An Argument-based Decision Support System for Assessing Natural Language Usage on the basis of the Web Corpus

Carlos Iván Chesñevar[1,*],
Mariona Sabaté-Carrové[2,†],
Ana Gabriela Maguitman[3,‡]

[1]Departament of Computer Science, Universitat de Lleida,
C/Jaume II, 69 – 25001 Lleida, Spain
[2]Departament of English Studies and Linguistics, Universitat de Lleida,
Pl. Victor Siurana, 1 – 25003 Lleida, Spain
[3]School of Informatics, Indiana University
Bloomington, IN 47406, USA

Abstract

The last decade bears witness to an exponential growth in the use of the World Wide Web. As a result, a huge amount of documents are accessible online through search engines, whose pattern-matching capabilities have turned out to be useful for mining the Web space as a particular kind of linguistic corpus, commonly known as the Web Corpus. This article presents a novel, argumentative approach to providing proactive assistance for language usage assessment on the basis of usage indices, which are good indicators of the suitability of an expression on the basis of the Web Corpus. The user preferences consist of a number of (possibly defeasible) rules and facts which encode different aspects of adequate language usage, defining the acceptability of different terms on the basis of the computed usage indices. A defeasible argumentation system determines if a given expression is ultimately acceptable by analyzing a defeasible logic program which encodes the user's preferences.

1 Introduction and motivations

The last decade has witnessed an exponential growth of the World Wide Web, resulting in a huge amount of documents stored as Web documents. A significant portion of

[*] Author to whom all correspondence should be addressed: cic@eps.udl.es.
[†] e-mail: msabate@dal.udl.es.
[‡] e-mail: anmaguit@cs.indiana.edu.

such documents are accessible through search engines, whose pattern-matching capabilities have turned out to be useful for many non-native speakers of a language who use GOOGLE as a reference for examples of language usage and mine the Web space as a particular kind of linguistic corpus, commonly known as *Web Corpus*.[1−5]

In linguistics, language usage is a vital aspect of language but, unfortunately, traditional dictionaries do not provide much usage information. Language usage patterns are studied and measured by means of surveys in which relevant features are distinguished (e.g. age of the speakers, geographical location, education level, etc). In order for such surveys to be reliable sources of information for statistical inference, the size of the samples considered plays a crucial role. Based on this same principle applied to the Web corpus, search engines can help evaluate the usage of language patterns very efficiently. This fact has been exploited to analyze frequencies of natural language expressions in different contexts by means of automated systems, such as *concordancers* or *concordance programs*.[6,7,3] Such programs have become particularly powerful with the evolution of the Web and provide useful assistance especially for those who have English as a second language (ESL) or English as a foreign language (EFL) or simply for those who need to check the appropriateness of language usage for different language situations and text-types.

Most concordancers (e.g. WEBCORP[6] and KWICFINDER[7]) are based on the presentation of matches of a given pattern obtained by the user within specified constraints (e.g. certain Web domains). Most corpora concordancers supply such a large amount of unclassified information that making use of them often becomes counter-productive in terms of time constrictions. For the linguist, relative and absolute frequencies of distinguished patterns can provide valuable information for assessing language usage. Apart from some on-line statistical corpora (as e.g. `titania.cobuild.collins.co.uk`) which are still rather limited in scope, such information is commonly not available from concordancers, as they offer mostly a “projection” of the Web space according to specifications given by the end user.

Absolute frequencies of natural language expressions can be the source of valuable information only after the end user performs some measured and complex analysis, in which several context-dependent features are taken into account. Consider, for example, a journalist who is uncertain about using a particular term T for a news report written in Spanish, intended for a Spanish-speaking audience in Spain. The fact that the term T has a high absolute frequency in Web documents (e.g. by performing a GOOGLE search query) does not imply *per se* that T is acceptable, as it might have a dialectal use or be a buzz word, that is a vogue word in one particular language community, e.g. Argentina, and consequently the term T should not be used in the news report. However, after analyzing several newspapers our journalist finds out that many Spanish-speaking media in Spain are also making use of this particular term T, so that it can no longer be seen as a dialectal variation from Argentina, but rather as a word with a well-defined meaning for Spanish-speakers in Spain. This last evidence –so thinks our journalist– leads him to believe that the term T can be used in the news report. In epistemological terms the above analysis is said to be *defeasible*[8,9], in that a reason to adopt a given language pattern as valid may be *defeated* in the light of additional information.

This article presents a novel approach to studying language usage patterns based upon *usage indices*, which prove to be reliable indicators of the suitability of a term using the Web corpus at hand. Such indices can be easily computed on the basis of information extracted from the Web by means of advanced search facilities provided by most search engines (e.g. GOOGLE). These facilities allow to restrict the search to certain domains, to search for phrases, or to specify the language for the results. Associated with a set of results, search engines typically supply an estimation of the number of hits for the user query. This information is exploited in our framework to compute indices reflecting the popularity of certain expressions in a particular language or domain. Usage indices provide a means of characterizing *defeasible reasons* for assessing language usage, allowing us to conclude whether a given term T is suitable (or not) in a particular context. This defeasible knowledge will be formalized in terms of Defeasible Logic Programming (DeLP), a logic programming formalism for defeasible argumentation. On the basis of this formalization we define ARGUETERM, an argument-based computational framework which aims to providing proactive assistance for language usage assessment.

The rest of the article is structured as follows. First, Section 2 summarizes the fundamentals of defeasible argumentation theory, an approach for commonsense reasoning which has gained wide acceptability in the Artificial Intelligence community in the last years. We present the central definitions of Defeasible Logic Programming (DeLP) along with a worked example, as this is the particular argumentation framework used in our approach. In Section 3 we introduce the concept of *usage indices*, which provide a way of analyzing different relative and absolute frequencies of distinguished string patterns on the Web. We also characterize three major groups of linguistic situations in which usage indices can be applied, providing a number of examples that illustrate such situations. Section 4 presents ARGUETERM, an argumentative framework for providing assessment on language usage based on usage indices, which are encoded as part of a DeLP program capturing defeasible preferences. We will also present a worked example that illustrates the behavior of the proposed framework. Finally, Section 6 discusses related work and presents the main conclusions obtained.

2 Defeasible Argumentation: Formalizing Knowledge and Commonsense Reasoning about Language Usage

In this section we first summarize some of the main concepts associated with *defeasible argumentation*, and then we present in more detail some characteristics of a particular argument-based formalism called *defeasible logic programming* (DeLP).[10] Finally we discuss how DeLP can be extended to incorporate additional features to model language usage assessment on the basis of the Web Corpus.

2.1 Background

Artificial Intelligence (AI) has long dealt with the challenge of modeling commonsense reasoning, which almost always occurs in the face of incomplete and potentially inconsistent information.[11,12] A logical model of commonsense reasoning demands the formalization of principles and criteria that characterize valid patterns of inference. In this respect, classical logic has proven to be inadequate, since it behaves *monotonically* [1] and cannot deal with inconsistencies at object level.[12]

When a rule supporting a conclusion may be defeated by new information, it is said that such reasoning is *defeasible*.[8,9,13] When we chain defeasible reasons or rules to reach a conclusion, we have *arguments* instead of proofs. Arguments may compete, rebutting each other, so a *process* of argumentation is a natural result of the search for arguments. Adjudication of competing arguments must be performed, comparing arguments in order to determine what beliefs are ultimately accepted as *warranted* or *justified*. Preference among conflicting arguments is defined in terms of a *preference criterion* which establishes a partial order " $\preceq$ " among possible arguments; thus, for two arguments A and B in conflict, it may be the case that A is strictly preferred over B ($A \succ B$), that A and B are equally preferable ($A \succeq B$ and $A \preceq B$) or that A and B are not comparable with each other. In the above setting, since we arrive at conclusions by building defeasible arguments, and since *mathematical argumentation* is usually called *argumentation*, we sometimes call this kind of reasoning *defeasible argumentation*.

For the sake of example, let us consider the well-known example of nonmonotonic reasoning in AI about the flying abilities of birds, recast in argumentative terms. Consider the following sentences:

1. Birds usually fly.
2. Penguins usually do not fly.
3. Penguins are birds.

The first two sentences correspond to *defeasible rules* (rules which are subject to possible exceptions). The third sentence is a *strict rule*, where no exceptions are possible. Given now the fact that *Tweety is a penguin* two different arguments can be constructed:

1. Argument A (based on rules 1 & 3): Tweety is a penguin. Penguins are birds. Birds usually fly. So Tweety flies.
2. Argument B (based on rule 2): Tweety is a penguin. Penguins usually do not fly. So Tweety does not fly.

