Knowl Inf Syst (2010) 25:421–443
DOI 10.1007/s10115-010-0320-1

REGULAR PAPER

Ontological reasoning for improving the treatment of emotions in text

Virginia Francisco · Pablo Gervás · Federico Peinado

Received: 31 October 2007 / Revised: 22 May 2009 / Accepted: 24 June 2010 /
Published online: 8 July 2010

Abstract With the advent of affective computing, the task of adequately identifying, representing and processing the emotional connotations of text has acquired importance. Two problems facing this task are addressed in this paper: the composition of sentence emotion from word emotion, and a representation of emotion that allows easy conversion between existing computational representations. The emotion of a sentence of text should be derived by composition of the emotions of the words in the sentence, but no method has been proposed so far to model this compositionality. Of the various existing approaches for representing emotions, some are better suited for some problems and some for others, but there is no easy way of converting from one to another. This paper presents a system that addresses these two problems by reasoning with two ontologies implemented with Semantic Web technologies: one designed to represent word dependency relations within a sentence, and one designed to represent emotions. The ontology of word dependency relies on roles to represent the way emotional contributions project over word dependencies. By applying automated classification of mark-up results in terms of the emotion ontology the system can interpret unrestricted input in terms of a restricted set of concepts for which particular rules are provided. The rules applied at the end of the process provide configuration parameters for a system for emotional voice synthesis.

Keywords Affective computing · Emotional annotation · Reasoning · Ontologies

V. Francisco (✉) · P. Gervás · F. Peinado
Departamento de Inteligencia Artificial e Ingeniería del Software,
Universidad Complutense de Madrid, Madrid, Spain
e-mail: virginia@fdi.ucm.es

P. Gervás
e-mail: pgervas@sip.ucm.es

F. Peinado
e-mail: email@federicopeinado.com

1 Introduction

The way in which the meaning of a sentence is built from the meaning of its words has been a subject of study in computational linguistics for a long time. No such study has been carried out for the way in which the emotional connotations of a sentence are affected by the emotional connotations of its words. Existing approaches to this task rely most often on a simplified representation of the sentence as a bag of words, where all words contribute in equal measure, much in the way information retrieval simplifies the treatment of text. However, intuitively certain words can probably be considered more significant, depending on the role they play in the word from their syntactic or semantic structure. An important hypothesis underlying this paper is that if this kind of sentence structure were to be represented computationally in a way that modeled how the emotional contributions of words affect the emotional connotations of the sentence, it would provide the means for capturing these intuitions. An static ontology of word dependencies within a sentence fulfills the requirements for such a representation.

An important challenge in addressing issues of affective computing is having an adequate representation of emotions. Existing approaches vary from identifying a set of basic categories—with a name tag assigned to each one of them—to design a multi-dimensional space in terms of primitive elements—or emotional dimensions—such that any particular emotion can be defined in terms of a tuple of values along the different dimensions. For different purposes, one approach is better suited than the other. For instance, when attempting to synthesize voice utterances that reflect emotion to some extent, it is easier to identify the parameters for voice production associated with conveying a particular emotion. For assigning emotional values to given utterances, on the other hand, human evaluators find it much easier to provide numbers along given dimensions. If one were to operate computationally with a representation of emotions expressed in more than one format, one is faced with the task of being able to convert from one to another. This task is reasonably easy when converting from emotional categories to emotional dimensions: it would suffice to assign a particular tuple of values for the emotional dimensions of each emotional category. When trying to convert from emotional values expressed in terms of emotional dimensions to a representation in terms of emotional categories this is not so simple. The problem lies in the fact that, given the subjectivity associated with emotional perception, the particular values assigned to a given impression by one person usually deviate slightly from what a different person would have assigned. This suggests that the process of converting from emotional dimensions to emotional categories should be carried out in a manner that allows a certain tolerance, so that a region of space in the universe of emotional dimensions is assigned to each emotional category, rather than just a single point in the universe.

A separate problem arises from the fact that there is a large number of emotional categories, and the differences and similarities between them are not clear cut. In some cases, it is reasonable to assume that certain emotional categories may be subsumed by others. For example, the emotion *anger* subsumes the emotions *sulking*, *displeasure* and *annoyance* which may be seen as different types of *anger*. This suggests that a taxonomy of emotional categories as a hierarchy might be useful in finding correspondence between more specific emotional categories and more general emotional categories.

In this context, the development of an ontology of emotional categories based on description logics, where each element is defined in terms of a range of values along the space of emotional dimensions, provides a simple and elegant solution. The ability to carry out automatic classification of concepts simplifies the addition of new concepts—possibly expressed only in terms of their values along the axes of emotional dimensions—without having to worry explicitly about where in the ontology they should be placed. Due to a taxonomical reasoning system an implicit hierarchy for the concepts represented in the ontology can be inferred automatically.

This paper describes the development of a system that relies on two ontologies such as those described above, together with its application as an interface between a text input marked up in terms of emotional dimensions and a set of rules for configuring an emotionally enabled voice synthesizer. By reasoning over the word dependency ontology, the emotional connotations of any node in the structure can be computed separately, from the contribution of single words to the contribution of entire sentences, including intervening substructures. By reasoning over the emotion ontology, insertion of new instances of emotional concepts into the ontology results in their automatic classification under the corresponding branch of the hierarchy. The system can then trace the ascendants in the ontology of the corresponding value, until a more general concept is found that satisfies the condition that specific rules are available for generating an appropriate voice synthesis configuration for expressing the intended emotional impression. Section 2 introduces the problem domain and the technologies that are used in the paper. Section 3 describes the two applications of ontological reasoning presented in the paper: how it is used to model compositionality of emotional contributions over sentence structure, and how it is used to model relations of subsumption between emotion concepts and emotion terms. Section 4 describes the application of these solutions in the task of providing input data and configuration information for an emotional voice synthesizer. Finally, Section 5 discusses the technological issues that have arisen, and Sect. 6 summarizes our conclusions and future work.

2 Related work

To understand the work proposed here, four basic topics must be covered: the computational representation of emotions, the Semantic Web technologies that have been employed in the project, the natural language processing technique employed to obtain the syntactic structure of sentences (dependency analysis) and the review of the system chosen as domain of application. These are covered in the following subsections.

2.1 Computational representation of emotions

Emotions are not easy to define, because there are a lot of factors that contribute to them. For Izard [13] a good definition of emotion must take into consideration: conscious feeling of emotion, process which takes place in the nervous system and in the brain and expressive models of emotion. Emotions take place when something unexpected happens and the so-called "emotional effects" begin to take control.

2.1.1 Classification of emotions

Interested readers can find more detail in the work of Cornelius [6] and Schröder [33].

Many of the terms used to describe emotions and their effects are difficult to tell apart from one another, as they are usually not well defined. This is due to the fact that the abstract concepts and the feelings associated with such concepts are very difficult to express with words. For this reason, there are a lot of methods for describing the characteristics of emotions: *emotional categories*—based on the use of emotion-denoting words—*descriptions based on psychology* [2] and *evaluation* [6], *circumflex models*—emotional concepts are represented by means of a circular structure [31], so that two emotional categories close in the circle are conceptually similar, and *emotional dimensions* which represent the essential aspects of emotional concepts.

In the following subsections, we describe in detail the two methods which are employed in our work: *emotional categories* and *emotional dimensions*.

Emotional categories: The most common method for describing emotions is the use of emotional words or affective labels. Different languages provide assorted labels of varying degrees of expressiveness for the description of emotional states. There are significant differences between languages in terms of the granularity with which these labels describe particular areas of emotional experience. Even within a given language, some areas of emotional experience have a higher density of labels than others. This diversity presents an additional difficulty. A lot of methods have been proposed to reduce the number of labels used to identify emotions: basic emotions, super ordinate emotional categories, and essential everyday emotion terms…

Emotional dimensions: Emotional dimensions represent the essential aspects of emotional concepts. There are three basic dimensions:

– *Evaluation*: Represents how positive or negative an emotion is. For example in a scale for the evaluation dimensions at one extreme we have emotions such as *happy*, *satisfied*, *hopeful*…the other end of the scale is for emotions such as *unhappy*, *unsatisfied*, *despaired*…
– *Activation*: Represents an active/passive scale for emotions. At one extreme of the activation are emotions such as *excited*, *aroused*…and at the other end of this scale are emotions such as *calm*, *relaxed*….
– *Power*: Represents the control exerted by the emotion. At one end of the scale we have emotions characterized as completely controlled, such as *care for*, *submissive*…and at the opposite end of this scale we have emotions such as *dominant*, *autonomous*…

This method is very useful because it provides a way of measuring the similarity between emotional states. Another important property of this method is that shifting the representational weight away from the actual labels employed allows for a relative arbitrariness when naming the different dimensions.

