An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus

Carlos Iván Chesñevar[1] **and Ana Gabriela Maguitman**[2]

Abstract. In spite of the significant evolution of spelling and grammar checkers for word-processing software, the problem of judging the appropriateness of language usage in different contexts remains to a large extent still unsolved. This paper presents a novel, argumentative approach to providing proactive assistance for language usage assessment on the basis of the web linguistic corpus. A defeasible argumentation system determines if a given expression is ultimately acceptable by analyzing a defeasible logic program which encodes the user's preferences. Those expressions assessed as unsuitable are further inspected automatically to help the user make the necessary repairs.

1 INTRODUCTION AND MOTIVATIONS

Although spelling and grammar checkers for word-processing software have helped to significantly reduce the overall amount of burden for checking documents, the problem of judging the appropriateness of language usage in different contexts remains to a large extent still unsolved. One effective solution to this problem is to provide the user with information about frequencies of natural language expressions in different contexts. Such systems, called *concordance programs* [6, 11], have become particularly powerful with the evolution of the web, as there is a huge collection of text-based information available online.

In linguistics, the so-called *language usage patterns* aim to analyze natural language expressions by means of surveys of different kinds classifying language patterns according to several possible criteria. Such surveys are carried out on adequate samples sizes to be representative for performing statistical inference for assessing and evaluating features of language usage. Following the same principle, most concordance programs provide frequency results obtained from thousands of web documents which form the so-called *Web language corpus* or just *Web corpus*.

Absolute frequencies of natural language expressions can be the source of valuable information only after the end user performs some measured analysis. Consider, for example, a journalist who requires assessment on a particular expression E for a news report. The fact that E has a high absolute frequency in web documents does not imply that E is acceptable, as it might be a regionalism belonging to a particular country. Clearly, such analysis is *defeasible*, as a reason to adopt a given language pattern as valid may be defeated in the light of additional information.

This paper presents a novel approach to providing proactive assistance for language usage assessment combining web-based linguistic corpora and defeasible argumentation. Textual expressions are extracted from the user's document and evaluated with respect to usage indices, which are good indicators of the suitability of an expression on the basis of the current web corpus. A defeasible argumentation system determines if a given expression is acceptable by analyzing a defeasible logic program which encodes the user's preferences. Those expressions assessed as unsuitable are further inspected automatically by the system to help the user make the necessary repairs.

2 THE WEB AS A LINGUISTIC CORPUS

A huge amount of sample sentences in different natural languages have been accumulated as part of Web documents on the Internet. Most of such documents are accessible through search engines, whose pattern-matching capabilities have turned out to be useful for using the web space as a linguistic corpus, also called *Web-Corpus* [8]. Such web corpus offers a number of advantages in comparison with traditional linguistic corpora [14]. In order to analyze relevant features of language usage patterns in web-based corpora, values associated with absolute or relative frequencies of string patterns wrt different web domains turn out to be particularly useful. We call such values *usage indices*. Such usage indices can be easily computed on the basis of advanced search facilities provided by most search engines (e.g. GOOGLE).

Next we introduce some definitions to formalize this concept. In the sequel, strings will be denoted with lowercase letters $s, t, u, \ldots$, possibly subscripted. We will use $d_1, d_2, \ldots$ to denote different web domains. Throughout this paper, the term *domain* will be used indistinctly to refer to complete web domain names (e.g., 'google.com') as to suffix portion of web domain names (e.g., '.com'). The distinguished constant name Web will be used to characterize the collection of all existing web domains. Given a domain d, we will use $\|d\|$ to denote the number of web pages found in the domain d. This notation can be extended to a set of domains $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ as $\|\mathcal{D}\| = \sum_{i=1}^{k} \|d_i\|$. [3] Similarly, given a domain d and a string s, we will use $\|d\|_s$ to denote the number of web pages in domain d containing the string s. [4]

[1] Dept. of Computer Science – Universitat de Lleida – C/Jaume II, 69 – E-25001 Lleida, SPAIN – TEL/FAX:(+34)(973)702764/2702 – EMAIL: cic@eup.udl.es

[2] Computer Science Department - Indiana University, Bloomington, IN 47405-7104, USA – Computer Science Department, Indiana University. Bloomington, IN 47405-7104, USA - Email: anmaguit@cs.indiana.edu.

[3] In the sequel, we will assume that domain names included in a domain set do not overlap, i.e. given a set of domains $\mathcal{D} = \{d_1, \ldots, d_k\}$ they satisfy that if $i \neq \jmath$ then d_i is not a suffix domain of d_j. In addition, we will assume that all domains contain at least one web page.

[4] The special syntax *site:*, available in certain search engines (e.g., GOOGLE), restricts the search to a specified domain, allowing to obtain an estimation of $\|d\|_s$ and $\|d\|$ by posing the queries 's site:d' and 'site:d', respectively.

Usage indices will be based on computing occurrences in sets of domains, as presented next.