In this particular situation, two arguments arise that cannot be accepted simultaneously (as they reach contradictory conclusions). Note that argument B seems rationally preferable over argument A, as it is based on more *specific* information. As a matter of fact, specificity is commonly adopted as a syntax-based criterion among conflicting arguments, preferring those arguments which are *more informed* or *more direct*.[14,15] In this particular case, if we adopt specificity as a preference criterion, argument B is justified, whereas A is not (as it is defeated by B). The above situation can easily

[1] Let $\mathcal{L}$ be a logical language, and let S, S' be arbitrary sets well-formed formulas in $\mathcal{L}$, such that $S \subseteq S'$. An inference relationship " $\vdash$ " is monotonic whenever $S \vdash f$ implies that $S' \vdash f$, for any arbitrary well-formed formula f in $\mathcal{L}$. Classical logic is monotonic (new information cannot invalidate already existing theorems), whereas commonsense reasoning is not. For a discussion see Ref. 12.

become much more complex, as an argument may be defeated by a second argument, which in turn can be defeated by a third argument, *reinstating* the first one.

In order to illustrate a more complex situation involving argumentative reasoning, we will consider in the subsequent analysis a knowledge base $K_{english}$ which contains incomplete and potentially inconsistent information about English language usage. We will formalize the contents of $K_{english}$ in terms of the following defeasible and strict rules:

1. English words are usually acceptable in English texts.
2. Archaisms are usually not acceptable in English texts.
3. Words intended for biblical texts are not usually considered archaisms.
4. Old English words are usually archaisms.
5. Biblical texts in English are English texts.
6. Medical texts in English are English texts.
7. Old English words are English words.

Note that rules 1, 2, 3 and 4 are defeasible, whereas rules 5, 6 and 7 are strict. Let us assume that we are concerned about assessing the correctness of an English translation of a biblical text. More concretely, we are given an old English word *"thou"* which appears in a biblical text t_{bib}. Different arguments leading to conflicting conclusions could be obtained from the above knowledge base $K_{english}$, namely:

1. Argument A (based on strict rules 5,7, defeasible rule 1): *"thou"* is an old English word. The text t_{bib} is a biblical text. Biblical texts in English are English texts. English words are usually acceptable in English texts. Therefore *"thou"* is acceptable in text t_{bib}.
2. Argument B (based on strict rules 5, defeasible rules 2, 4): *"thou"* is an archaism. Archaisms are usually not acceptable in English texts. The text t_{bib} is a biblical text. Biblical texts in English are English texts. Therefore *"thou"* is not acceptable in text t_{bib}.
3. Argument C (based on strict rules 5, defeasible rules 3): *"thou"* is not an archaism, since old English words intended for biblical texts are not usually considered archaisms, and *"thou"* appears in a biblical text t_{bib}.

Assuming that we adopt specificity as preference criterion, as done before, it can be established that argument B is strictly more specific than argument A, and argument C is strictly more specific than argument B. In order to determine the epistemic status of e.g. argument A, all possible defeaters for A have to be analyzed. As defeaters are arguments, they may be on their turn be defeated by other arguments. This situation prompts for a recursive analysis, in which to determine whether our initial argument A is *ultimately acceptable*, its defeaters, the defeaters for these defeaters, and so on, should be taken into account.

The interplay among the three arguments above can be summarized as follows: there is an argument A supporting the conclusion that *"thou"* is acceptable. On the basis of the knowledge base $K_{english}$, this argument can only be defeated by a second, more specific argument B, supporting the conclusion that *"thou"* is not acceptable, as it is an archaism. At this intermediate point, argument A is defeated, and not justified. But there is an argument C which defeats argument B, stating that *"thou"* is not an archaism in the particular context of biblical texts. In this sequence of arguments, argument C reinstates indirectly argument A, defeating argument B and making argument A to be ultimately justified.

Argument-based approaches to modelling commonsense reasoning drove the development of new logical languages, resulting in new formalisms which extended classical logic for performing nonmonotonic reasoning. In this context, defeasible argumentation[16,17] evolved in the last decade as a successful computational approach to formalize commonsense reasoning. In the last few years particular attention has been given to extensions of *logic programming*, which has turned out to be a suitable language for formalizing knowledge representation and argumentative inference. In the next subsection we will introduce *defeasible logic programming* (DeLP) a defeasible argumentation formalism based on logic programming.

2.2 Defeasible Logic Programming: fundamentals

Defeasible logic programming (DeLP) 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 $p \mathrel{-\!\!\!<} q_1, \ldots, q_k$, which stands for "$q_1, \ldots q_k$ provide a *tentative reason* to believe p." In DeLP contradiction stands for deriving two complementary literals with respect to strict (p and $\sim p$) or default negation (p and $\mathsf{not}\, p$). The underlying logical language is that of extended logic programming, enriched with a special symbol " $\mathrel{-\!\!\!<}$ " to denote defeasible rules. Both default and classical negation are allowed (denoted not and $\sim$, resp.). Syntactically, the symbol " $\mathrel{-\!\!\!<}$ " is all that distinguishes a *defeasible* rule $p \mathrel{-\!\!\!<} q_1, \ldots q_k$ 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.

Example 2.1 *Consider the commonsense knowledge base $K_{english}$ for modelling defeasible criteria about usage of English language presented before. Such knowledge base could be modelled as a DeLP program $P_{english}$ = (Π, Δ) as follows:*

$$\Pi = \left\{ \begin{array}{rcl} englishText(Text) & \leftarrow & biblicalText(Text). \\ englishText(Text) & \leftarrow & medicalText(Text). \\ englishWord(Word) & \leftarrow & oldEnglishWord(Word). \\ englishWord(body) & \leftarrow & \\ oldEnglishWord(thou) & \leftarrow & \\ biblicalText(t_{bib}) & \leftarrow & \\ medicalText(t_{med}) & \leftarrow & \end{array} \right.$$

$$\Delta = \left\{ \begin{array}{rcl} acceptable(Word, Text) & \mathrel{-\!\!\prec} & englishWord(Word), \\ & & englishText(Text). \\ \sim acceptable(Word, Text) & \mathrel{-\!\!\prec} & englishWord(Word), \\ & & englishText(Text), \\ & & archaism(Word, Text). \\ archaism(Word, Text) & \mathrel{-\!\!\prec} & oldEnglishWord(Word). \\ \sim archaism(Word, Text) & \mathrel{-\!\!\prec} & oldEnglishWord(Word), \\ & & biblicalText(Text). \end{array} \right.$$

Note that the strict rules in Π correspond to the rules 5, 6 and 7 in the knowledge base $K_{english}$, whereas the defeasible rules in Δ correspond to the rules 1, 2, 3 and 4. In order to make our example richer, we have also included some additional information concerning medical texts. Facts in Π tell us that "thou" *is an old English word,* "body" *is an English word, and t_{bib} and t_{med} are biblical and medical texts in English, respectively.*

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 2.1 *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. *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'[18] on arguments: any superset $\mathcal{A}'$ of $\mathcal{A}$ can be proven to be 'weaker' than $\mathcal{A}$ itself, as the former relies on more defeasible information. The non-contradiction requirement forbids the use of (ground instances of) defeasible rules in an argument $\mathcal{A}$ whenever $\Pi \cup \ \mathcal{A}$ logically entails two complementary literals. It must be noted that given an $\langle \mathcal{A}, q \rangle$, the set A **only** accounts for the defeasible rules required for in the derivation of the conclusion q.