2.1.2 Structure of emotions

There are several approaches in the literature for determining which are the basic emotions, or which emotions are more general than others. There is a general agreement that some full-blown emotions are more basic than others. The number of basic emotions is usually small [6]. Some emotion categories have been proposed as more fundamental than others on the grounds that they include the others. Scherer [32] and Ortony [26] suggest that an emotion A is more fundamental than another emotion B if the set of evaluation components of the emotion A are a subset of the evaluation components of the emotion B. An example that may clarify the idea: five prototypes are proposed as underlying all emotional categories: *anger*, *love*, *joy*, *fear* and *sadness*. *Joy*, for example, would be subdivided into *pride*, *contentment*, and *zest*. Cowie and Cornellius [6] give a short overview of recent proposals of such lists.

Many psychologists have claimed that certain emotions are more basic than others. Ortony[26] summarizes some of these theories as follows. Plutchik's [29] postulates that there is a small number of basic, primary, or prototype emotions (*anger*, *anticipation*, *disgust*, *joy*, *fear*, *sadness* and *surprise*). All other emotions are mixed or derivative states; that is, they occur as combinations, mixtures, or compounds of the primary emotions. Plutchik states that all emotions vary in their degree of similarity to one another and that each emotion can exist in varying degrees of intensity or levels of arousal. Ekman [8] has focused on a set of six basic emotions that have associated facial expressions: *anger*, *disgust*, *fear*, *joy*, *sadness* and *surprise*. Those emotions are distinctive, among other properties, by the facial expression characteristic to each one. Izard [14] determines that the basic emotions are *anger*, *contempt*, *disgust*, *distress*, *fear, guilt, interest, joy, shame* and *surprise*. The OCC Model [25] has established itself as the standard model for emotional synthesis. It presents 22 emotional categories: *pride–shame, admiration–reproach*, *happy–resentment*, *gloating–pity*, *hope–fear, joy–distress*, *satisfaction–fear-confirmed*, *relief–disappointment*, *gratification–remorse*, *gratitude–anger* and *love–hate*. The OCC Model considers that categories are based on the valence reactions to situations constructed as: goal relevant actions and attractive or unattractive objects. Parrot [27] presents a deeper list of emotions, where emotions were categorized into a short tree structure, this structure has three levels: primary emotions, secondary emotions and tertiary emotions. As primary emotions, Parrot presents: *love*, *joy*, *surprise*, *anger*, *sadness*, and *fear*.

2.2 Semantic Web technologies

The Semantic Web is being developed with the intention of providing a global framework for describing data, its properties and relationships in a standard fashion. Many developers and researchers on knowledge systems are taking the approach of using Semantic Web technologies to obtain more interoperability and reusability with existing software and to take advantage of the strong trend of development that these technologies are experiencing nowadays.

In this section, we review the tools used in our project explaining what were the technological choices and the different criteria behind them.

2.2.1 Ontology web language

The Semantic Web relies heavily on ontologies. Concretely, ontologies based on Description Logics paradigm include definitions of concepts -OWL classes-, roles -OWL properties- and individuals. The most common language to formalize Semantic Web ontologies is OWL (Ontology Web Language [4]), a proposal of the W3C. The goal of this standard is to formalize the semantics that was created ad hoc in old frame systems and semantic networks. OWL has three increasingly expressive sub-languages: OWL Lite, OWL DL, and OWL Full.

OWL Lite is the simplest subset of OWL, specially designed to provide a quick migration path for other taxonomical structures.

OWL DL is the subset of OWL designed for applications that need the maximum expressiveness without losing computational completeness and decidability. It is based on Description Logics, a particular fragment of first-order logic, in which concepts, roles, individuals, and axioms that relate them (using universal and existential restrictions, negation, etc.) are defined. These entailments may be based on a single document or multiple distributed documents that we combine using the import OWL mechanisms. The OWL DL reasoning capabilities relies on the good computational properties of DLs. OWL DL has support for polyhierarchical automatic classification.

OWL Full ignores some significant restrictions of OWL DL, becoming a more powerful language for representing complex statements, but less useful for reasoning with them due to their computational properties.

2.2.2 Frameworks and APIs

An all-in-one framework or a versatile application programming interface would be a very desirable tool for any novice Semantic Web application developer. Java is probably the most important general-purpose language for developing Semantic Web applications, and it is also the language in which the original voice synthesizer was made, so the choice was obvious. However, there are at least two very promising Java frameworks available. One of them is Sesame [1], an open source RDF framework with support for RDF Schema inferencing and querying. The other one is Jena [12], another open source framework with a programmatic environment for RDF, RDFS, OWL, SPARQL, and its own rule-based inference engine.

Sesame has a local and remote access API, several query languages (included SPARQL) and it is more oriented to offer flexible and fast connections with storage systems.

Jena has also RDF and OWL APIs, tools to deal with RDF/XML, N3 and N-Triples formats, an SPARQL query engine and also some persistent storage functionality. It is important to consider that Jena is a useful tool for exploring the strong relation between SPARQL queries and OWL-DL ontologies [15].

For our purposes, performance issues can be ignored and only inference support for Description Logics is taken into consideration. The architecture of Sesame is probably easier to extend than the architecture of Jena, but from the point of view of the client building a wrapper for the functionality of the underlying framework, Jena is the most intuitive and usable API.

The framework that has been selected to carry out the development work in this paper is a short extension of DLModel's [28] functionalities. DLModel is a very straightforward open source API for accessing a Description Logic model instantiated in a set of ontologies, a set of DL-Safe rules [24] and a knowledge base. Although it has an abstract DL interface, it can be viewed as a wrapper on top of Jena that offers simpler methods to access concepts, roles, attributes, annotations, and individuals of the knowledge base from any Java application. This management includes DL reasoning and DL-Safe rule reasoning as implicit services that will benefit the programmer acting as an "intelligent black box" that knows how to use Semantic Web low-level technologies.

2.2.3 Ontology editor

Another important tool is the Integrated Development Environments (IDE) used to edit the ontology and the knowledge base. During our review of the state-of-art, we found two interesting editors able to perform this task: SWOOP and Protégé.

SWOOP [22] is a hypermedia-based OWL ontology browser and editor written in Java. It is open source and it tries to simplify the ontology development using an interface similar to a web browser. It includes some advanced features as ontology partitioning, debugging, and different kinds of visualization, so it makes ontologies more scalable, maintainable, and easy to use.

Protégé [7], specially the Protégé-OWL version, focuses on editing OWL ontologies. It is a powerful Java open source tool with a user-friendly interface that lets you edit and visualize ontologies in a very easy way. It can be seen as a framework for developing Semantic Web applications itself. The number of plugins (including some plugins for knowledge acquisition), the stability of the last version, the extensibility of its architecture (plug-and-play environment) software allows rapid prototyping and application development, just what we were looking for.

Ontology building is still more of a craft than an engineering task and that is the reason why tools as Protégé-OWL, which is fundamentally open to new collaborative features gaining importance within the Semantic Web paradigm, are a better choice for us.

2.2.4 Reasoner

Two different reasoners were considered for this project: Pellet [21] and Racer Pro [30].

Pellet is an open source DL reasoner completely implemented in Java. It deals not only with taxonomical reasoning but also with datatype reasoning, DL-Safe rules implemented in SWRL and other features that are considered very important for our project. Pellet is the default reasoner integrated with SWOOP.