Definition 1 (Usage indices U_g, U_c, U_r, and U_p) *Let s be a string, and let $\mathcal{D}$, $\mathcal{D}_1$ and $\mathcal{D}_2$ be non-empty sets of web domains, with $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$. We define the concepts of* general usage U_g, constrained usage U_c, ratio usage U_r, *and* prefix usage U_p *as follows:*

- $U_g(s) =_{def} \|Web\|_s$.
- $U_c(s, \mathcal{D}) =_{def} \|\mathcal{D}\|_s = \sum_{i=1}^{k} \|d_i\|_s$.
- $U_r(s, \mathcal{D}_1, \mathcal{D}_2) =_{def} ((U_c(s, \mathcal{D}_1) + 1)/(U_c(s, \mathcal{D}_2) + 1) \times (\|\mathcal{D}_2\|/\|\mathcal{D}_1\|)$.
- $U_p(s_1, s, \mathcal{D}) =_{def} U_c(s_1 \bullet s, \mathcal{D})/U_c(s, \mathcal{D})$ *if* $U_c(s, \mathcal{D}) \neq 0$, *and* 0 *otherwise.*

Given a string s, the constrained usage $U_c(s, \mathcal{D})$ represents the frequency of pages containing s restricted to the set $\mathcal{D}$ of web domains. The ratio usage $U_r(s, \mathcal{D}_1, \mathcal{D}_2)$ represents the ratio of the frequency of pages with s in $\mathcal{D}_1$ to the frequency of pages with s in $\mathcal{D}_2$. Finally, the prefix usage U_p informs about the likelihood of finding a string s_1 immediately preceding another string s in a page from some domain in $\mathcal{D}$.

Example 2 *Consider the strings s_1=rearing children, s_2=parents, and s_3=of twins. Let d_1=`'.uk'` and d_2=`'.babycentre.co.uk'`. Then it holds that $\|Web\|$ = 3307998701, $\|\{d_1\}\|$ = 28000000, $U_c(s_1, \{d_1\}) = 435$, $U_c(s_1, Web) = 13700$, $U_r(s_1, \{d_1\}, Web)$=$(436/13701) * (3307998701/28000000) = 3.76$, and $U_p(s_2, s_3, \{d_2\}) = 677/747 = 0.906$.*

Note in the above example that statistical inference can be performed from usage indices (e.g. 90% of occurrences of the phrase of twins in `'.babycentre.co.uk'` are preceded by the word parents). Note also that the above computations are time-dependent (as they depend on the current Web corpus).

3 DEFEASIBLE LOGIC PROGRAMMING: FUNDAMENTALS

Defeasible argumentation [3, 10] has evolved in the last decade as a successful approach to formalize defeasible, commonsense reasoning. Argument-based applications have been developed in many areas, such as agent theory, web recommendation [2], knowledge engineering and legal reasoning [1, 3]. *Defeasible logic programming* (DeLP) [7] is a defeasible argumentation formalism based on logic programming. A defeasible logic program is a set $K = (\Pi, \Delta)$ of Horn-like clauses, where Π and Δ stand for sets of strict and defeasible knowledge, respectively. The set Π of strict knowledge involves *strict rules* of the form $p \leftarrow q_1, \ldots, q_k$ and *facts* (strict rules with empty body), and it is assumed to be *non-contradictory*. The set Δ of defeasible knowledge involves *defeasible rules* of the form $p \mathrel{-\!\!\prec} q_1, \ldots, q_k$, which stands for "$q_1, \ldots q_k$ provide a *tentative reason* to believe p." The underlying logical language is that of extended logic programming, enriched with a special symbol "$\mathrel{-\!\!\prec}$" to denote defeasible rules. Both default and classical negation are allowed (denoted not and $\sim$, resp.). Syntactically, the symbol "$\mathrel{-\!\!\prec}$" is all that distinguishes a *defeasible* rule $p \mathrel{-\!\!\prec} q_1, \ldots q_k$ from a *strict* (non-defeasible) rule $p \leftarrow q_1, \ldots, q_k$. DeLP rules are thus Horn-like clauses to be thought of as *inference rules* rather than implications in the object language. Deriving literals in DeLP results in the construction of *arguments*. An argument $\mathcal{A}$ is a (possibly empty) set of ground defeasible rules that together with the set Π provide a logical proof for a given literal h, satisfying the additional requirements of *non-contradiction* and *minimality*.