Example 2.2 *Consider the DeLP program $P_{english}$ from example 2.1. Then*

$$\mathcal{A}_1 = \{\ acceptable(thou, t_{bib}) \mathrel{-\!\!\prec} englishWord(thou),\ englishText(t_{bib})\ \}$$

is an argument for $acceptable(thou, t_{bib})$. *Note that* $acceptable(thou, t_{bib})$ *can be derived by backward chaining from* $\Pi \cup \mathcal{A}_1$, $\mathcal{A}_1$ *is non-contradictory (no contradictory literals* p *and* $\sim p$ *can be derived from* $\Pi \cup \mathcal{A}_1$*), and* $\mathcal{A}_1$ *is minimal (since* $acceptable(thou, t_{bib})$ *cannot be derived from* Π*). We have also that*

$$\begin{array}{lll} \mathcal{A}_2 = \{ & \sim acceptable(thou, t_{bib}) \;\mathrel{-\!\!\prec} & englishWord(thou), \\ & & englishText(t_{bib}), \\ & & archaism(thou, t_{bib}) \; ; \\ & \multicolumn{2}{l}{archaism(thou, text_1) \mathrel{-\!\!\prec} oldEnglishWord(thou) \; \}} \end{array}$$

is an argument for $\sim acceptable(thou, t_{bib})$. *Note that for the sake of clarity we use semicolons to separate elements in an argument, e.g.* $\mathcal{A} = \{e_1 \; ; \; e_2 \; ; \ldots ; \; e_k \; \}$. *In the latter case, for the argument* $\mathcal{A}_2$ *with conclusion* $\sim acceptable(thou, t_{bib})$ *a subargument* $\langle \mathcal{A}_2', archaism(thou, t_{bib}) \rangle$ *can be distinguished, with*

$$\mathcal{A}_2' = \{ \; archaism(thou, t_{bib}) \mathrel{-\!\!\prec} oldEnglishWord(thou) \; \}$$

Definition 2.2 *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 a 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* $\mathcal{A}_1$.

Example 2.3 *Consider the DeLP program from example 2.1 and the two arguments given in example 2.2 (viz. the argument* $\langle \mathcal{A}_1, acceptable(thou, t_{bib}) \rangle$ *and the argument* $\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib}) \rangle$*). In this case,* $\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib}) \rangle$ *counterargues* $\langle \mathcal{A}_1, acceptable(thou, t_{bib}) \rangle$*, since the set*

$$\Pi \cup \{\sim acceptable(thou, t_{bib}), acceptable(thou, t_{bib})\}$$

is contradictory.

Given two conflicting arguments associated with a given DeLP program $\mathcal{P}$, a preference criterion is required in order to decide which of them prevails over the other, or if both are equally acceptable. As in most argumentation frameworks, a partial order $\preceq \subseteq Args(\mathcal{P}) \times Args(\mathcal{P})$ is used in DeLP, which is induced by the *specificity* relationship among arguments, as defined in Ref. 18. A discussion on computing specificity efficiently in the context of DeLP can be found in Ref. 19. It must be remarked that other alternative partial orders could also be used as preference criterion among arguments.

Definition 2.3 *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 argument* $\langle \mathcal{A}_2, q_2 \rangle$ *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* $\langle \mathcal{A}_2, q_2 \rangle$.

Example 2.4 *Consider the DeLP program from example 2.1 and the arguments $\mathcal{A}_1$ for concluding $acceptable(thou, t_{bib})$ and $\mathcal{A}_2$ for concluding $\sim acceptable(thou, t_{bib})$ in example 2.3. In this case we have that the argument $\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib})\rangle$ is a proper defeater for $\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$, as it is based on more specific information: argument $\mathcal{A}_2$ relies on the defeasible rule*

$$\begin{array}{ll}\sim acceptable(thou, t_{bib}) \prec\!\!\!- & englishWord(thou), englishText(t_{bib}),\\ & archaism(thou, t_{bib})\end{array}$$

which is more informed than the defeasible rule

$$acceptable(thou, t_{bib}) \prec\!\!\!- englishWord(thou), englishText(t_{bib})$$

used in $\mathcal{A}_1$.

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 exhaustive 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.[2] An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite.[10]

Example 2.5 *Consider the DeLP program $\mathcal{P}$ from example 2.1. As already discussed in the previous examples, different arguments can be derived from $\mathcal{P}$. Argument $\mathcal{A}_1$ for concluding $acceptable(thou, t_{bib})$ was shown to be defeated by argument $\mathcal{A}_2$ for $\sim acceptable(thou, t_{bib})$, with*

$$\mathcal{A}_1 = \{\ acceptable(thou, t_{bib}) \prec\!\!\!- englishWord(thou), englishText(t_{bib})\ \}$$

$$\begin{array}{rll}\mathcal{A}_2 = \{ & \sim acceptable(thou, t_{bib}) \prec\!\!\!- & englishWord(thou),\\ & & englishText(t_{bib}),\\ & & archaism(thou, t_{bib})\ ;\\ & archaism(thou, t_{bib}) \prec\!\!\!- & oldEnglishWord(thou)\ \}\end{array}$$

Note that the latter argument can on its turn be defeated by a third argument $\mathcal{A}_3$ for concluding $\sim archaism(thou, t_{bib})$, with

$$\mathcal{A}_3 = \{\ \sim archaism(thou, t_{bib}) \prec\!\!\!- oldEnglishWord(thou), biblicalText(t_{bib})\ \}$$

which is a proper defeater for $\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib})\rangle$. Note that no defeater for $\langle \mathcal{A}_3, \sim archaism(thou, t_{bib})\rangle$ can be obtained from $\mathcal{P}$. The sequence of arguments

$$[\ \langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle, \langle \mathcal{A}_2, \sim acceptable(thou, t_{bib})\rangle,\\ \langle \mathcal{A}_3, \sim archaism(thou, t_{bib})\rangle\ ]$$

[2] For an in-depth treatment of dialectical constraints in DeLP the reader is referred to Ref. 10.

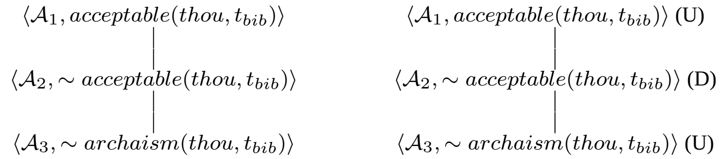
$\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$
$\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib})\rangle$
$\langle \mathcal{A}_3, \sim archaism(thou, t_{bib})\rangle$
$\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$ (U)
$\langle \mathcal{A}_2, \sim acceptable(thou, t_{bib})\rangle$ (D)
$\langle \mathcal{A}_3, \sim archaism(thou, t_{bib})\rangle$ (U)

Figure 1: Dialectical tree for $\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$ (example 2.1): before applying the marking procedure (left side) and after (right side)

constitutes an argumentation line. Note that this can be thought of as an exchange of arguments or dialogue between two parties, Pro and Con, where Pro is defending the hypothesis that thou *is acceptable, and Con is supporting the opposite stance. Pro advances argument A_1, which is defeated by Con with argument A_2. Then Pro rebuts A_2 by advancing a third argument A_3 which defeats A_2. No more arguments can be advanced in the dialogue. Note that Pro "wins" the dialogue, as defeating A_2 accounts for reinstating Pro's first argument A_1.*

Given a DeLP program $\mathcal{P}$ and an initial argument $\langle \mathcal{A}_0, Q_0\rangle$, the set of all acceptable 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 2.4 *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 $i = 0 \ldots n-1$.*

Example 2.6 *Consider again the DeLP program $\mathcal{P}$ from example 2.1 and the argumentation line computed in example 2.5. Note that in the case of the argument $\mathcal{A}_1$ for $acceptable(thou, t_{bib})$ this is the only possible argumentation line. Hence the dialectical tree rooted in the argument $\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$ has a unique branch, as shown in Figure 1(left).*

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*.