RacerPro is an OWL reasoner and inference server for the Semantic Web. It is a well-known system for OWL/RDF which claims to be the most efficient and robust DL reasoner available. Nowadays, it is a commercial product, but some educational licenses are available.

When compared with Racer Pro, Pellet may have drawbacks, but ignoring again the problem of performance, Pellet is certainly one of the most feature-rich OWL reasoners. It is also supported by a strong development team and community, which is important if you are looking for different approaches and uses of the same tool, allowing the programmer to see what is happening in the internal code when something goes wrong or does not act as it is supposed to.

The language that the Jena implementation of DLModel used to communicate with any DL reasoner is DIG 1.1 [3], but Pellet is used as the default reasoner, being integrated as a Java library in the API.

2.3 Dependency analysis

The basic idea of dependency analysis is to describe the syntactic structure of a sentence in terms of dependency relations between pairs of words (a parent and its child). These relations compose a tree (the dependency tree). Dependency analysis has been used successfully for several applications: multilingual machine translation [19], recognizing textual entailment [16], and automatic evaluation of question–answer systems [11].

MINIPAR [17] analyses English texts with high accuracy and efficiency in terms of time. The main information that MINIPAR provides for each node in the dependency analysis is: its identifier, word, stem, part-of-speech, identifier of the parent node, and the dependency relation with the parent node. An example of the output generated by MINIPAR for the sentence “Two of her tears wetted his eyes and they grew clear again” is shown in Table 1.

The structure generated by MINIPAR can be transformed into a tree representation where the dependency relations are more easily visualized. Figure 1 shows the graphical representation of the dependency tree generated for the sentence in Table 1. The nodes in the structure are numbered, the arcs between the nodes represent dependency relation, and each dependency relation is labeled with a tag identifying the kind of relation involved.

2.4 Automated mark up of emotions in text

As a starting point of our approach, we have chosen a system that marks up text with emotional dimensions, EmoTag [9]. In these texts every emotional unit is marked up with the three emotional dimensions (*activation*, *evaluation*, and *power*). EmoTag currently uses the

Table 1 Example of dependency tree for the sentence "two of her tears wetted his eyes and they grew clear again"

0(							
E1	()	fin	C	*			
1	Two		N	5	s	(gov wet)	
2	of		Prep	1	comp1	(gov Two)	
3	her		N	4	gen	(gov tear)	
4	tears	tear	N	2	pcomp-n	(gov of)	
5	wetted	wet	V	E1	i	(gov fin)	
E3	()	Two	N	5	subj	(gov wet)	(antecedent 1)
6	his		N	7	gen	(gov eye)	
7	eyes	eye	N	5	obj	(gov wet)	
8	and		U	E1	punc	(gov fin)	
E0	()	fin	C	E1	conj	(gov fin)	
9	they		N	10	s	(gov grow)	
10	grew	grow	V	E0	i	(gov fin)	(antecedent 9)
E4	()	they	N	10	subj	(gov grow)	
11	clear		A	10	desc	(gov grow)	
12	again		A	10	mod	(gov grow)	
)							

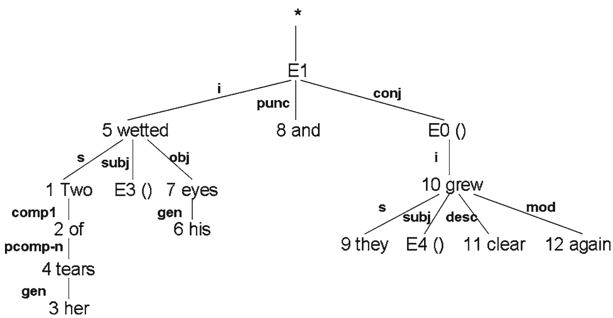

Fig. 1 Example of the graphical representation of the dependency tree for the sentence "Two of her tears wetted his eyes and they grew clear again"

sentences of the text as emotional units. This implies that every sentence has a value for each of the three dimensions. The emotions associated with each of the sentences try to rate how the reader or listener will feel while reading or listening to each sentence.

EmoTag classifies sentences into emotions. A corpus of example texts previously annotated by human evaluators was mined for an initial assignment of emotional features to words. This results in a list of emotional words (LEW) which becomes a useful resource for later automated mark up. EmoTag employs for the assignment of emotional features a combination of the LEW resource, the ANEW word list [5],[1] and WordNet [20] for knowledge-based expansion of words not occurring in either. In the mark-up process, the first step

[1] The ANEW word list is a set of normative emotional ratings for a large number of words in the English language. Words are rated in terms of *evaluation*, *activation* and *power*.

Table 2 Fragment of a marked up tale

```
...
<emotion act=5 eval=5 pow=5>"Good day, Mistress Crow"</emotion>
<emotion act=9 eval=7 pow=5>"How well you are looking today: how glossy your feathers; how
bright your eye."</emotion>
<emotion act=9 eval=7 pow=5>"I feel sure your voice must surpass that of other birds, just
as your figure does;</emotion>
<emotion act=9 eval=7 pow=5>let me hear but one song from you that I may greet you as the
Queen of Birds."</emotion>
...
```

is to split the texts into sentences and split each sentence into words to carry out the process based in the relation between words and different emotions. The operation of the mark-up process involves three different tasks: identifying which words should contribute to the emotional value computed for the sentence, computing the emotional values for those words, and combining the emotional values of the words into an emotional value for the sentence.

The existing version of EmoTag carries out these tasks in the following way:

– The words in the sentence are filtered using a stop list, and dependency analysis is applied to identify which words are under the scope of negation. Words in the stop list and words under the scope of negation are treated specially.
– The emotional value associated with each word is obtained by looking them up in an affective dictionary (known as the LEW list).
– Emotional values are computed for words under the scope of the negations by inverting the values of the original words.
– Once all the words of the sentences have been evaluated, the average value for each dimension is calculated, with no consideration for sentence structure.

A sample part of a marked tale by EmoTag is given in Table 2.

3 Ontology-supported mark up of emotions

The contribution of the paper involves two different applications of ontologies in the mark-up process: a word dependency ontology is used to permit reasoning processes in the composition of sentence emotion from word emotions, and an emotion ontology is used to interpret the resulting values assigned for emotional dimensions in terms of the emotional concepts in the ontology, which are directly mapped to emotional categories. Emotional concepts are used here as a form of interlingua that simplifies the semantic mapping [18].

3.1 Dependency analysis ontology for the obtaining of words implied in the mark-up process

We have developed an ontology for the representation of the dependency analysis of the sentence. This ontology is based in the dependency trees generated by Minipar as described in Sect. 2.3.

3.1.1 Structure

The ontology has a root concept called *Node* which has two disjoint subconcepts. These subconcepts are *SentenceNode*, representing the set of individual sentences of the tale, and *WordNode*, representing the set of individual words that appears in the text. There is also a

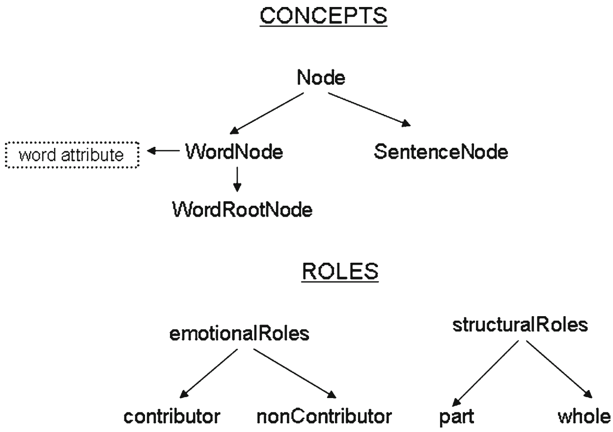

Fig. 2 Structure of the Word Dependency Ontology

special subconcept called *WordRootNode* that is the first node to be connected to a sentence in the MINIPAR tree.

The most interesting part of the ontology is the role hierarchy. It contains the relations of MINIPAR organized under two useful super roles called *contributor* and *nonContributor* that distinguish between relations that allow a word to contribute to the global emotion of the sentence and those that do not. The contributor role is marked as transitive and it connects a word node with any kind of node, no matter if it is a word or a sentence. There are also other structural roles as *part* and its inverse role called *whole*, both allowing the connection between words and their corresponding sentences. Finally, an attribute called *word* is used to store the String representation of each word. Figure 2 shows graphically the structure of the ontology.

