topic

Semanticemo

Semanticemo is a web application that allows the user to remotely use the functionalities provided by wonderland. It mainly contains a mixture of Java servlets and JSP pages. A simple CSS file is used to provide a consistent look for the entire application

User interface

The main page provides a text box that allows the user to specify the desired topic. Below that there are some checkboxes that can be used to select the sites where articles related to the desired topic will be searched. The search begins by pushing the right angle button (“>”).

Figure : **Main web application interface**

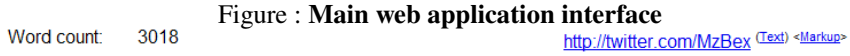


Figure : **Search result**

The result section is divided in three areas:

32. The top-left area contains a partial emotion wheel where only the emotions found at least once appear. It can be used as a quick view, a summary of all the emotions detected in the contexts related to the desired topic. Also provided are the names of every represented emotion category.
33. In the lower-left area some statistics related to the search are provided, particularly:
    a. Search time – the time needed to harvest the Web for the articles
    b. Detection time – the time required by the emotion detection algorithm
    c. Article count – the total number of articles that were harvested during the search
    d. Word count – the total number of words contained in all the articles
34. The right side is reserved to the links to the articles that were found relevant by the emotion detection. The articles are grouped by emotion category starting with Satisfaction and ending with Relief (see page 8 for the entire list). For each category the articles are listed in the descendant order of the emotion intensities. Every emotion is named and has a bullet whose color and size matches the one on the color wheel. Every article has three links:
    a. the main link, identified by the URL will open a new browser window/tab and an HTML page will be presented that will show the topic the link to the real article, the text and the emotions in a way that is easy to be red

b. the (Text) link will open a new browser window/tab and will display only the text representation of the article
c. the <Markup> link will display the XML EmotionML markup (like on page 13)

Figure : **HTML emotion markup**

Deployment and configuration

After compiling the application to a WAR archive it could be installed in a Tomcat [15] application server by using the admin interface. In web.xml configuration file set the **wonderland.data.path** `<env-entry>` to the folder containing the following two files:

- *Affect_Ontology1262693413691.owl* – the emotion ontology that can be found on the project's page
- *englishPCFG.ser.gz* – part of the Standford Parser distribution

15 http://tomcat.apache.org

Future development

There are many improvements that can be added to the current system. From simple to more complex can be mentioned:

— adding support for searching more web sites for articles
— increase the number of the emotions that can be detected by extending Affect_Ontology1262693413691 to include more emotion markers
— Add a quiz/test/questionnaire inviting the users to rate emotions according to varying dimensions. User options can also be used to better the emotion categories.
— use the parser to determine the relations between words and make the emotion detection process context-aware
— combine more articles in order to build an “opinion network” related to a certain topic and provide high quality statistical reports

Bibliography and links

1. *Emotion* – Stanford Encyclopedia of Philosophy – http://plato.stanford.edu/entries/emotion
2. Buraga, Sabin-Corneliu – *Interacțiune om-calculator (Curs)* – http://profs.info.uaic.ro/~busaco/teach/courses/interfaces/presentations/hci02-Emotie-InteractiuneNeconventionala.pdf
3. López, Juan Miguel; Gil, Rosa; García, Roberto; Cearreta, Idoia; Garay, Nestor - *Towards an ontology for describing emotions*
4. The HUMAINE Portal – http://emotion-research.net
5. Geneva Emotion Research Group (site) – http://www.unige.ch/fapse/emotion
6. Scherer, Klaus R. – *What are emotions? And how can they be measured?* – (2005)
7. Tran, Véronique – *The influence of emotions on decision-making processes in management teams* – (2004)
8. Emotion Markup Language - http://www.w3.org/TR/emotionml
9. Kim, Elsa; Gilbert, Sam; Edwards, Michael J.; Graeff, Erhardt – *Detecting Sadness in 140 Characters: Sentiment Analysis and Mourning Michael Jackson on Twitter* – http://www.webecologyproject.org/2009/08/detecting-sadness-in-140-characters
10. Boucouvalas, Anthony C. – *Real Time Text-to-Emotion Engine for Expressive Internet Communications*
11. Cosmovici A. – *Psihologie generală* – (1996)
12. Dorenboos, Robert B. – *Production Matching for Large Learning Systems* – (1995) – http://reports-archive.adm.cs.cmu.edu/anon/1995/CMU-CS-95-113.pdf