Example 2.7 *Consider the dialectical tree for $\langle \mathcal{A}_1, acceptable(thou, t_{bib})\rangle$ shown in example 2.6. Figure 1(right) shows the resulting dialectical tree after applying the marking procedure described before.*

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. Answer YES: Believe q when there is a warranted argument for q that follows from $\mathcal{P}$;
2. Answer NO: Believe $\sim q$ when there is a warranted argument for $\sim q$ that follows from $\mathcal{P}$;
3. Answer UNDECIDED: 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: 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 $\mathcal{P}$.[10]

Example 2.8 *Consider the DeLP program from example 2.1. A sequence of possible queries associated with $\mathcal{P}$ and the associated output according to DeLP semantics is shown below:*

- *Given the query $acceptable(thou, t_{bib})$, the associated answer is* YES, *as there is a warranted argument $\langle \mathcal{A}_1, acceptable(thou, t_{bib}) \rangle$ supporting the conclusion $acceptable(thou, t_{bib})$ (as shown in the previous examples).*

- *Given the query $acceptable(thou, t_{med})$, the associate answer is* NO. *There is an argument $\langle \mathcal{B}_1, acceptable(thou, t_{med}) \rangle$, with*

  $$\mathcal{B}_1 = \{ \; acceptable(thou, t_{med}) \mathrel{-\!\!\prec} englishWord(thou), englishText(t_{med}) \; \}$$

  which is defeated by argument $\langle \mathcal{B}_2, \sim acceptable(thou, t_{med}) \rangle$, with

  $$\mathcal{B}_2 = \{ \; \sim acceptable(thou, t_{med}) \mathrel{-\!\!\prec} englishWord(thou), englishText(t_{med}), archaism(thou) \; \}$$

  There are no more arguments to consider (note that in this case, the second argument has no defeaters). After computing and marking the associated dialectical tree as described in Definition 2.4, the root of the tree turns out to be labelled as D-node. On the contrary, when analyzing the complementary literal $\sim acceptable(thou, t_{med})$, we get one single argument (namely, $\langle \mathcal{B}_2, \sim acceptable(thou, t_{med}) \rangle$) Consequently, the associated dialectical tree rooted in $\langle \mathcal{B}_2, \sim acceptable(thou, t_{med}) \rangle$ (i.e. $\mathcal{T}_{\langle \mathcal{B}_2, \sim acceptable(thou, t_{med}) \rangle}$) has a single node, marked as U-node. Therefore $\langle \mathcal{B}_2, \sim acceptable(thou, t_{med}) \rangle$ is warranted.

- *Given the query $acceptable(body, t_{med})$, the resulting answer would be* YES. *There is an argument $\langle \mathcal{C}_1, acceptable(body, t_{med}) \rangle$, with*

  $$\mathcal{C}_1 = \{ \; acceptable(body, t_{med}) \mathrel{-\!\!\prec} englishWord(body), englishText(t_{med}) \; \}$$

for which no defeaters can be found. Therefore the corresponding dialectical tree $\mathcal{T}_{\langle \mathcal{C}_1, acceptable(body, t_{med})\rangle}$ *has a single node, marked as* U*-node. Therefore the argument* $\langle \mathcal{C}_1, acceptable(body, t_{med})\rangle$ *is warranted.*

2.3 Using DeLP for language usage assessment

In the last years defeasible logic programming has been successfully used in a variety of real-world applications based on argumentation, such as Web recommendation systems[20,21], clustering classification[22], and multiagent systems[23], among others. As we have seen in the previous analysis, situations involving language usage can be also modelled in terms of DeLP programs. In the particular case of archaisms, we could assume that they could be encoded as a list of facts in a DeLP program modelling language usage, providing thus a "dictionary" of archaisms in English language. Thus, on the basis of such background knowledge, DeLP rules will allow us to infer the acceptability of different terms by posing suitable queries.

However, defining concepts associated with language usage is not always such an easy task. In the case of archaisms, the corpus of words associated with this concept tends to suffer only minor changes as times goes by. Other concepts (such as the notion of "buzz words or vogue expressions") are constantly undergoing change and modifications, and cannot be so easily captured in DeLP. Let us consider again the case of the journalist described in the introduction: making a list of terms which are *"...geographical variations corresponding to one country but are widely used in other countries because of their popularity in the media"* does not seem so easy to model.

Our proposal aims at enriching DeLP capabilities for modelling commonsense reasoning about language usage by incorporating specialized built-in predicates called *usage indices*. Usage indices will provide a way of characterizing *defeasible reasons* for assessing language usage, allowing us to conclude whether a given term T is suitable (or not) in a particular context on the basis of the current Web corpus. As we will see, such indices can be easily computed on the basis of information extracted from the Web by means of advanced search facilities provided by most search engines (e.g. GOOGLE). The resulting defeasible knowledge will be formalized in terms of Defeasible Logic Programming (DeLP), a logic programming formalism for defeasible argumentation. On the basis of this formalization we define ARGUETERM, an argument-based computational framework which aims to providing proactive assistance for language usage assessment.

3 Usage indices: detecting patterns on the Web

An extensive amount of sample sentences in different natural languages have been accumulated as Web documents on the World Wide Web. A significant portion of such documents are accessible through search engines, whose pattern-matching capabilities have turned out to be useful to exploit the Web-Corpora. The Web as a Corpus offers a number of advantages in comparison with traditional linguistic corpora, namely:

- **Updated and free information.** Building up large linguistic corpora requires considerable effort and keeping them up to date might prove to be a difficult, if not impossible task. The Web corpus is huge and exists as it is, namely, as a free tool[3].

- **State-of-the-art linguistic database.** The Web corpus reflects the current status of language, as Web documents are created, updated and eventually deleted. Different language registers and levels of formality (colloquial, formal, standard, etc.) and text-types (technical, scientific, medical, legal, journalistic, etc.) can be found on the Web corpus[4].
- User-friendly handling of documents. Several Web-based applications have been developed for effective pattern-matching, clustering and text classification. Such applications provide a natural tool for dealing with Web based corpora.

In order to analyze relevant features of language usage patterns in Web-based corpora, values associated with absolute or relative frequencies of string patterns with respect to different Web domains turn out to be particularly useful. We call such values *usage indices*. Such usage indices can be easily computed by means 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. Sans serif font will be used for natural language expressions to be analyzed, e.g. this is example. Henceforth the term *domain* will be used indistinctly to refer to complete Web domain names (e.g., `'google.com'`) as well as to the suffix portions 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 $\|\mathcal{D}\| = \sum_{i=1}^{k} \|d_i\|$. [5] Similarly, given a domain d and a string s, we will use $\|d\|_s$ to denote the number of *hit counts* for s with respect to d, i.e. the number of Web pages in domain d containing the string s. [6] Usage indices will be based on computing occurrences in sets of domains, as presented below.

Definition 3.1 *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,

[3] Paper dictionaries and reference books such as the Collins Cobuild English Usage Dictionary, 1992, while appropriate and updated, are doomed to becoming obsolete and limited in scope.

[4] Even though concordancers continue to be widely used in linguistic study and the study of usage for reference (See John de Szendeffy (2004): "Vocabulary and Usage Activities with concordances, in `http://www.bu.edu/celop/mll/tutorials/pdf_public/concordance.pdf`), GOOGLE searches provide a wider range of usage data for vocabulary, grammar and even punctuation.

[5] 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.

[6] 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 means of the queries 's site:d' and 'site:d', respectively.

constrained usage U_c, ratio usage U_r, prefix usage U_p, *and* relative usage U_{rel} *as follows:*[7]

- $U_g(s) =_{def} \|Web\|_s$.
- $U_c(s, \mathcal{D}) =_{def} \|\mathcal{D}\|_s = \sum_{i=1}^{k} \|d_i\|_s$.
- $U_r(s, \mathcal{D}_1, \mathcal{D}_2) =_{def} \frac{(U_c(s,\mathcal{D}_1)+1)}{(U_c(s,\mathcal{D}_2)+1)} \times \frac{\|\mathcal{D}_2\|}{\|\mathcal{D}_1\|}$.
- $U_p(s_1, s, \mathcal{D}) =_{def} U_c(s_1 \bullet s, \mathcal{D})/U_c(s, \mathcal{D})$ *if* $U_c(s, \mathcal{D}) \neq 0$, *and* 0 *otherwise.*
- $U_{rel}(s_1, s_2, \mathcal{D}) =_{def} \frac{(U_c(s_1,\mathcal{D})+1)}{(U_c(s_2,\mathcal{D})+1)}$

Given a string s, the constrained usage $U_c(s, \mathcal{D})$ represents the number 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 $\mathcal{D}_2$. 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 $\mathcal{D}$. Finally, the relative usage U_{rel} allows to contrast the usage of two different strings s_1 and s_2 with respect to a given domain set $\mathcal{D}$.

Example 3.1 *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*[8]

$$\begin{aligned}
\|Web\| &= 3307998701 \\
\|\{d_1\}\| &= 28000000 \\
U_c(s_1, \{d_1\}) &= 435 \\
U_c(s_1, Web) &= 13700 \\
U_r(s_1, \{d_1\}, Web) &= (436/13701) * (3307998701/28000000) = 3.76 \\
U_p(s_2, s_3, \{d_2\}) &= 677/747 = 0.906.
\end{aligned}$$

Note in the above example that statistical-based inferences can be performed from usage indices (e.g. 90% of occurrences of the phrase of twins associated with the URL `'.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).

Several language usage phenomena can be analyzed in the light of the usage indices that we have defined. We distinguish three major groups for study: a) the analysis of calque or mimetism[24] in non-native English speakers; b) the study of dialectal usage of language; and c) the scope of common usage-related phenomena at written level. These major groups will be discussed in detail in the next subsections.