3.1.2 Generation of the OWL file based on the dependency analysis of Minipar

The starting point for the creation of the ontology is the dependency analysis generated by Minipar for each of the sentences that compose the text. Instead of using a formal concept-based process, a more empirical method is used to transform this data into an OWL file. The method works as follows:

Minipar returns six main fields for each node of the tree: the identifier, the word which it refers to, the stem of the word, the part-of-speech of the word in the sentence, the identifier of the parent node and the relation between the node and its parents. For the tree representation that we want to obtain in the ontology, we require four of these fields: the identifier, the word, the identifier of the parent node and the relation between the parent and the child. Based on these values, we generate the OWL file which is composed of the following fragments:

– Head of the OWL File.
– Root node of each sentence in the text, indicating the node identifier of the root node and the sentence which belongs to.

```
<emo:WordRootNode rdf:ID="idRootNode">
<emo:part rdf:resource="#idSentence"/>
</emo:WordRootNode>
```

– For each node in the dependency tree we must specify the identifier, the word as a datatype property of the node, the identifier of the parent, the relation with the parent and the sentence which the node belongs to.

```
<emo:WordNode rdf:ID="idNodo"/>
<emo:word rdf:datatype="#string">wordNode</emo:word>
<emo:relation rdf:resource="#idParent"/>
<emo:part rdf:resource="#idSentence"/>
</emo:WordNode>
```

3.1.3 Obtaining the words implied in the mark-up process

When considering that the root node of a sentence is correctly connected with the contributor role to that sentence (which can be done automatically when creating the knowledge base file), nothing more is necessary to obtain the words implied in the mark-up process. Owing to the transitive property of the contributor role, every single word that has an "unbroken" path of contributor subroles that reach the root node of the sentence is a contributor to the emotional value of the sentence. So, due to the predefined semantics of OWL, the contributors of each sentence are obtained automatically.

In this way, the assignment of *contributor* or *nonContributor* roles to each one of the dependencies used by Minipar basically encodes the compositionality of emotional value in terms of sentence structure. The current configuration of the ontology assigns the *nonContributor* role to the following set of Minipar dependency relations: `Det`, `amod`, `Desc`, `mod`, `gen`, `punc`, `whn`, `be`, `aux`, `c`, `by-subj`, `neg`, `guest`, `whp`, `have`, `inv-aux`, `wha`. The selection has been carried out on the basis of the following criteria:

– Certain dependencies allow the identification of word types that were originally contemplated in the stop list
– Other dependencies constitute instances of structural subordination that suggest some variation in the transmission of emotional value.

This initial role assignment is intended as a first approximation, subject to subsequent refinement based on analysis of available corpora of sentences hand-tagged with emotions [9] and word collections with emotional values assigned by human evaluators [5].

3.1.4 Example

Figure 3 shows an example of the generation of the ontology for the sentence "Two of her tears wetted his eyes and they grew clear again". In the figure, the first table contains the dependency analysis returned by Minipar, based on this information, we obtain automatically the required fields for the representation of dependency analysis in our ontology. Based on these fields shown in the second table, we generate the OWL File which is going to be used for the definition of the ontology. Once the ontology has been defined, the reasoner automatically classifies the contributor nodes (*wetted*, *tears*, *eyes*, *grew* and *clear*) which are used by EmoTag to determine the emotion of the sentence.

3.2 Emotional ontology for the correct interpretation of final mark-up values

The assignment of emotional values provided by the mark-up system at this stage provides for each sentence a triple of values for its three emotional dimensions. To enable the system to interpret these value assignments in terms of emotional concepts that people may

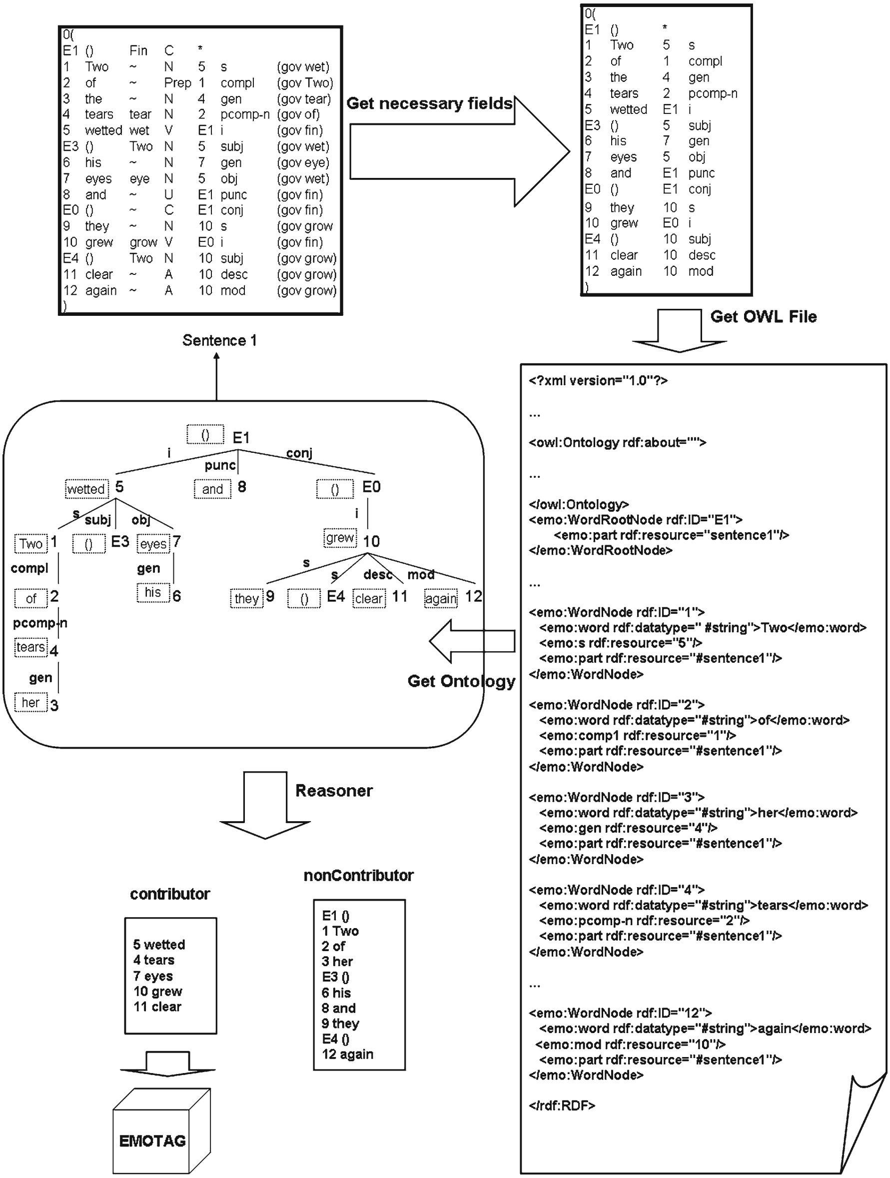

Fig. 3 Example of how the emotional contribution of the sentence is obtained

understand more easily, we have developed an ontology of emotional categories. They are structured in a taxonomy that covers from basic emotions to the most specific emotional categories. This ontology is based on the emotional structures mentioned in the Sect. 2.1.2. As basic emotions we have: *sadness*, *happiness*, *surprise*, *fear* and *anger*. We have adapted the Parrot model to these basic emotions, and we have integrated in this model all the emotions which appeared in other models. Then we have added all the emotion-denoting words of the

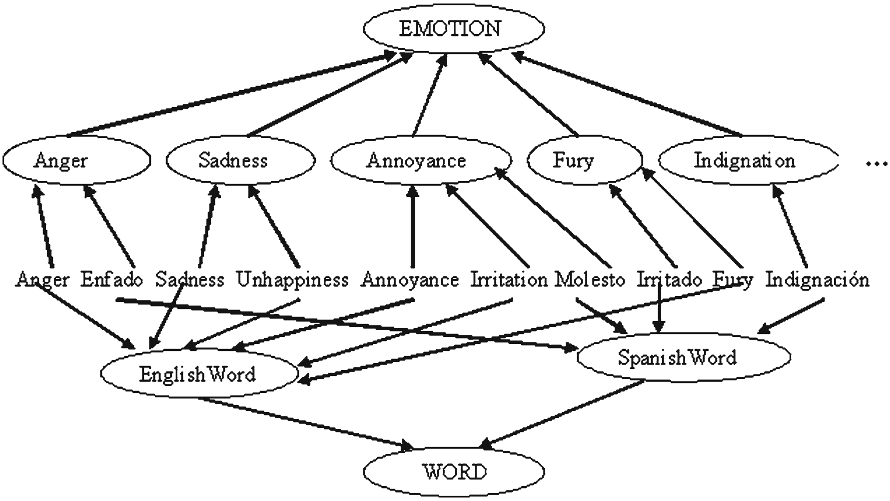

Fig. 4 Fragment of the emotional ontology

English and Spanish languages. Finally, each of the emotional categories is related with the three emotional dimensions by means of data ranges.