Definition 3 (Argument) *Given a DeLP program $\mathcal{P}$, an* argument *$\mathcal{A}$ for a query q, denoted $\langle \mathcal{A}, q \rangle$, is a subset of ground instances of defeasible rules in $\mathcal{P}$ and a (possibly empty) set of default ground literals "not L", such that: 1) there exists a* defeasible derivation *for q from $\Pi \cup \mathcal{A}$; 2) $\Pi \cup \mathcal{A}$ is non-contradictory (i.e, $\Pi \cup \mathcal{A}$ does not entail two complementary literals p and $\sim p$ (or p and not p)), and 3) $\mathcal{A}$ is minimal with respect to set inclusion.*
An argument $\langle \mathcal{A}_1, Q_1 \rangle$ is a sub-argument *of another argument $\langle \mathcal{A}_2, Q_2 \rangle$ if $\mathcal{A}_1 \subseteq \mathcal{A}_2$. Given a DeLP program $\mathcal{P}$, $Args(\mathcal{P})$ denotes the set of all possible arguments that can be derived from $\mathcal{P}$.*

The notion of defeasible derivation corresponds to the usual query-driven SLD derivation used in logic programming, performed by backward chaining on both strict and defeasible rules; in this context a negated literal $\sim p$ is treated just as a new predicate name no_p. Minimality imposes a kind of 'Occam's razor principle' [12] on arguments. The non-contradiction requirement forbids the use of (ground instances of) defeasible rules in an argument $\mathcal{A}$ whenever $\Pi \cup \mathcal{A}$ entails two complementary literals.

Definition 4 (Counterargument – Defeat) *An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a* counterargument *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ iff*

1. *There is an subargument $\langle \mathcal{A}, q \rangle$ of $\langle \mathcal{A}_2, q_2 \rangle$ such that the set $\Pi \cup \{q_1, q\}$ is contradictory.*
2. *A literal* not q_1 *is present in some rule in $\mathcal{A}_1$.*

A partial order $\preceq \subseteq Args(\mathcal{P}) \times Args(\mathcal{P})$ will be used as a preference criterion *among conflicting arguments. An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a* defeater *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ if $\langle \mathcal{A}_1, q_1 \rangle$ counterargues $\langle \mathcal{A}_2, q_2 \rangle$, and $\langle \mathcal{A}_1, q_1 \rangle$ is preferred over $\langle \mathcal{A}_2, q_2 \rangle$ wrt $\preceq$. For cases (1) and (2) above, we distinguish between* proper *and* blocking defeaters *as follows:*

- *In case 1, the argument $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* proper defeater *for $\langle \mathcal{A}_2, q_2 \rangle$ iff $\langle \mathcal{A}_1, q_1 \rangle$ is strictly preferred over $\langle \mathcal{A}, q \rangle$ wrt $\preceq$.*
- *In case 1, if $\langle \mathcal{A}_1, q_1 \rangle$ and $\langle \mathcal{A}, q \rangle$ are unrelated to each other, or in case 2, $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, q_2 \rangle$.*

Specificity [12] is used in DeLP as a syntax-based criterion among conflicting arguments, preferring those arguments which are *more informed* or *more direct* [12, 13]. However, other alternative partial orders could also be used.

An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0 \rangle$ (denoted $\lambda^{\langle \mathcal{A}_0, q_0 \rangle}$) is a sequence $[\langle \mathcal{A}_0, Q_0 \rangle, \langle \mathcal{A}_1, Q_1 \rangle, \langle \mathcal{A}_2, Q_2 \rangle, \ldots, \langle \mathcal{A}_n, Q_n \rangle \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). Each $\langle \mathcal{A}_i, Q_i \rangle$ is a defeater for the previous argument $\langle \mathcal{A}_{i-1}, Q_{i-1} \rangle$ in the sequence, $i > 0$. In order to avoid *fallacious* reasoning, dialectics imposes additional constraints on such an argument exchange to be considered rationally acceptable in a program $\mathcal{P}$. These constraints involve disallowing repetition of arguments in argumentation lines (circular argumentation), requiring that the set of arguments belonging to proponent (resp. opponent) be non-contradictory and enforcing the use of stronger arguments to defeat arguments acting as blocking defeaters.[5]

An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite [7]. Given a DeLP program $\mathcal{P}$ and an initial argument $\langle \mathcal{A}_0, Q_0 \rangle$, the set of all acceptable

[5] For an in-depth treatment of dialectical constraints in DeLP the reader is referred to [7].

argumentation lines starting in $\langle \mathcal{A}_0, Q_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0 \rangle$ (ie., all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0 \rangle$), formalized as a *dialectical tree*.

Definition 5 (Dialectical Tree) *Let $\mathcal{P}$ be a DeLP program, and let $\langle \mathcal{A}_0, Q_0 \rangle$ be an argument in $\mathcal{P}$. A* dialectical tree *for $\langle \mathcal{A}_0, Q_0 \rangle$, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$, is a tree structure defined as follows:*

1. *The root node of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ is $\langle \mathcal{A}_0, Q_0 \rangle$.*
2. *$\langle \mathcal{B}', H' \rangle$ is an immediate children of $\langle \mathcal{B}, H \rangle$ iff there exists an acceptable argumentation line $\lambda^{\langle \mathcal{A}_0, Q_0 \rangle}$ = [$\langle \mathcal{A}_0, Q_0 \rangle$, $\langle \mathcal{A}_1, Q_1 \rangle$, ..., $\langle \mathcal{A}_n, Q_n \rangle$] such that there are two elements $\langle \mathcal{A}_{i+1}, Q_{i+1} \rangle$ = $\langle \mathcal{B}', H' \rangle$ and $\langle \mathcal{A}_i, Q_i \rangle$ = $\langle \mathcal{B}, H \rangle$, for some $i = 0 \ldots n-1$.*

Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ can be marked as *undefeated* and *defeated* nodes (U-nodes and D-nodes, resp.). A dialectical tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as *D-node* iff it has at least one U-node as a child, and as *U-node* otherwise. An argument $\langle \mathcal{A}_0, Q_0 \rangle$ is ultimately accepted as valid (or *warranted*) wrt a DeLP program $\mathcal{P}$ iff the root of its associated dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ is labeled as *U-node*.

Given a DeLP program $\mathcal{P}$, solving a query q wrt $\mathcal{P}$ accounts for determining whether q is supported by a warranted argument. Different doxastic attitudes are distinguished when answering q according to the associated status of warrant, in particular: (1) Believe q (resp. $\sim q$) when there is a warranted argument for q (resp. $\sim q$) that follows from $\mathcal{P}$; (2) Believe q is *undecided* whenever neither q nor $\sim q$ are supported by warranted arguments in $\mathcal{P}$. It should be noted that that the computation of warrant cannot lead to contradiction [7]: if there exists a warranted argument $\langle A, h \rangle$ on the basis of a program $\mathcal{P}$, then there is no warranted argument $\langle B, \sim h \rangle$ based on $\mathcal{P}$.

4 ASSESSING LANGUAGE USAGE USING WEB CORPORA AND DEFEASIBLE ARGUMENTATION

Although the Web corpus provides very useful resources for language usage assessment on the basis of the relative and absolute frequencies in web documents, coming up with suggestions about language patterns requires a meta level analysis from the end user, who must perform an additional inference process based on such frequency values. Let us consider again the case of the journalist presented in the introduction, who thinks that a given expression E is not suitable for a news report intended for a Spanish newspaper, as he suspects that E is a regionalism e.g. from Argentina. This last assumption can be supported on the basis of the ratio $R = U_r(E, \{\texttt{'.ar'}\}, \{\texttt{'.es'}\})$. The fact that $R > 1$ provides a *tentative* reason for concluding that E is a regionalism associated with Argentina. Knowing that E is already in use in other Spanish newspaper may make the journalist change his mind, as he would have a reason that *defeats* the previous assumption. Once again, the above situation can be captured by computing $R' = U_c(E, \mathcal{D}_{news})$, where $\mathcal{D}_{news}$ corresponds to a set of domains corresponding to the Spanish mass media. The fact that $R' > \theta$, where θ is a particular threshold value, provides a reason to think that E is a common expression in the Spanish mass media, and therefore it can be used.

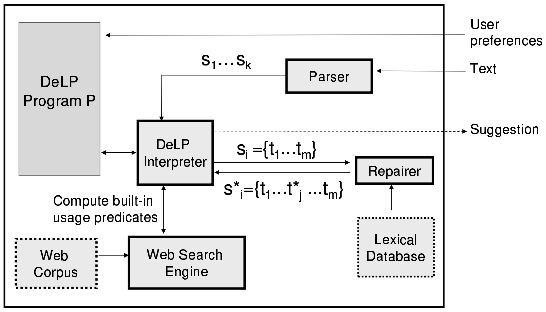

Figure 1. Framework Outline

Our proposal aims at modeling the kind of analysis described above by integrating a front-end parser for the text entered by the user with a DeLP interpreter, which provides recommendations by solving queries on the basis of usage indices. An outline of the proposed approach is shown in Fig. 1. Given a text T corresponding to a user document, a front-end parser extracts a list T' = [s_1, s_2, ..., s_k] of syntactic elements from T. Every $s_i \in T'$ is analyzed wrt a DeLP program $\mathcal{P}$, which encodes criteria for language usage in terms of strict and defeasible rules. Rules in $\mathcal{P}$ may include references to built-in predicates U_g, U_c, U_r and U_p which stand for usage indices as presented in Def. 1. A distinguished predicate name $solve$ will be used for analyzing the *acceptability* of every expression s_i with respect to language usage criteria specified in $\mathcal{P}$. Program $\mathcal{P}$ contains the definition of a predicate called acc, which is used to evaluate the acceptability of its argument expression. Thus, the existence of a warranted argument $\langle A, acc(s_i) \rangle$ built on the basis of $\mathcal{P}$ will allow to conclude that s_i is an acceptable expression. Similarly, the existence of a warranted argument $\langle A, \sim acc(s_i) \rangle$ indicates that s_i is *not* acceptable.