[7] In some of the definitions that follow, we adopt the usual strategy of computing some ratios of the form $value_1/value_2$ as $(value_1 + 1)/(value_2 + 1)$ in order to avoid the case of division by zero.

[8] Computations of usage indices in this article were performed using GOOGLE with the existing Web corpus on Feb. 19, 2004. Due to space limitations a detailed computation of some usage indices is not included.

3.1 Studying Calque in texts of non-native English speakers

People using English as a foreign language (EFL) tend to make mistakes whenever they convert into English equivalents those syntactic structures which are valid in their mother tongue.[9] This phenomenon is known in translation theory as *calque d'expression*.[24] Some irregular patterns are more frequent among native speakers of Romance languages (e.g. Spanish), whereas others are more frequent among native speakers of Germanic ones (e.g. Dutch). Usage indices can be used to analyze and provide assessment on such situations, by filtering occurrences according to the Web domains associated with particular countries.

Example 3.2 *Consider the case of the English verb to associate, that normally takes the preposition with (e.g. in association with.., this is associated with..). The Spanish verb asociar has an equivalent meaning, and two prepositions are possible: asociar a (quite frequent) or asociar con (not so frequent).*

Remarkably there exists a common calque d'expression *of the Spanish preposition a into the English preposition to, derived from cases such as to go to school = ir a la escuela and This is going to be sent to my parents = esto va a ser enviado a mis padres. Thus, a common tendency by non native English speakers whose mother tongue is Spanish is to apply to instead of other prepositions which would be used by native speakers instead.*

The extent of this phenomenon in a Romance language in comparison with Germanic languages in general and English language in particular can be analyzed by computing hit counts for be associated to and be associated with in domains '.es', '.de', '.uk'. The usage index U_{rel} helps provide a measure of the above phenomenon. Let s_1=be associated to, and s_2=be associated with. Computing $U_{rel}(s_1, s_2, Web)$ we have:

$$U_{rel}(s_1, s_2, Web) \quad = \quad \frac{(U_c(s_1, Web) + 1)}{(U_c(s_2, Web) + 1)} = \frac{43900}{1880000} = 0.02$$

Similarly, we get the following results for other domains:

$$U_{rel}(s_1, s_2, \texttt{'.es'}) = \tfrac{600}{3300} = 0.18$$

$$U_{rel}(s_1, s_2, \texttt{'.de'}) = \tfrac{1780}{23700} = 0.08$$

$$U_{rel}(s_1, s_2, \texttt{'.uk'}) = \tfrac{938}{125000} = 0.008$$

$$U_{rel}(s_1, s_2, \texttt{'.au'}) = \tfrac{349}{53500} = 0.006$$

From the above figures it follows that in Australian and British Web pages the proportion of use of be associated to with respect to be associated with is less than 1%, whereas in the the whole Web space is only 2%. Notably there is considerably stronger incidence of this phenomenon in non-native English speakers (particularly Spanish speakers (18%) in contrast with in German ones (8%)). This can be explained in terms of the *calque d'expression* discussed above.

[9] For a study on English lexical structures converted into their Spanish counterparts, also known as *false friends*, see Ref. 25.

String	'.es'	'.ar'
a que esperas	15400	632
que esperas	17400	22000

Figure 2: Hit counts for [a| ∅] qué esperas

3.2 Assessing the dialectal usage of language

Several dialectal aspects deriving from language usage may also be evaluated using the proposed approach. An example could be the assessment of language usage across linguistic communities which share a common natural language. A particularly interesting case are the geographical variations of widespread languages such as English and Spanish. There are many well-known examples of lexical differences between American and British English (e.g. elevator and lift) or Mexican Spanish and peninsular Spanish (e.g. carro and coche). Clearly, there is a series of dialectal expressions (geographical variations) which will be fairly standard in some particular areas[26], but may well not necessarily be understood by speakers from other geographical areas.

A more subtle phenomenon occurs with the use of a particular syntactic structure which is not a regionalism, but for some reason is more common and/or frequent in a particular country than in others. Such phenomenon can also be surveyed by applying the usage indices presented before, as shown in the following examples.

Example 3.3 *In Spain, the sentence ¿A qué esperas para comprarlo? (=What are you waiting for to buy it?) is commonly used in everyday language, whereas an Argentinian speaker would rather say ¿Qué esperas para comprarlo?, removing the preposition A from the sentence. In fact, the use of such preposition in front of the above question will sound rather strange for Spanish speakers in Argentina.*

The extent of this language phenomenon is very difficult to assess, as it involves everyday language usage, strongly influenced by the mass media. Such utterances are difficult to find in standard language corpora (in, say, the British National Corpus). We can get a better understanding of this phenomenon by computing the hit counts associated with a qué esperas and qué esperas in the Web domains corresponding to Argentina and Spain (i.e. '.ar' and '.es'), as shown in Figure 2. From this the following usage indices can be computed:

$$U_p(\text{a, que esperas, '.ar'}) = \frac{632}{22000} = 0.03$$

$$U_p(\text{a, que esperas, '.es'}) = \frac{15400}{17400} = 0.89$$

This shows that only 3% of Web documents found on Argentinian Web sites has the preposition a before the string qué esperas, in contrast with 89% of occurrences in Spanish Web sites.

Example 3.4 *The sentence in Spanish merece la pena intentarlo (=it is worth doing it) is commonly used in Spain, whereas in other Spanish-speaking countries such phrasing tends to be replaced by vale la pena intentarlo. It must be remarked that vale and merece are, in this context, semantically equivalent. Thus, both sentences are grammatically*

String	'.es'	'.ar'
vale la pena intentarlo	113	337
merece la pena intentarlo	176	11
la pena intentarlo	402	399

Figure 3: Hit counts for [vale|merece| ∅] la pena intentarlo in Web domains '.es' and '.ar'

correct in Spanish, and understood by Spanish speakers both in Spain and Argentina. However, a Spanish speaker seems to be more prone to use the first version than an Argentinian one. This situation can be handled in terms of computing hit counts for [merece|vale| ∅]la pena intentarlo], using the domains associated with Spain and Argentina, as shown in Figure 3. We can compute the following prefix usage indices:

$$U_p\textit{(merece, la pena intentarlo,'.ar')} = \frac{11}{399} = 0.03$$

$$U_p\textit{(merece, la pena intentarlo,'.es')} = \frac{176}{402} = 0.44$$

This shows that only 3% of Web documents found on Argentinian Web sites have merece as a prefix string for la pena intentarlo, in contrast with 44% of Spanish Web sites.

3.3 Analyzing the scope of common usage-related errors

Usage indices help also to distinguish to what extent some words and expressions are falling into disuse and are being replaced by new terms. This phenomenon may be best illustrated with examples of recurrent and fairly fossilized patterns of language such as *set phrases* (e.g. safe and sound), spick and span (as characterized in Ref. 27), fixed expressions (in Ref. 27), fixed expressions (having said that, best, as a matter of fact (as characterized in Ref. 26) and proverbs.

Some of these expressions (e.g. as a matter of fact vs. *as a matter of facts) are sometimes misleading for speakers, as the establishment of their meaning cannot be accounted for through prescriptive, hard-and-fast grammatical rules but rather through their usage by both native and non-native speakers of the language. Likewise, two expressions may be grammatically equivalent, but only one of them is the correct idiom to be used (e.g. he works like a dog vs. *he works as if he were a dog). Speakers may thus construct incorrect sentences by trying to re-phrase some existing idiom.

Dictionaries both on paper and on-line are not always reliable tools when it comes to reflecting usage-related features and certain nuances of meaning deriving from the continuous evolution of languages. As languages evolve over time, new words and expressions are coined while others become obsolete, disappear or change their usage patterns. Compare for example the usage of the English conjunction whilst, now used in fairly formal, poetic and specialized contexts, and commonly found in British English in front of the word while, used in standard day-to-day English. Although “whilst is a perfectly valid synonym of “while, in American usage it would be considered old-fashioned and pretentious. Their frequency of use and their appropriateness may be difficult to assess using a monolingual English dictionary[10]. Usage indices allow to obtain a dynamic, up-to-date measure for such situations, as shown in the following examples.