3.2.1 Structure

Our ontology has two root concepts:

– Emotion: This is the root for all the emotional concepts which are used to refer to emotions. Each of the emotional concepts are subclasses of the root concept Emotion. Some examples of these subclasses are: *Anger*, *Annoyance*, *Displeasure*, *Sad*, *Happy*, *Surprise*, *Fright*, *Horror*…
– Word: This is the root for the emotion-denoting words, the specific words which each language provides for denoting emotions. Our ontology is currently available for two different languages: English and Spanish. To classify the words into their corresponding language, the root concept Word has two subclasses: *EnglishWord* and *SpanishWord*.

As instances of the *EnglishWord* and *SpanishWord* subclasses, there are emotion-denoting words, which are all the words used for denoting *Anger*, *Annoyance*, *Displeasure*, *Terror*…Each of these instances has two parents: a concept from the Emotion hierarchy (which indicates the type of abstract emotion denoted by the word) and a concept from the Word hierarchy (which indicates the language of the word).

It is important to note here that, because the ontology is intended to operate over input in the form of language utterances, the ontology must include the means for representing emotional words. Therefore, it includes the specific concept of Word. All actual emotional categories handled by the system must be instances of this concept or one of its subclasses. Specific sub-hierarchies are added to group together all words which refer to emotions in a given language.

Figure 4 shows a fragment of the ontology. In this fragment, it can be seen how the emotional categories are related both to one emotional concept and to one word concept, for example, the word *unhappiness* is an instance of the emotional concept *Sadness* at the same time it is an instance of the word concept *EnglishWord*, which means that *unhappiness* is an English word for denoting the emotion sadness.

Another valid way of representing these relations might be to create a new property called "language" to connect each word with an instance of the language it belongs. We have chosen the in-built "type" relation because individuals with many different types are considered natural in OWL DL, and it is easier to retrieve every word of a specific type than "every word that has a relation with a specific individual".

In handling words, the system may need to identify synonyms for a particular word, that is, other emotional categories which may be used to refer to the same concept. This is specially relevant considering the importance of this type of semantic relations in text processing, as compared to the classic bag of words approach [34].

Given the semantics, we have chosen for our ontology, two instances of the Word concept can be considered to be synonyms if they are also instances of the same single Emotion concept from the parallel Emotion subhierarchy. For example, in the figure above, we can find that the words *annoyance* and *irritation* are synonyms because they are both instances of the Emotion concept *Annoyance*.

To summarize we can conclude that our emotional ontology represents the emotional categories as instances of a tree structure of emotional concepts. Each emotional word is an instance of two concepts: an emotional concept which represents the emotion denoted by the emotional word and a word-of-a-particular-language concept which determines the language to which the word belongs. From a given emotion-denoting word by means of our ontology we obtain the direct emotional concept associated with it as well as the more general emotional concept related to the direct emotional concept. It is also possible to obtain the synonyms for an emotional word, and the corresponding word in the other language. For example, given the emotional word *grief*, we have as direct emotional concept *Grief*, as general emotional concepts *Distress*, *Sadness*, and *Emotion*, as Spanish translation *Agonia* and finally as synonyms *agony*, *anguish*, and *sorrow*.

3.2.2 Datatype properties

Once we have a hierarchy of emotions, relations between the emotion-denoting words and their language and the concept they represent, we want to link the emotional concepts with the three emotional dimensions. Numeric data can be represented in OWL using datatype properties. To achieve this, we have declared three datatype properties:

– *hasEvaluation*: Represents the data range for the dimension evaluation.
– *hasActivation*: Represents the data range for the dimension activation.
– *hasPower*: Represents the data range for the dimension power.

Each of the emotional concepts is defined by specifying appropriate data ranges for these properties as described in the following section.

3.2.3 Data range

We have defined each of the emotional concepts through the emotional dimensions defined as datatype properties. Each emotional concept takes up a region in the three-dimensional space of emotional dimensions. To describe this with the datatype properties, we have to define our own datatype restrictions, because we are using specific intervals between numbers of type float. This can be done using data range definitions.

Table 3 Fragment of the OWL ontology

```
<owl:Restriction>
 <owl:allValuesFrom>
  <owl:DataRange>
   <owl:onDataRange rdf:resource=‘‘http://www.w3.org/2001/XMLSchema#float’’/>
   <owl:minInclusive rdf:datatype=‘‘http://www.w3.org/2001/XMLSchema#float’’>
   7.0</owl:minInclusive>
  </owl:DataRange
 </owl:allValuesFrom>
 <owl:onProperty>
  <owl:FunctionalProperty rdf:about=‘‘#hasActivation’’/>
 </owl:onProperty>
</owl:Restriction>
<owl:Restriction>
 <owl:onProperty>
  <owl:FunctionalProperty  rdf:about=‘‘#hasActivation’’/>
 </owl:onProperty>
 <owl:allValuesFrom>
  <owl:DataRange>
   <owl:onDataRange rdf:resource=‘‘http://www.w3.org/2001/XMLSchema#float’’/>
   <owl:maxInclusive rdf:datatype=‘‘http://www.w3.org/2001/XMLSchema#float’’>
   10.0</owl:maxInclusive>
  </owl:DataRange>
 </owl:allValuesFrom>
</owl:Restriction>
```

For example, we have the *Anger* emotional concept, we can describe the region of the space associated with it in the following way: 7<=hasActivation<=10, 0<=hasEvaluation<=3, 3<=hasPower<=5.

The fragment of the OWL file which correspond to the data range for the *hasActivation* property is shown in Table 3.

In this way, by means of the data ranges on the datatype properties, the link between the abstract emotional concepts and the three-dimensional space of emotional dimensions is established.

3.2.4 Automatic classification of emotions using datatype properties

A requirement to be taken into consideration when representing emotions using numerical data are to have some reasoning device capable of processing such data in an appropriate way. Pellet is able to classify concepts with restrictions formed by combinations of user-defined datatypes.

Once we have defined the emotional concepts by means of the emotional dimensions, Pellet automatically classifies the concepts into a hierarchy of emotional concepts. This means that Pellet obtains a hierarchy of emotions in which the most basic concepts are at the top of the hierarchy and the concepts which are more specific appear as descendants of the more general ones.