An interesting feature in automated systems for language assessment is the possibility of suggesting *repairs* whenever a particular user expression seems not suitable. This sort of funcionality can be embedded in the proposed framework by means of a specialized predicate $repair$. Should an expression s_i be assessed as unacceptable, then $repair$ can be used to seek for alternatives. An expression s_{new} is a potential repair for s_i if s_{new} is the result of replacing some words in s_i by synonyms found in a lexical database (e.g. WordNet [5]). If a warranted argument $\langle A, acc(s_{new}) \rangle$ is built on the basis of $\mathcal{P}$, then s_{new} is presented to the user as a possible alternative to s_i. This process is outlined in the algorithm of Fig. 2.

```
ALGORITHM ProvideAssessment
INPUT: Text $T$, DeLP program $\mathcal{P}$ modeling user preferences
OUTPUT:  Assessment on $T$ {according to web corpus and $\mathcal{P}$}
         Suggest repairs when necessary
         {according to web corpus, lexical database and $\mathcal{P}$}
BEGIN
 Compute $T'$ = $[s_1, s_2, \ldots s_k]$ on the basis of $T$
 {$T'$ results from parsing $T$. Every $s_i$ is a piece of text.}
 FOR EVERY $s_i \in T'$
  DO {try to solve $s_i$}
   Solve query $acc(s_i)$ based on $\mathcal{P}$ and web corpus
   IF $acc(s_i)$ is warranted
   THEN Do nothing {assume $s_i$ is correct.}
   ELSE
      Solve query $\sim acc(s_i)$ based on $\mathcal{P}$ and web corpus
      IF $\sim acc(s_i)$ is warranted
     THEN {search for repairs}
        REPEAT
           Let $s'_i$ be a new candidate repair for $s_i$
           IF $acc(s'_i)$ is warranted
           THEN Suggest $s'_i$ as an alternative
        UNTIL (Repair $s'_i$ found) or (no more repairs available)
      ELSE {neither $acc(s_i)$ nor $\sim acc(s_i)$ holds}
        there is no suggestion about $s_i$
END
```

Figure 2. High-level algorithm for providing language usage assessment using defeasible argumentation

5 A WORKED EXAMPLE

Consider the case of an American journalist who writes articles in Spanish about Latinamerican issues, intended for audiences in Spain and Argentina. As Spanish is not his mother tongue, he usually makes mistakes related to properly assessing the correct language usage. A sample paragraph from such a journalist (and its corresponding English translation) could be as follows:

> *"El corralito fue un fenómeno muy complejo [...] Para el colectivo de los trabajadores autónomos cualquier liviano error tenía consecuencias [...]."*
> *"The "corralito"[6] was a very complex phenomenon [...] For the syndicate of autonomous workers any *light* mistake had consequences [...]."*

Let us assume that the editor of the newspaper will check every article written by our journalist before it is sent to print, guided by a number of criteria which characterize a "well-written document". In the above text some anomalous situations will be detected: "corralito" is a common term in Argentina, but not so common in Spain (except in the news). The expression "colectivo de trabajadores autónomos" (syndicate of autonomous workers) has a clear meaning in Spain, but is not understood in Argentina (as "gremio" is the Argentinean equivalent for "colectivo"). The noun phrase "liviano error" is wrong in Spanish language, as the correct fixed idiom would be "ligero error" (=light mistake), even though the adjectives "ligero" and "liviano" are synonyms. Some of the possible criteria the editor could apply to avoid such anomalies can be characterized in terms of the DeLP program shown in Fig. 3.

Rules 1 to 4 characterize the behavior of the $solve$ predicate as outlined in Section 4. Rule 5 defines the $repair$ predicate restricted to simple noun phrases of the form $[Noun, Adj]$. Repairs consist in just replacing Adj for an alternative synonym obtained from an ad-hoc predicate syn (Rule 6).[7] For the sake of simplicity, in this example the definition of synonym is restricted to the Spanish adjective liviano ("light"). Defeasible rules 7 to 12 capture language usage preferences on the basis of usage indices computed in rules 13 to 15. Rule 7 establishes that strings whose general frequency in Spanish speaking countries is above a certain threshold value are defeasibly acceptable. From Rule 8 it follows that strings which cannot be proven to be common in web domains from Spanish speaking countries are usually not acceptable. Rules 9 and 10 establish that regionalisms from Argentina and Spain are usually not acceptable. Rule 11 specifies when a given expression can be defeasibly assumed to be a regionalism in terms of its frequency, computed using the $locally\_freq$ predicate. Rule 12 provides an exception for the above rule: a string S which is locally frequent in Argentina but is also frequent in the Spanish media is not considered to be a regionalism. A string s is considered frequent in the Spanish media if a considerable percentage of all the hits found for s in Spain are found in newspapers. Rule 18 specifies that Spanish speaking countries to be considered for the analysis are Spain and Argentina.[8]

Control rules for language usage assessment:

1) $solve(S) \leftarrow acc(S),$
   $write('\text{Acceptable}').$
2) $solve(S) \leftarrow\ \sim acc(S), repair(S,R),$
   $acc(R),$
   $write('\text{Acc. if rephrased as}', R).$
3) $solve(S) \leftarrow\ \sim acc(S),$
   $write('\text{Not acceptable}').$
4) $solve(\_) \leftarrow write('\text{Undecided. No suggestion found}').$
5) $repair(S,R) \leftarrow simple\_nphrase(S), S = [Noun, Adj],$
   $syn(Adj, NAdj), R = [Noun, NAdj].$
6) $syn(Adj, NAdj) \leftarrow list\_syn(Adj, L), member(NAdj, L).$

Defeasible rules capturing language usage preferences:

7) $acc(S) \mathrel{-\!\!\!\prec} common\_in\_spanish(S).$
8) $\sim acc(S) \mathrel{-\!\!\!\prec} rare\_in\_spanish(S).$
9) $\sim acc(S) \mathrel{-\!\!\!\prec} common\_in\_spanish(S),$
   $regionalism(S, [\text{'.ar'}]).$
10) $\sim acc(S) \mathrel{-\!\!\!\prec} common\_in\_spanish(S),$
   $regionalism(S, [\text{'.es'}]).$
11) $regionalism(S, Ctry) \mathrel{-\!\!\!\prec} locally\_freq(S, Ctry).$
12) $\sim regionalism(S, [\text{'.ar'}]) \mathrel{-\!\!\!\prec} locally\_freq(S, Ctry),$
   $appears\_in\_news(S, \text{'.es'}).$

Predicates based on computing Usage Indices:

13) $common\_in\_spanish(S) \leftarrow spanish\_speaking(Cs),$
   $V \text{ is } U_c(S, Cs), V > 200.$
14) $rare\_in\_spanish(S) \leftarrow \mathsf{not}\ common\_in\_spanish(S).$
15) $appears\_in\_news(S, C) \leftarrow news\_domains(Ds, C),$
   $V \text{ is } U_c(S, Ds), V > 200.$
16) $locally\_freq(S, [\text{'.ar'}]) \leftarrow V \text{ is } U_r(S, C, [\text{'.es'}]), V > 10$
17) $locally\_freq(S, [\text{'.es'}]) \leftarrow V \text{ is } U_r(S, C, [\text{'.ar'}]), V > 10$

Additional predicates:

18) $news\_domains$(['elmundo.es', 'elpais.es'], '.es').
19) $spanish\_speaking$(['.es', '.ar').
20) $list\_syn$(liviano, [ligero, sutil, ...]).
21) $member(X, [X|\_]).$
22) $member(X, [Y|Z]) \leftarrow member(X, Z).$
23) $simple\_nphrase(S) \leftarrow$[computed elsewhere].

Figure 3. DeLP program modeling preference criteria for acceptable language usage patterns in newspaper articles

Suppose we apply now the high-level algorithm presented in Fig. 2, where the strings extracted from the above text are s_1, s_2 and s_3, with s_1=corralito, s_2=colectivo de los trabajadores autónomos, and s_3=liviano error. Consider the case for string s_1. The search for a warranted argument for $acc(s_1)$ returns $\langle \mathcal{A}_1, acc(s_1) \rangle$, with $\mathcal{A}_1$={ $acc(s_1) \mathrel{-\!\!\!\prec} common\_in\_spanish(s_1)$ }. This argument holds since U_c(s_1,['.es','.ar'])$>$ 200.[9] The DeLP inference engine will then search for defeaters for $\langle \mathcal{A}_1, acc(s_1) \rangle$. A proper defeater $\langle \mathcal{A}_2, \sim acc(s_1) \rangle$ is found: s_1 is not acceptable as there are reasons to think it is a regionalism from Argentina. Here we have $\mathcal{A}_2$={ $\sim acc(s_1) \mathrel{-\!\!\!\prec} common\_in\_spanish(s_1), regionalism(s_1, \texttt{'.ar'})$; $regionalism(s_1, \texttt{'.ar'}) \mathrel{-\!\!\!\prec} locally\_freq(s_1, \texttt{'.ar'})$}. [10] Note that $\langle \mathcal{A}_2, \sim acc(s_1) \rangle$ is a proper defeater for $\langle \mathcal{A}_1, acc(s_1) \rangle$ as the first argument is based on more specific information than the second. Note also that predicate $locally\_freq(s_1, \texttt{'.ar'})$ holds, as U_r(s_1,[$\texttt{'.ar'}$],[$\texttt{'.es'}$])$=$ 33.1 $>$ 10. A defeater for this argument can be found on its turn: corralito is not a regionalism in Argentina as it is fairly frequent in the Spanish news. Here we have the argument $\mathcal{A}_3$={ $\sim regionalism(s_1, [\text{'.ar'}]) \mathrel{-\!\!\!\prec} locally\_freq(s_1, \texttt{'.ar'})$, $appears\_in\_news(s_1, \text{'es'})$}. Note that predicate $appears\_in\_news(s_1, \text{spain})$ holds, as $U_c(\text{corralito}, \mathcal{D})$=40, with $\mathcal{D}$ representing domains from Span-

[6] The term "corralito" (little baby crib) was coined in Argentina in Dec. 2001 to denote severe restrictions on money drawing from banks due to an economic crisis in the country. The term became popular as mass media from different Spanish-speaking countries (including Spain) reported about the economic situation in Argentina, becoming hence an expression used to refer to an "abnormal situation in which customers are not allowed to draw their money from a bank for a long period of time".