Example 3.5 *Consider s_1=while and s_2=whilst. Computing $U_{rel}(s_2, s_1, Web)$ returns to what extent whilst is used in comparison to while in the whole Web space. We get:*

$$U_{rel}(s_2, s_1, Web) = \frac{6080000}{127000000} \simeq 0.05$$

Similar results are obtained by computing $U_{rel}(s_2, s_1, \texttt{'.uk'})$ (i.e, restricting the analysis to Web pages from United Kingom). Notably, whilst seems to has a higher use in Australia, as we have:

$$U_{rel}(s_2, s_1, \texttt{'.au'}) = \frac{420000}{1710000} \simeq 0.25$$

The correct use of prepositions in English is also a common source of trouble for non-native speakers of English. Complex verb structures and formulaic language (I am looking forward to hearing from you) constitute ambiguous situations for many speakers. In the example above, the use of to as a preposition causes a gerund form to be used immediately afterwards (hearing). However, the situation is ambiguous as to is also used for infinitive verb forms, so *I look forward to hear from you seems also reasonable. In fact, Looking forward to hear from you is ungrammatical, since the construction is Looking forward to [Noun Phrase] and hear from you is not a noun phrase (while hearing from you is).[11] Usage indices also provide a useful tool for studying such situations, as shown in the following example.

Example 3.6 *Consider the strings s_1 = looking forward to hearing and s_2 = *looking forward to hear Computing the relative usage U_{rel} for these two expressions with respect to different Web domains allows to gain a better perspective of the influence of this mistake. For UK, Australia and Germany we obtain*

$$\begin{aligned} U_{rel}(s_2, s_1, \texttt{'.uk'}) &= \tfrac{372}{7590} = 0.049 \\ U_{rel}(s_2, s_1, \texttt{'.au'}) &= \tfrac{142}{2660} = 0.05 \\ U_{rel}(s_2, s_1, \texttt{'.de'}) &= \tfrac{442}{1980} = 0.22 \end{aligned}$$

These figures show that the phenomenon has a considerably higher impact in German Web pages (22%) in contrast to Webpages in English-speaking countries such as UK or Australia (5%).

4 ARGUETERM: Assessing Language Usage combining Usage Indices and Defeasible Argumentation

Although the Web corpus provides useful resources for language usage assessment on the basis of the relative and absolute frequencies in Web documents, coming up with

[10] E.g. Collins Cobuild English Language Dictionary.

[11] See remark pointed out by Stuart Robinson (Linguistics Dept. of the Australian National University) at http://www.linguistlist.org/ ask-ling/archive-1997.5/msg00278.html.

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. The user's analysis will be guided by a number of (mostly implicit) preference criteria to build and evaluate alternative *hypotheses* for coming up with a particular suggestion. As an example, finding a considerable number of hits when searching with GOOGLE for a particular string s in English can be used as a reason to believe that s is suitable for use in any English text. Such an assumption is *defeasible*, in the sense that it can be revoked in the light of additional information (e.g. if most hits for s correspond to Australian websites).

Usage indices provide us with a handy tool for formalizing situations like the one mentioned above in more precise terms. Let us consider again the case of the journalist presented in the introduction, who believes that a given expression E is not suitable for a news report intended for a Spanish newspaper, as he suspects that E is a dialectal variation e.g. from Argentina. This last assumption can be supported on the basis of a ratio $R = U_r(E, \{\texttt{'.ar'}\}, \{\texttt{'.es'}\})$, contrasting the number of hits for E found in Argentinean Web sites with respect to those found in Spanish ones. The fact that $R > 1$ provides a *tentative* reason for concluding that E is a dialectal variation associated with Argentina. However, knowing that E is already in use in other Spanish newspapers may make the journalist change his mind, as he would have a reason that *defeats* the previous hypothesis. Once again, the above situation can be captured by computing $R' = U_c(E, \mathcal{D}_{news})$, where $\mathcal{D}_{news}$ corresponds to the set of Web domains corresponding to representative Spanish mass media. The fact that $R' > \theta$, where θ is a particular threshold value (adopted by the user as reasonable on the basis of his/her experience) provides a new tentative reason to think that E is a common expression in the Spanish mass media, and therefore it can be used.

The preceding analysis shows that usage indices can be used as a numerical basis to come up with hypotheses about language usage patterns. Clearly, such hypotheses may be in conflict, so that the user has to perform some kind of introspective analysis, weighing such hypotheses and determining which ones are to be ultimately accepted. Defeasible argumentation frameworks such as DeLP provide a sound mathematical formalization of such rational procedure in dialectical terms. Arguments correspond to hypotheses, and the defeat relationship among arguments is analogous to weighing conflicting hypotheses. As we have discussed in Section 2.2, in the case of DeLP there exists a computational procedure to determine whether a given argument is to be ultimately accepted or warranted by means of the corresponding dialectical tree.

Following these ideas, our proposal aims at integrating usage indices and defeasible argumentation in a single computational framework called ARGUETERM.[12] Usage indices will provide a way of characterizing a number of *defeasible rules* for assessing language usage, allowing us to conclude whether a given term T is suitable (or not) in a particular context. This defeasible knowledge will be encoded as DeLP rules, and will be part of a larger DeLP program, which will be able to provide recommendations by solving distinguished queries. An outline of the architecture of the ARGUETERM approach is shown in Figure 4. Given a text T corresponding to a user document, a

[12] The main ideas underlying the approach described in this section were first suggested in a conference paper (see Ref. 28).

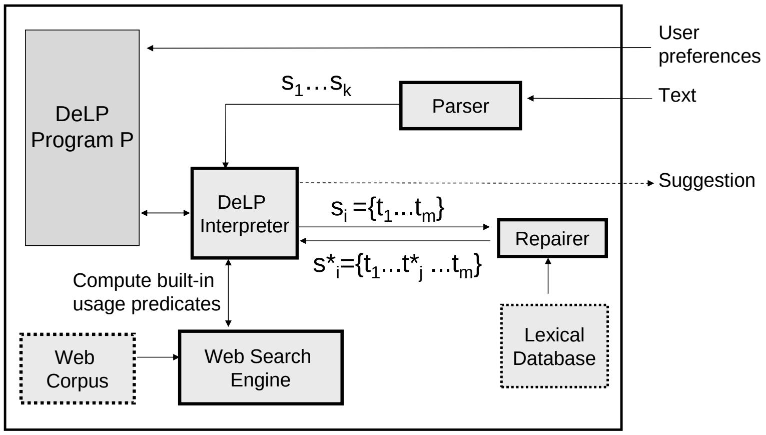

Figure 4: The ARGUETERM Framework: outline of the different components involved

front-end parser extracts a list T' = [s_1, s_2, ..., s_k] of relevant syntactic elements from T.[13] Every s_i in the list T' is analyzed with respect to 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 Definition 3.1. 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 unsuitable. This sort of funcionality can be embedded in ARGUETERM 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[29]). 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 shown in Figure 5.