Datatype properties transform the classification of the emotional concepts into a relatively simple task. It is not necessary for the designer of the ontology to know which concepts are more specific than others because it is the reasoner that carries out the task automatically. For example, we have the following emotional concepts: *Anger*, *Annoyance*, *Fury* and *Indignation*. *Anger* is one of the basic emotions and *Annoyance*, *Indignation* and *Fury* are different forms of *anger* that differ from one another in their intensity of arousal. We define the four

Table 4 Example of datatype properties assigns to *Anger*, *Annoyance*, *Fury* and *Indignation*

Emotion	hasActivation	hasEvaluation	hasPower
Anger	$7 <= x <= 10$	$0 <= y <= 3$	$3 <= z <= 5$
Annoyance	$7 <= x < 8$	$0 <= y <= 3$	$3 <= z <= 5$
Indignation	$8 <= x < 9$	$0 <= y <= 3$	$3 <= z <= 5$
Fury	$9 <= x <= 10$	$0 <= y <= 3$	$3 <= z <= 5$

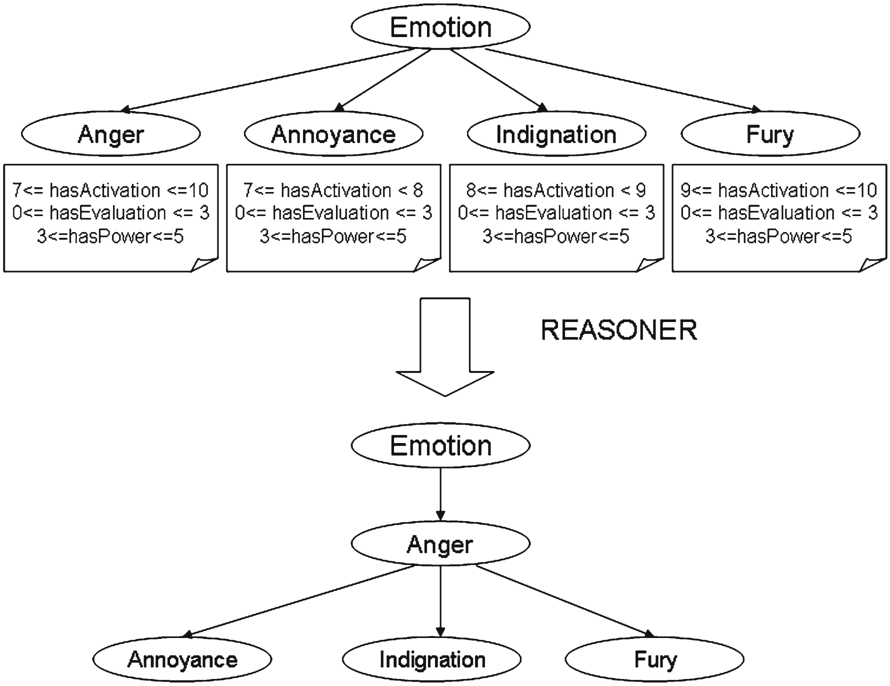

Fig. 5 Example of the automatic classification of emotions using datatype properties

concepts as subclasses of the root concept Emotion, and we define the ranges in Table 4 for the three datatype properties.

Just by loading the ontology in DLModel, the reasoner automatically classifies the concepts *Annoyance*, *Indignation* and *Fury* as subclasses of the emotional concept *Angry* which is automatically identified as more general. In Fig. 5, we can see how the reasoner classifies the concepts in the correct way.

4 An example application of the whole process

In this section, an example of application of the complete process is explained. The first subsection explains how the Word Dependency Ontology is used to obtain the words that should be considered to compute the emotional dimensions of the sentence. The second subsection explains the application of the Emotion Ontology to convert a text marked up with emotional dimensions into a text marked up with emotional categories. The third subsection explains how the resulting marked up text is used as the input of an emotional synthesizer.

4.1 Using dependency ontology to obtain the words implied in the mark-up process

The new version of EmoTag replaces the list of stop POS tags with the Dependency Ontology. This process now is carried out in the following way:

– Obtain the words affected by negations (by means of the dependency analysis), the stem and the part-of-speech of each of the words.
– Creation of the ontology from the dependency analysis generated by Minipar.
– By means of the roles defined in the dependency ontology, Pellet automatically discards the words which are related with its parents by *nonContributor* relations such as *mod* (modifier), *punc* (punctuation), *det* (determiner)…Once we have identified the words which are going to take part in the mark-up process, EmoTag by means of DLModel obtains this words.
– Obtaining the emotional value associated with the words obtained in the previous step looking in the affective dictionary (LEW list).
– Processing the words under the scope of the negations.
– Obtaining the final value of the sentence based on the emotions associated with the words which compound it. Once all the words of the sentences have been evaluated, the average value for each dimension is calculated.

4.2 Using automatic classification to interpret emotional dimensions as emotional categories

Each sentence of the marked up text is related to a point in the three-dimensional space of emotions. This point is the input to our ontology of emotions, which by means of the datatype properties and the datarange restrictions, automatically classifies this point under a given emotional concept. Once we have identified the specific emotional concept to which the input point is related, by means of DLModel we recursively obtain its ancestors until we locate the one which corresponds to one of the five basic emotions (*anger*, *happiness*, *sadness*, *fear* and *surprise*).

4.3 Configuring a voice synthesizer based on the emotional mark up

EmoSpeech [10] is a system capable of modulating the voice quality of a synthesizer while reading aloud children's tales, so that the voice conveys at least part of the emotions expressed by the corresponding text. This is achieved by controlling those parameters in the synthesizer that have been identified as having more relevance in the expression of emotions in human voice. EmoSpeech operates with five basic emotions:*anger*, *happiness*, *sadness*, *fear* and *surprise*. The aspects of the voice that act as personality identifiers are: volume, rate, pitch baseline and pitch range. EmoSpeech uses a group of rules which relates the five basic emotions to the specific changes on voice parameters involved in the communication of emotion in human voice utterances. The values of these parameters for every emotion were obtained by refining an original proposal by Schröder [33], based on the analysis of emotional material generated by actors. The optimal values were obtained through the systematic variation of the parameters during synthesis. Table 5 summarizes the rules of the synthesizer for the basic emotions.

Using the particular configuration of parameters for that particular basic emotion, the synthesizer reads out aloud the text with the emotion assigned by EmoTag to the sentences.

Table 5 Configuration parameters for emotional voice synthesis

	Volume (%)	Rate (%)	Pitch baseline (%)	Pitch range (%)
Anger	+10	+21	+0	+173
Surprise	+10	+0	+25	+82
Happiness	+10	+29	+35	+27
Sadness	−10	−8	−10	−36
Fear	+10	+12.5	+75	+118

4.4 A complete example

In Fig. 6, we can see how this process works for a concrete example. In this example, we have a sentence of input text which EmoTag marks up with the following values: *activation* = 7, *evaluation* = 1 and *power* = 5. In the mark-up process, EmoTag only takes into consideration the words *cried*, *he*, *voice* and *children* which are the words identify by the reasoner as *contributor* based on the dependency relations generated by Minipar. The values obtained by EmoTag in the mark-up process represent a point in the dimensional space which is classified by means of the emotional ontology under the *annoyance* emotional concept. We ask DLModel for the parents of *annoyance* and the *anger* emotional concept is returned. EmoSpeech then receives the sentence of the input text and the emotion *anger* as the one associated with the sentence, so it selects the rules corresponding to this basic emotions. Once EmoSpeech has the suitable rules for the emotional meaning of the sentence, the synthesizer reads aloud the sentence in an angry way.

5 Discussion

The introduction of a specific ontology to model the dependencies between words in a sentence provides the means for defining specific roles between concepts to capture how the emotion of a given node in the dependency graph is affected by the emotions of other nodes related to it. The version described in this paper outlines the definition of a single type of relation “contributor”, which represents the fact that a given subnode contributes its emotional connotations to the node that it depends on. The absence of such a relation indicates that the subnode does not contribute. This is a simple model that presents a significant advantage: it provides a way of computing the emotional contribution of any node in a given dependency graph. This emotional contribution can be computed from the emotional contributions of all the nodes that contribute to that node. This allows easy computation of the emotional contribution of any kind of linguistic construct (words, noun phrases, verb phrases, sentences...) as long as the particular construct can be identified with a particular fragment of the dependency analysis starting from a given node.

The emotional ontology provides the means for connecting text marked up in terms of emotional dimensions to a synthesizer that is in principle only capable of processing material marked up in terms of emotional categories. The effect of the emotional ontology on the quality of speech output is limited,[2] because the emotion ontology is being used only as interface between the emotional mark-up application and the voice synthesizer. For inputs originally tagged with emotional categories, the addition of the ontology has not much impact. Nevertheless, emotional categories as a method of representing emotions provide only very limited

[2] The quality and emotional precision of the resulting voice has been discussed elsewhere. Details can be found in [10].