[7] A lexical database such as WordNet [5] can provide a list of synonyms (synset) for an arbitrary adjective.

[8] For the sake of simplicity, in this example we restrict our analysis to only these two countries (Spain and Argentina), and we only focus on exceptions for regionalisms in Argentina based on sample Spanish news domains.

[9] Computations of usage indices were performed using Google with the existing Web corpus on Feb. 15, 2004. For space reasons, the detailed computation of usage indices is not included.

[10] For the sake of clarity, we use semicolons to separate elements in an argument $\mathcal{A}$ = $\{e_1 ; e_2 ; \ldots ; e_k\ \}$.

ish newspapers. Note also that the definition of dialectical tree (Def. 5) does not allow the use of $\langle \mathcal{A}_1, acc(s_1)\rangle$ to defeat again $\langle \mathcal{A}_2, \sim acc(s_1)\rangle$, as this would imply falling into *fallacious*, circular argumentation. After the above analysis no other defeater can be found. The resulting dialectical tree rooted in $\langle \mathcal{A}_1, acc(s_1)\rangle$ as well as its corresponding marking is shown in Fig. 4a. The root node is marked as U-node (undefeated), which implies that the argument $\langle \mathcal{A}_1, acc(s_1)\rangle$ is warranted.

Consider now the case for string s_2=colectivo de los trabajadores autónomos. There is an argument $\langle \mathcal{B}_1, acc(s_2)\rangle$, with $\mathcal{B}_1$=$\{acc(s_2) \mathrel{-\!\!\prec} common\_in\_spanish(s_2)\ \}$ which holds following the same reasoning as above. However, there is a defeater for $\langle \mathcal{B}_1, acc(s_2)\rangle$, namely $\langle \mathcal{B}_2, \sim acc(s_2)\rangle$, with $\mathcal{B}_2$=$\{\sim acc(s_2) \mathrel{-\!\!\prec} common\_in\_spanish(s_2),\ regionalism(s_2, [\text{'.es'}]);$ $regionalism(s_2, \text{'.es'}) \mathrel{-\!\!\prec} locally\_freq(s_2, \texttt{'.es'})\}$. As above, predicate $locally\_freq(s_2, \texttt{'.es'})$ holds, as it is the case that $U_r(s_2$,[$\texttt{'.es'}$],[$\texttt{'.ar'}$])$= 41.4$. No other arguments can be computed from here onwards. The $solve$ predicate will thus fire the search for a warranted argument for $\sim acc(s_2)$, which is successful (a dialectical tree rooted in $\langle \mathcal{B}_2, \sim acc(s_2)\rangle$ with no defeaters). The resulting situation is shown in Fig. 4b. Note that no repair is possible here, as $repair$ is only for simple noun phrases.

Finally, let us consider the case for the string s_3=liviano error. There is no argument (and consequently no warranted argument) for $acc(s_3)$, as $common\_in\_spanish(s_3)$ does not hold: s_3 is syntactically correct but is pragmatically wrong as noun phrase in Spanish. In contrast, there is a warranted argument $\langle \mathcal{C}_1, \sim acc(s_3)\rangle$ which provides a reason *not* to accept s_3, based on rule 8, with $\mathcal{C}_1$=$\{\sim acc(s_3) \mathrel{-\!\!\prec} rare\_in\_spanish(s_3)\}$. The predicate $solve$ will try to repair s_3, obtaining a new alternative string s_3' = ligero error, searching then for a warranted argument for $acc(s_3')$. A warranted argument for $acc(s_3')$ can be found, namely $\mathcal{D}_1$=$\{\ acc(s_3') \mathrel{-\!\!\prec} common\_in\_spanish(s_3')\}$. As a side effect, the message "Accepted if rephrased as ligero error" will be given to the user. This situation is shown in Fig. 4c.