4.1 Language usage assessment with ARGUETERM: 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

[13] We will assume that the input text can be parsed into strings, singling out those strings associated to nouns or noun phrases. See discussion in Section 6.

```
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 5: High-level algorithm for providing language usage assessment in ArgueTerm

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"*[14] *was a very complex phenomenon [...] For the syndicate of autonomous workers any *slight* 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 related to wrong language usage will be detected: corralito is a common term in Argentina, but not so common in Spain (except in the news). First, the noun phrase colectivo de trabajadores autónomos (syndicate of autonomous workers) has a clear meaning in Spain, but is not understood in Argentina (as gremio or agrupación is the Argentinean equivalent for the Spanish word colectivo in this context). Secondly, the noun phrase liviano

[14] The term "corralito" (little baby crib, playpen, in Peninsular Spanish, according to the Dictionary of the Spanish Academy of Language (DRAE), 21st edition) 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".

Control rules for language usage assessment:

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

Defeasible rules capturing language usage preferences:

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

Predicates based on computing Usage Indices:

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

Additional predicates:

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

Figure 6: A DeLP program modeling preference criteria for acceptable language usage patterns in newspaper articles

error (minor, slight mistake[15]) should be considered as a phenomenon known as collocation, that is, "semantically arbitrary restrictions which do not follow logically from the propositional meaning of a word".[26] In this case, liviano and error do not tend to co-occur regularly in peninsular Spanish. The adjective liviano would normally collocate with maleta (suitcase), paquete (parcel), comida (food), masa (mass), obra (work) and película (film) and its meaning is associated with *"inconstant, incontinent, of little importance"*[16]. The noun error would normally collocate with leve, ligero rather than liviano, even though the adjectives ligero and liviano are synonymous. Some of the possible criteria the editor could apply to avoid such anomalies could be summarized as

[15] Consulted in Benson, Benson and Ilson (1986): "The BBI Combinatory Dictionary of English: A Guide to Word Combinations". John Benjamins Publishing Company, Amsterdam/Philadelphia.

[16] Consulted in Diccionario de la Real Academia de la Lengua Española (DRAE), 21st. edition.

follows:

- C_1: An expression S written in Spanish is usually acceptable when it is commonly used (i.e., appears in a considerable number of already existing documents).
- C_2: An expression S is usually not acceptable if it is not commonly used.
- C_3: Regionalisms from Argentina are usually not acceptable.
- C_4: Regionalisms from Spain are usually not acceptable.
- C_5: An expression S in Spanish is usually a regionalism if it is frequently used in a Spanish-speaking country but not in others.
- C_6: Expressions which are frequently used in Argentina but have gained widespread use in the Spanish media are usually not considered as regionalisms.

The ARGUETERM framework would allow our editor to automate the above criteria for language usage assessment in terms of a DeLP program, as shown in Figure 6. Rules 1 to 6 provide a Prolog implementation of the algorithm detailed in Figure 5. Note that in this context Prolog rules are a particular instance of DeLP strict rules. Rules 1 to 4 characterize the behavior of the $solve$ predicate as outlined in Section 4. Given a text string s, the query $solve(s)$ always succeed, either because s is can be warranted as acceptable (rule 1), or because it can be replaced by a repair which is warranted as acceptable (rule 2), or because it can be warranted as not acceptable (rule 3), or because no decision is possible (rule 4). Note that rule 4 follows from the possibility of having UNDECIDED as a possible answer to a DeLP query. 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).[17] For the sake of simplicity, in this example the definition of synonym is restricted to the Spanish adjective liviano (slight). Defeasible rules 7 to 12 capture language usage preferences according to the editor's criteria C_1 to C_6 given above, defined on the basis of predicates which rely on usage indices (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 geographical variations from Argentina and Spain are usually not acceptable. Rule 11 specifies when a given expression can be defeasibly assumed to be a geographical variation 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 geographical variation. 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.[18]

Suppose we apply now the high-level algorithm presented in Figure 5, where the strings extracted from the above text are s_1, s_2 and s_3, with s_1=corralito, s_2=colectivo

[17] A lexical database such as WordNet[29] can provide a list of synonyms (synset) for an arbitrary adjective.

[18] 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 geographical variations in Argentina based on sample Spanish news domains.

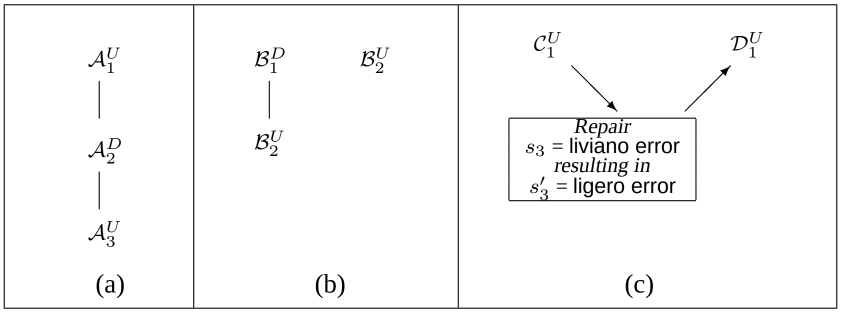

Figure 7: Dialectical trees associated with (a) $\langle \mathcal{A}_1, acc(s_1)\rangle$ (b) $\langle \mathcal{B}_1, acc(s_2)\rangle$ and $\langle \mathcal{B}_2, \sim acc(s_2)\rangle$; (c) $\langle \mathcal{C}_1, \sim acc(s_3)\rangle$ and $\langle \mathcal{D}_1, acc(s'_3)\rangle$

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 the argument $\langle \mathcal{A}_1, acc(s_1)\rangle$, with

$$\mathcal{A}_1\text{=}\{\ acc(s_1) \mathrel{-\!\!\prec} common\_in\_spanish(s_1)\ \}.$$

This argument holds since U_c(s_1,['.es','.ar'])> 200. The DeLP inference engine will then search for defeaters associated with the argument $\langle \mathcal{A}_1, acc(s_1)\rangle$. 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 geographical variation from Argentina. The argument $\langle \mathcal{A}_1, acc(s_1)\rangle$ is as follows:

$$\mathcal{A}_2\text{=}\{ \sim acc(s_1) \mathrel{-\!\!\prec} common\_in\_spanish(s_1), regionalism(s_1, \texttt{'.ar'})\ ;\\ regionalism(s_1, \texttt{'.ar'}) \mathrel{-\!\!\prec} locally\_freq(s_1, \texttt{'.ar'})\}.$$

Note that the argument $\langle \mathcal{A}_2, \sim\ acc(s_1)\rangle$ 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 $U_r(s_1$,[$\texttt{'.ar'}$],[$\texttt{'.es'}$])$= 33.1 > 10$. However, a defeater for this argument $\langle \mathcal{A}_2, \sim\ acc(s_1)\rangle$ can be found on its turn: corralito cannot be deemed as a geographical variation in Argentina, since it is fairly frequent in the Spanish news. Here we have the argument

$$\mathcal{A}_3\text{=}\{ \sim regionalism(s_1, \text{['.ar']}) \mathrel{-\!\!\prec} locally\_freq(s_1, \texttt{'.ar'}),\\ appears\_in\_news(s_1, \text{'es'})\}.$$

Note that predicate $appears\_in\_news(s_1, \text{spain})$ holds, as U_c(corralito, $\mathcal{D}$)=40, with $\mathcal{D}$ representing domains from Spanish newspapers. Note also that the definition of dialectical tree (Definition 2.4) does not allow the reuse 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 Figure 7a. 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

$$\mathcal{B}_1\text{=}\{acc(s_2) \mathrel{-\!\!\prec} common\_in\_spanish(s_2)\ \}$$

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

$$\mathcal{B}_2\text{=}\{ \sim acc(s_2) \prec\!\!\!- common\_in\_spanish(s_2), regionalism(s_2, [\text{'.es'}]);\\ regionalism(s_2, \text{'.es'}) \prec\!\!\!- locally\_freq(s_2, \texttt{'.es'})\}.$$

As above, the predicate $locally\_freq(s_2, \texttt{'.es'})$ holds here, 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 Figure 7b. 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 the conclusion $acc(s_3)$, as the literal $common\_in\_spanish(s_3)$ 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

$$\mathcal{C}_1\text{=}\{ \sim acc(s_3) \prec\!\!\!- rare\_in\_spanish(s_3)\}.$$

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

$$\mathcal{D}_1\text{=}\{ acc(s_3') \prec\!\!\!- common\_in\_spanish(s_3')\}$$

As a side effect, the message *"Accepted if rephrased as ligero error"* will be given to the user. This situation is shown in Figure 7c.

5 Related work and implementation issues

Providing assessment in word-processing activities has long been a source of research in the natural language processing community.[30] 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 user-entered text, helping to discover and point out errors that might otherwise remain unnoticed. Most popular word-processing critiquing systems include spelling-, grammar-, and style-checkers.[31] 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.[6,3] Several concordancers and writing assistant tools were developed (e.g. WebLEAP[3], WebCorp[6], KWICFinder[7] and Bonito).[19] 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 performed by the end user. In our proposal, such analysis is automated on the basis of usage indices (computed

[19] See http://nlp.fi.muni.cz/projekty/bonito/.