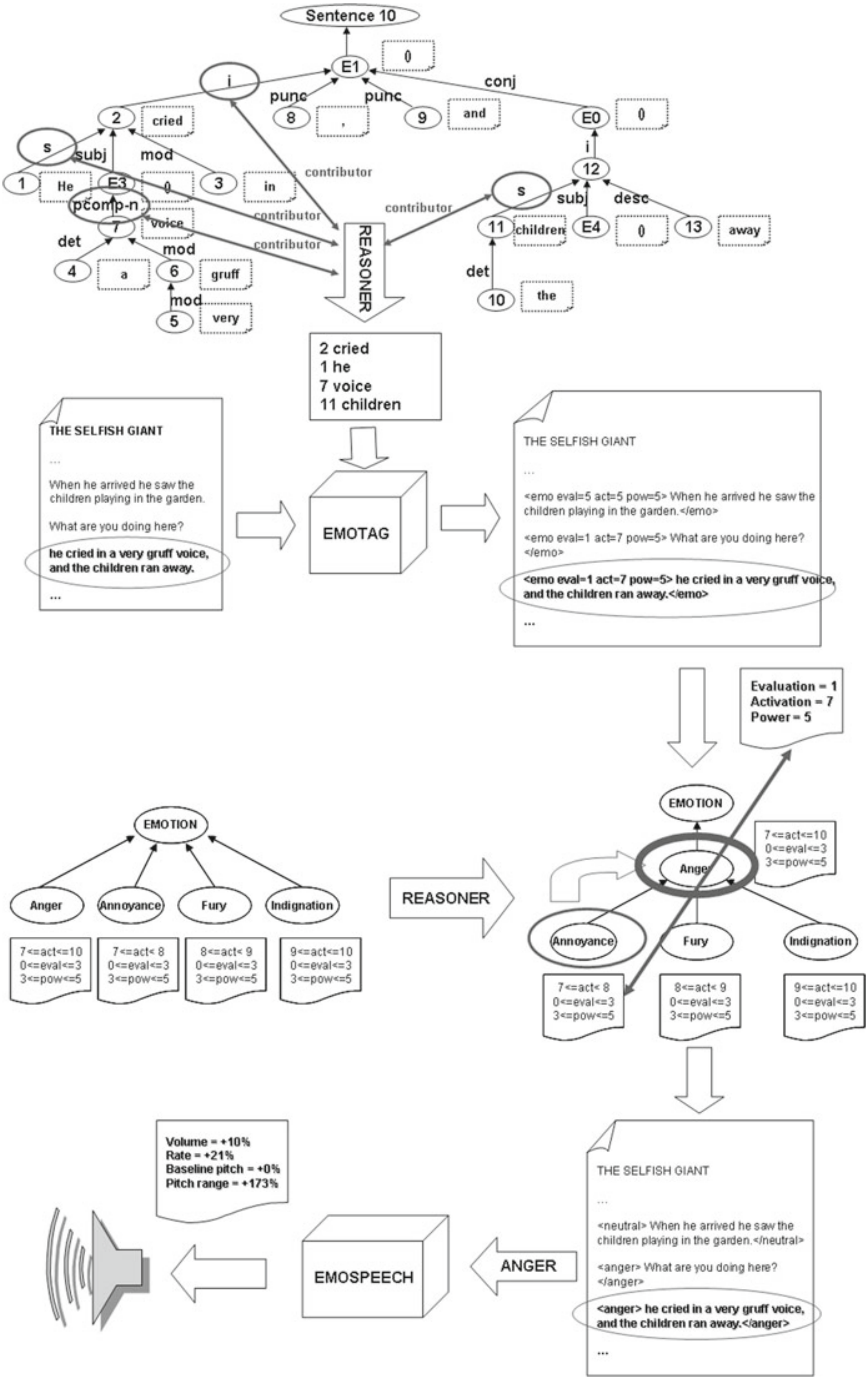

Fig. 6 Example of the entire process

granularity, restricted to the five basic emotions. On the other hand, emotional dimensions provide a much more flexible means of representing emotions, with greater expressive power. The main obstacle in switching from one representation to another lies in the fact that there is no easy way of converting from emotional dimensions to voice synthesizer configurations. At best, the three-dimensional space of emotional dimensions could be partitioned into restricted volumes of space, and a particular configuration of the synthesizer assigned to each volume. The option of using a description logic ontology—and the associated abilities to carry out instance recognition and automatic classification—as an interface to achieve this conversion as proposed in this paper, presents two distinct advantages:

– It provides a method for the automatic association of any point in the three-dimensional space to whatever is the closest available configuration of the speech synthesizer, based on the information that is defined at the conceptual level—even if it relies on an underlying level of geometrical representation.
– Any subsequent refinement of the set of configurations available for the synthesizer—for instance, if the existing configurations are refined into a larger set of options by fine tuning them to better represent more specific emotions—it would be enough to associate the new configurations to the corresponding concepts, and to refine the search algorithm to stop at the first ancestor that has some configuration data associated with it.

Regarding the technologies that have been applied in this proposal, some of these are not generally accepted as standard. Datatypes (and "reasoning" with numbers and strings) are not part of the essence of Description Logics. OWL DL considers datatype properties disjoint with every object property. In the EL, QL, and RL profiles of the next OWL 2 [23] support for datatypes have been improved, because it is useful for many applications. The version of OWL that we have used only supports some standard XML Schema datatypes and it lacks a standard solution for representing user-defined datatypes. DIG 1.1, being a standard designed for the communication with DL reasoners, does not accept restrictions over datatype properties. This obstacle made it impossible for us to send an ontology that includes such restrictions directly from Protégé to Pellet for its automatic classification. DIG 2.0, with support for the new OWL 2 will offer those features, but for now other shortcuts must be used to reason with restrictions on datatype properties. Our old version of Protégé had a proprietary solution to represent user-defined datatypes, which allows the creation of restrictions with interesting datatype properties and even visualization of the limits of a numeric interval and things like that in the GUI. However, DIG 1.1 does not allow that kind of information to travel to a DL reasoner. Pellet, by itself, can deal with user-defined datatype restrictions, and now the last version supports the inline syntax proposed by OWL 2. So because we are using Protégé as the editor for our ontology and knowledge base, we have to edit the files manually to add those restrictions before loading everything in DLModel using the "Pellet-Java" default configuration. We hope that some of these shortcomings might be solved when updating to the later versions of these technologies.

6 Conclusions and future work

The word dependency ontology and the initial assignment of `nonContributor` roles to a subset of Minipar dependency relations provides a simple model of the compositionality of emotional connotations. Although simple, the reasoning capabilities provided by the ontology ensure that this model allow correct treatment of sentence structure no matter how complex, since the reasoner deals with the recursive application of the relevant roles. As noted in Sect. 3, future work contemplates the empirical validation of the initial assignment of `nonContributor` roles to Minipar dependency relations. A very simple extension of the ontological approach to computing sentence emotion from word emotion can be achieved by defining more than one mode of contribution (contribute with half the intensity, invert the sign of the contribution...). The final stage of the process, which computes a single emotional representation for the node from the set of elements that contribute to it, would then be refined so that different forms of contribution are considered differently. This would allow the system to treat differently linguistic constructions that affect the emotional contribution, such as for instance modal verbs ("I wish I was getting married" need not have the same emotional effect as "I am getting married"), reported speech ("He said I jumped into the fire" is different from "I jumped into the fire"), subordinate clauses ("She saved the dog that had fallen off the cliff" is different from "The dog had fallen off the cliff")…The possibility of defining specific roles for each dependency relation provides the means for blocking some of the contributions of emotional content, or for modifying the sign or the importance of the contribution.

An emotional ontology based on description logics has been implemented using Semantic Web technologies. Each emotional concept is defined in terms of a range of values along the three-dimensional space of emotional dimensions, that allows the system to make inferences concerning the location of new concepts with respect to the taxonomy. This constitutes a valid solution to the problem of finding a relationship between an arbitrary point in a space of emotional dimensions and the set of basic emotional categories usually identified with specific names. The importance of being able to identify such relationships is strengthened by the fact that configuration of synthesizer parameters for artificially producing emotional voice tends to be established in terms of basic emotional categories. The ontology described in this paper has demonstrated its usefulness as part of a complex process of converting unmarked input text to emotional voice, resolving the problems that originated at the interface between the emotional tagging in terms of emotional dimensions and the synthesis of emotional voice in terms of basic emotional categories. In this process, both the capability for automatic classification provided by the reasoner, and the hierarchical structure provided by the ontology played important roles.