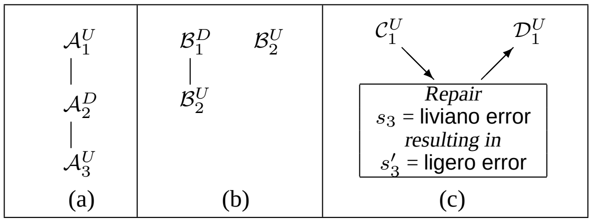

Figure 4. Dialectical trees associated with (a) $\langle \mathcal{A}_1, acc(s_1)\rangle$ (b) $\langle \mathcal{B}_1, acc(s_2)\rangle$ and $\langle \mathcal{B}_2, \sim acc(s_2)\rangle$; (c) $\langle \mathcal{C}_1, \sim acc(s_3)\rangle$ and $\langle \mathcal{D}_1, acc(s_3')\rangle$

6 RELATED WORK. CONCLUSIONS

Providing assessment in word-processing activities has long been a source of research in the natural language processing community [9]. The term *critiquing system* is the common denomination for those cooperative tools that observe the user interacting with a word-processing tool and present reasoned opinions about the typed document, helping to discover and point out errors that might otherwise remain unnoticed. Most popular word-processing critiquing systems include spelling-, grammar-, and style-checkers [4].

In the last years some word-processing critiquing systems evolved towards the analysis of language usage patterns, taking advantage of the rich source of textual material that the Web offers as a linguistic corpus [11, 14]. Several concordancers and writing assistant tools were developed (e.g. WebLEAP [14], WebCorp [11] and KWICFinder [6]). Such systems provide recommendations on language pattern on the basis of frequency values found on the Web corpus, including also advanced facilities for restricting search to particular domains and finding grammatical patterns. In such systems, the ultimate analysis of a language pattern is to be done by the end user.

In this paper we have presented a novel approach in which the preceding analysis is automated on the basis of usage indices (computed from the current Web corpus) and a defeasible argumentation framework. Preference criteria for language usage can be formalized by the user in a declarative manner in terms of defeasible and strict rules. To the best of our knowledge, no similar approach has been developed to support the assessment of natural language usage.

Performing defeasible argumentation is a computationally complex task. For an efficient implementation of DeLP an extension of the WAM (Warren's Abstract Machine) for Prolog has been developed [7]. Several features leading to improving computational aspects of DeLP have also been recently studied (e.g. optimizing the comparison arguments by specificity [13]).We contend that the evolution of automated systems for language processing will result in sophisticated environments, in which an appropriate assessment of language usage patterns will play a major role. We believe that the proposed approach is a first step to help fulfill this long-term goal.

ACKNOWLEDGEMENTS

This research was partially supported by CYCIT Projects TIC2001-1577-C03-01/03 and TIC2003-00950, by Ramón y Cajal Program (MCyT, Spain) and by CONICET (Argentina). The authors would like to thank anonymous reviewers for providing helpful comments to improve the final version of this paper.

REFERENCES

[1] D. Carbogim, D. Robertson, and J. Lee, 'Argument-based applications to knowledge engineering', *The Knowledge Engineering Review*, **15**(2), 119–149, (2000).

[2] C. Chesñevar and A. Maguitman, 'ArgueNet: An Argument-Based Recommender System for Solving Web Search Queries', in *Proc. Intl. IEEE Conference on Intelligent Systems. Varna, Bulgaria*, (2004).

[3] C. Chesñevar, A. Maguitman, and R. Loui, 'Logical Models of Argument', *ACM Computing Surveys*, **32**(4), 337–383, (December 2000).

[4] K. Church and L. Rau, 'Commercial applications of natural language processing', *CACM*, **38**(11), 71–79, (November 1995).

[5] C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[6] W. Fletcher, 'Concordancing the Web with KWiCFinder', in *Proc. 3rd North American Symposium on Corpus Linguistics and Language Teaching*, (2001).

[7] A. García and G. Simari, 'Defeasible Logic Programming: An Argumentative Approach', *Theory and Practice of Logic Programming*, **4**(1), 95–138, (2004).

[8] A. Kilgarriff, 'Web as Corpus', in *Proc. Corpus Linguistic Conf.*, pp. 342–344. UCREL-Lancaster Univ, UK, (2001).

[9] K. Kukich, 'Techniques for automatically correcting words in text', *ACM Computing Surveys*, **24**(4), 377–439, (1992).

[10] H. Prakken and G. Vreeswijk, 'Logical Systems for Defeasible Argumentation', in *Handbook of Phil. Logic*, eds., D. Gabbay and F.Guenther, 219–318, Kluwer, (2002).

[11] A. Renouf, 'Webcorp: providing a renewable data source for corpus linguists', in *Extending the scope of corpus-based research: new applications, new challenges.*, ed., Granger et.al, 219–318, Rodolpi, (2002).

[12] G. Simari and R. Loui, 'A Mathematical Treatment of Defeasible Reasoning and its Implementation', *Art. Intelligence*, **53**, 125–157, (1992).

[13] F. Stolzenburg, A. García, C. Chesñevar, and G. Simari, 'Computing Generalized Specificity', *J. of Non-Classical Logics*, **13**(1), 87–113, (2003).

[14] T. Yamanoue, T. Minami, I. Ruxton, and Wataru Sakurai, 'Learning Usage of English KWICly with WebLEAP/DSR', in *Proc. 2nd. Intl. Conf. on Inf. Technology and Applications (to appear)*, (2004).