from the current Web corpus) and a defeasible argumentation framework. Preference criteria for language usage can be specified 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. A particular abstract machine called JAM (Justification Abstract Machine) has been specially developed for an efficient implementation of DeLP.[10] The JAM provides an argument-based extension of the traditional WAM (Warren's Abstract Machine) for Prolog. On the basis of this abstract machine an on-line interpreter of DeLP has been developed, and is freely available on the web.[20] A Java-based Integrated Development Environment (IDE) for Defeasible Logic Programming has also been developed.[32] This Java version of DeLP allows to compile DeLP code into JAM opcodes. A visual environment for interacting with DeLP programs is provided. Among other things, this environment allows to visualize the dialectical tree associated with the query being solved.[21] Several features leading to efficient implementations of DeLP have been also recently studied, in particular those related to comparing conflicting arguments by specificity[19] and extending DeLP to incorporate possibilistic reasoning.[33] Equivalence results with other extensions of logic programming have also been established.[34]

6 Conclusions and Future Work

In this article we have introduced a novel, argumentative approach to using Web search technologies as a tool for studying different language usage phenomena. Two hypotheses have guided our research. On the one hand, the fact that the large amount of existing Web pages provides a reliable source of information for detecting language usage patterns present in written text. On the other hand, those Web pages associated with a particular country domain (e.g. '.uk') are assumed to be mostly written by (or intended for) people whose natural language is the one spoken in that country (e.g. British English)[22]. Such features can be exploited by means of usage indices, which capture values resulting from performing advanced searches. It must be remarked that most search engines (such as GOOGLE) incorporate advanced tools for filtering documents according to features such as the ones used to compute usage indices, as defined in this article.

We have presented a bottom-up evaluation of the proposed framework, focusing on the ability of ARGUETERM to assess language usage in controlled scenarios. The results returned by the system are encouraging for different analyzed cases. In particular, several language phenomena corresponding to the examples discussed in Section 3 were successfully formalized and solved using ARGUETERM. All these situations, as well as the case study presented in Section 4 were encoded and solved on the basis of a prototype version of ARGUETERM implemented using the existing Java-based DeLP environment.[32] However, it must be remarked that these initial experiments only serve

[20] See `http://lidia.cs.uns.edu.ar/DeLP`.

[21] See `http://cs.uns.edu.ar/∼ags/DLP/` for details.

[22] Note that more generic domains (such as '.com' or '.org') fall outside the scope of the present analysis.

as a “proof of concept” prototype, as thorough evaluations are still being carried out. As part of our future work we plan to design different experiments to directly test the ability of the system to suggest repairs during the word processing task.

We believe that usage indices may provide valuable information about language usage and help to identify and survey potential language-related problems. This leads us to believe that usage indices may be of assistance in some language-related fields of study such as comparative linguistics, discourse analysis, translation studies and certain areas of applied linguistics such as ESL (English as a second language) and EFL (English as a foreign language). The computational approach presented in this article can also be applied to the development of advanced online style checkers which could be integrated into a conventional word processor. Thus, for example, if the user types in “this is associated to”, the style checker would deem “associated to” to be a non-acceptable expression and warn the user about a possible syntax error. A non-trivial challenge for such a system is how to determine which subpatterns are to be considered and which possible alternative suggestions can be automatically given. Part of our current work involves studying the possibility of developing such style checkers. Research in this direction is currently underway.

Acknowledgments

This research was partially supported by CYCIT Projects TIC2001-1577-C03-01/03, TIC2003-00950, TIN 2004-07933-C03-03, by Ramón y Cajal Program (MCyT, Spain) and by CONICET (Argentina).

References

1. Kilgarriff A. Web as Corpus. In Proc. Corpus Linguistic Conf., pages 342–344. UCREL-Lancaster Univ, UK, 2001.
2. Fletcher W. Facilitating the compilation and dissemination of ad-hoc web corpora. In Proc. 5th. Intl. Conf. of Teaching and Language Corpora (TALC 2002), 2002.
3. Yamanoue T, Minami T, Ruxton I, Sakurai W. Learning Usage of English KWICly with WebLEAP/DSR. In Proc. 2nd. Intl. Conf. on Inf. Technology and Applications (ICITA-2004), 2004.
4. Volk M. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In Proc. of Corpus Linguistics, Lancaster, UK, Mar. 2001.
5. Scharl A, Bauer C. Mining large samples of web-based corpora. Knowledge-Based Systems, 17(5-6):229–233, 2004.
6. Renouf A. Webcorp: providing a renewable data source for corpus linguists. In et.al G, editor, Extending the scope of corpus-based research: new applications, new challenges., pages 219–318. Rodolpi, 2002.
7. Fletcher W. Concordancing the Web with KWiCFinder. In Proc. 3rd North American Symposium on Corpus Linguistics and Language Teaching. American Assoc. for Applied Corpus Linguistics, 2001.

8. Pollock J. Knowledge and Justification. Princeton, 1974.
9. Pollock J. Defeasible Reasoning. Cognitive Science, 11:481–518, 1987.
10. García A, Simari GR. Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming, 4(1):95–138, 2004.
11. McCarthy J, Hayes P. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In Meltzer B and Mitchie D, editors, Machine Intelligence 4, pages 463–502. Edinburgh University Press, 1969.
12. Reiter R. A Logic for Default Reasoning. Artificial Intelligence, 13(1,2):81–132, Apr. 1980.
13. Nute D. Defeasible Reasoning. In Fetzer J. H, editor, Aspects of Artificial Intelligence, pages 251–288. Kluwer Academic Publishers, Norwell, MA, 1988.
14. Poole D. On the Comparison of Theories: Preferring the Most Specific Explanation. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pages 144–147. IJCAI, 1985.
15. Poole D. Explanation and Prediction: an Architecture for Default and Abductive Reasoning. Computational Intelligence, 5:97–110, 1989.
16. Chesñevar CI, Maguitman A, Loui R. Logical Models of Argument. ACM Computing Surveys, 32(4):337–383, Dec. 2000.
17. Prakken H, Vreeswijk G. Logical Systems for Defeasible Argumentation. In Gabbay D and F.Guenther , editors, Handbook of Phil. Logic, pages 219–318. Kluwer, 2002.
18. Simari GR, Loui R. A Mathematical Treatment of Defeasible Reasoning and its Implementation. Artificial Intelligence, 53:125–157, 1992.
19. Stolzenburg F, García A, Chesñevar CI, Simari GR. Computing Generalized Specificity. J. of Non-Classical Logics, 13(1):87–113, 2003.
20. Chesñevar CI, Maguitman A. ARGUENET: An Argument-Based Recommender System for Solving Web Search Queries. In Proc. of the 2nd IEEE Intl. IS-2004 Conference. Varna, Bulgaria, pages 282–287, June 2004.
21. Chesñevar CI, Maguitman A, Simari GR. A first approach to argument-based recommender systems based on defeasible logic programming. In Proc. of the 10th Intl. Workshop on Non-Monotonic Reasoning (NMR-2004). Whistler, Canada, pages 109–117, June 2004.
22. Gómez S, Chesñevar CI. A Hybrid Approach to Pattern Classification Using Neural Networks and Defeasible Argumentation. In Proc. of 17th Intl. FLAIRS Conference. Miami, Florida, USA, pages 393–398. American Association for Artificial Intelligence, May 2004.
23. Capobianco M, Chesñevar CI, Simari GR. An argument-based framework to model an agent's beliefs in a dynamic environment. In Proc. of the First International Workshop on Argumentation in Multiagent Systems. AAMAS 2004 Conference, New York, USA, pages 163–178, July 2004.
24. Vinay J, Darbelnet J. Stylistique comparée du franc̃ais et de l'anglais. Didier, 1973.

25. Sabaté M, Chesñevar CI. False friends in computer science literature. Perspectives: Studies in Traslatology, 6(1):47–60, 1998.
26. Baker M. In Other Words. A Coursebook on Translation. Routledge, 1992.
27. Newmark P. A Textbook of Translation. Prentice Hall International Language Teaching, 1988.
28. Chesñevar CI, Maguitman A. An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In Proc. of the ECAI-2004 Conference. Valencia, Spain, pages 581–585, Aug. 2004.
29. Fellbaum C. WordNet: An Electronic Lexical Database. MIT Press, 1998.
30. Kukich K. Techniques for automatically correcting words in text. ACM Computing Surveys, 24(4):377–439, 1992.
31. Church K, Rau L. Commercial applications of natural language processing. CACM, 38(11):71–79, November 1995.
32. Stankevicius A, Garcia A, Simari GR. Compilation techniques for defeasible logic programs. In Proc. of the 6th Intl. Congress on Informatics Engineering, pages 1530–1541. Univ. de Buenos Aires, Bs. Aires, Argentina, Ed. Fiuba, Apr. 2000.
33. Chesñevar CI, Simari GR, Alsinet T, Godo L. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In Proc. of the Intl. Conference in Uncertainty in Artificial Intelligence (UAI 2004). Banff, Canada, pages 76–84, July 2004.
34. Chesñevar CI, Dix J, Stolzenburg F, Simari GR. Relating Defeasible and Normal Logic Programming through Transformation Properties. Theoretical Computer Science, 290(1):499–529, 2003.