Although reasoning support for datatype properties in OWL DL is still not standard, technologies are available that let us experiment with these features and allow us to develop affective computing applications, such as the emotional voice synthesizer described in the paper. OWL, Jena, DLModel, Protégé and Pellet are the choices we made before developing this new iteration of the software. Still more improvements are needed in editors such as Protégé to be compatible with reasoners such as Pellet. Testing SWOOP is going to be one of our next steps in order to facilitate the acquisition of knowledge for the emotional knowledge base.

Acknowledgments This research is funded by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01 project) and a joint research group Grant (CCG08-UCM/TIC-4300) from the Universidad Complutense de Madrid and the Dirección General de Universidades e Investigación of the Comunidad Autónoma de Madrid.

References

1. Aduna and NLnet-Foundation (n.d.) Sesame. http://www.openrdf.org
2. Alter K, Rank E, Kotz S, Toepel U, Besson M, Schirmer A, Friederici A (2000) Accentuation and emotions—two different systems? In: Proceedings of the ISCA workshop on speech and emotion, Northern Ireland, pp 138–142

3. Bechhofer S, Moller R, Crowther P (2003) The DIG description logic interface. In: Description logics. CEUR Workshop Proceedings
4. Bechhofer S, van Harmelen F, Hendler J, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein A (n.d.) OWL Web Ontology Language Reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/
5. Bradley M, Lang P (1999) Affective norms for English words (ANEW): stimuli, instruction manual and affective ratings. technical report c-1. Technical Report, The Center for Research in Psychophysiology, University of Florida
6. Cowie R, Cornelius R (2003) Describing the emotional states that are expressed in speech. In: Speech communication special issue on speech and emotion
7. Crubzy M, Dameron O, Fergerson RW, Knublauch H, Musen MA, Noy NF, Rubin D, Tu SW, Vendetti J (n.d.) Protg project. http://protege.stanford.edu/
8. Ekman P (1992) Are there basic emotions? Psychol Rev 99(3):550–553
9. Francisco V, Gervás P (2006) Exploring the compositionality of emotions in text: word emotions, sentence emotions and automated tagging. In: Proceedings of the AAAI-06 workshop on computational aesthetics: AI approaches to beauty and happiness, Boston
10. Francisco V, Hervás R, Gervás P (2005) Análisis y síntesis de expresión emocional en cuentos leídos en voz alta. In: 'Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural', Granada, Spain
11. Herrera J, Peas A, Rodrigo A, Verdejo F (2006) UNED at PASCAL RTE-2 Challenge. In: Proceedings of the second PASCAL challenges workshop on recognising textual entailment, Venezia, Italy
12. Hewlett-Packard (n.d.) Jena: a semantic web framework for java. http://jena.sourceforge.net/
13. Izard C (1971) The face of emotion. Appleton-Century-Crofts, New York
14. Izard C (1977) Human emotions. Plenum Press, New York
15. Jing Y, Jeong D, Baik D-K (2008) SPARQL graph pattern rewriting for OWL–DL inference queries. Knowl Inform Syst (online first)
16. Kouylekov M, Magnini B (2006) Tree edit distance for recognizing textual entailment: estimating the cost of insertion. In: Proceedings of the second PASCAL challenges workshop on recognising textual entailment, Venezia, Italy
17. Lin D (1998) Dependency-based evaluation of MINIPAR. In: Proceedings of workshop on the evaluation of parsing systems, Granada, Spain
18. Linhalis F, de Mattos Fortes R, de Abreu Moreira D (2009) OntoMap: an ontology-based architecture to perform the semantic mapping between an interlingua and software components. Knowl Inform Syst (online first)
19. Maxwell D, Schubert K (1989) Metataxis in practice: dependency syntax for multilingual machine translation. Foris Publications, Dordrecht
20. Miller G (1995) Wordnet: a lexical database for english. Commun ACM 38:39–41
21. Mindswap (n.d.a) Pellet owl reasoner. http://www.pellet.owldl.com/
22. Mindswap (n.d.b) SWOOP: a hypermedia-based featherweight OWL ontology editor. http://www.mindswap.org/2004/SWOOP/
23. Motik B, Cuenca-Grau B, Horrocks I, Wu Z, Fokoue A, Lutz C (n.d.) OWL 2 Web Ontology Language Profiles. http://www.w3.org/TR/owl2-profiles/
24. Motik B, Sattler U, Studer R (2005) Query answering for OWL–DL with rules. Rules systems. Web Semantics Sci Services Agents World Wide Web 3(1):41–60
25. Ortony A, Clore G, Collins A (1988) The cognitive structure of emotions. Cambridge University Press, New York
26. Ortony A, Turner T (1990) 'What's basic about basic emotions? Psychol Rev 97:315–331
27. Parrott W (2001) Emotions in social psychology: essential readings. Psychology Press, Philadelphia
28. Peinado F (n.d.) DLModel, a tool for dealing with description logics. http://www.federicopeinado.com/projects/dlmodel/
29. Plutchik R (1980) A general psychoevolutionary theory of emotion. In: Plutchik R, Kellerman H (eds) Emotion: theory, research, and experience, vol 1. Theories of emotion. Academic Press, New York, pp 3–33
30. Racer-Systems (n.d.) Racerpro. http://www.racer-systems.com/
31. Russell J (1980) A circumflex model of affect. J Pers Soc Psychol 39:1161–1178
32. Scherer KR (1984) On the nature and function of emotion: a component process approach. In: Scherer K, Ekman P (eds) Approaches to emotion. Erlbaum, Hillsdale, pp 293–317
33. Schrder M. (2004) Dimensional emotion representation as a basis for speech synthesis with non-extreme emotions. In: Proceedings of workshop on affective dialogue systems. Kloster Irsee, Germany
34. Wang P, Hu J, Zeng H-J, Chen Z (2008) Using wikipedia knowledge to improve text classification. Knowl Inform Syst (online first)

Author Biographies

Virginia Francisco (Ph.D.) works as a Teaching Assistant (Profesor Ayudante) at the Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense de Madrid. She is also a member of the Instituto de Tecnología del Conocimiento (http://www.itc.ucm.es). Her research is oriented towards emotions and the importance of emotions in any attempt to make computer interfaces respond in some way to the affective nature of the content they are handling. Virginia Francisco also works in the field of Dependency Analysis, exploring how this analysis could be used to determine the ambit of negation and exploring how increase the currency of emotional systems.

Pablo Gervás works as associate professor (profesor titular de universidad) at the Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense de Madrid. He is the director of the NIL research group (http://nil.fdi.ucm.es) and also of the Instituto de Tecnología del Conocimiento (http://www.itc.ucm.es). Over the years, Pablo Gervás has worked on the development of expert systems, information retrieval, user modelling, knowledge representation, case-based reasoning, affective computing, processing natural language input, generating natural language output, building resources for related tasks, and generating stories. At present, his main topic of research is studying the role of stories in human communication and extending it to human-computer interaction. This task brings together his previous research lines on knowledge representation, identification of emotions, user modelling and analysis and generation of text.

Federico Peinado works as Ph.D. Teaching Assistant (Profesor Ayudante Doctor) at the Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática, Universidad Complutense de Madrid. He is a member of the NIL research group (http://nil.fdi.ucm.es) and also of the Instituto de Tecnología del Conocimiento (http://www.itc.ucm.es). Federico Peinado's research interest focus on Artificial Intelligence, Computational Narrative and Entertainment Computing. He has worked on Knowledge Representation, developing ontologies and technologies for the Semantic Web, and applying Case-Based Reasoning techniques for Story Generation, Interactive Digital Storytelling and Computational Creativity. Federico has given lectures and published articles on many fields as Affective Computing and Computer Game Design.