GramCheck: A Grammar and Style Checker

Flora Ramírez Bustamante
Escuela Politécnica Superior
Universidad Carlos III de Madrid
c/ Butarque 15
28011 Leganés, Madrid, SPAIN
flora@ia.uc3m.es

Fernando Sánchez León
Laboratorio de Lingüística Informática
Facultad de Filosofía y Letras
Universidad Autónoma de Madrid
28049 Madrid, SPAIN
fernando@maria.lllf.uam.es

Abstract

This paper presents a grammar and style checker demonstrator for Spanish and Greek native writers developed within the project GramCheck. Besides a brief grammar error typology for Spanish, a linguistically motivated approach to detection and diagnosis is presented, based on the generalized use of PROLOG extensions to highly typed unification-based grammars. The demonstrator, currently including full coverage for agreement errors and certain head-argument relation issues, also provides correction by means of an analysis-transfer-synthesis cycle. Finally, future extensions to the current system are discussed.

1 Introduction

Grammar checking stemmed as a logical application from former attempts to natural language understanding by computers. Many of the NLU systems developed in the 70's included a kind of error recovery mechanism ranging from the treatment only of spelling errors, PARRY (Parkinson *et al.*, 1977), to the inclusion also of incomplete input containing some kind of ellipsis, LADDER/LIFER (Hendrix *et al.*, 1977).

The interest in the 80's begun to turn considering grammar checking as an enterprise of its own right (Carbonell & Hayes, 1983), (Hayes & Mouradian, 1981), (Heidorn *et al.*, 1982), (Jensen *et al.*, 1983), though many of the approaches were still in the NLU tradition (Charniak, 1983), (Granger, 1983), (Kwasny & Sondheimer, 1981), (Weischedel & Black, 1980), (Weischedel & Sondheimer, 1983). A 1985 Ovum report on natural language applications (Johnson, 1985) already identifies grammar and style checking as one of the seven major applications of NLP. Currently, every project in grammar checking has as its goal the creation of a writing aid rather than a robust man-machine interface (Adriaens, 1994), (Bolioli *et al.*, 1992), (Vosse, 1992).

Current systems dealing with grammatical deviance have been mainly involved in the integration of special techniques to detect and correct, when possible, these deviances. In some cases, these have been incorporated to traditional parsing techniques, as it is the case with feature relaxation in the context of unification-based formalisms (Bolioli *et al.*, 1992), or the addition of a set of catching error rules specially handling the deviant constructions (Thurmair, 1990). In other cases, the relaxation component has been included as a new add-in feature to the parsing algorithm, as in the IBM's PLNLP approach (Heidorn *et al.*, 1982), or in the work developed for the Translator's Workbench project using the METAL MT-system (TWB, 1992).

Besides, an increasing concern in current projects is that of linguistic relevance of the analysis performed by the grammar correction system. In this sense, the adequate integration of error detection and correction techniques within mainstream grammar formalisms has been addressed by a number of these projects (Bolioli *et al.*, 1992), (Vosse, 1992), (Genthial *et al.*, 1992), (Genthial *et al.*, 1994).

Following this concern, this paper presents results from the project GramCheck (A Grammar and Style Checker, MLAP93-11), funded by the CEC. GramCheck has developed a grammar checker demonstrator for Spanish and Greek native writers using ALEP (ET6/1, 1991), (Simpkins, 1994) as the NLP development platform, a client-server architecture as implemented in the X Windows system, Motif as the 'look and feel' interface and Xminfo as the knowledge base storage format. Generalized use of extensions to the highly typed and unification based formalism implemented in ALEP has been performed. These extensions (called Constraint Solvers, CSs) are nothing but pieces of PROLOG code performing different boolean and relational operations over feature values. Besides, GramCheck has used ongoing results from LS-GRAM (LRE61029), a project aiming at the implementation of middle coverage ALEP grammars for a number of European languages.

The demonstrator checks whether a document contains grammar errors or style weaknesses and, if found any, users are provided with messages, suggestions and, for grammar errors only, automatic correction(s).

2 Brief grammar error typology for Spanish

The linguistic statements made by developers of current grammar checkers based on NLP techniques are often contradictory regarding the types of errors that grammar checkers must correct automatically. (Veronis, 1988) claims that native writers are unlikely to produce errors involving morphological features, while (Vosse, 1992) accepts such morpho-syntactic errors, in spite of the fact that an examination of texts by the author revealed that their appearance in native writer's texts is not frequent. Both authors agree in characterizing morpho-syntactic errors as a sample of lack of competence.

On the other hand, an examination of real texts produced by Spanish writers revealed that they do produce morpho-syntactic errors[1]. Spanish is an inflectional language, which increases the possibilities of such errors. Nevertheless, other errors related to structural configuration of the language are produced as well.

Errors found fall into one of the following subtypes, assuming that featurization is the technique used in parsing sentences:

1. Mismatching of features that do not affect representational issues (intra- or inter-syntactic agreement on gender, number, person and case for categories showing this phenomenon). These mismatchings produce *non-structural errors*.
2. Mismatching of features which describe certain representational properties for categories, as wrong head-argument relations, word order and substitution of certain categories. These mismatchings produce *structural errors*.

Table 1 shows the percentages of different types of errors found in the corpus. Punctuation errors must be considered as structural violations, while for style weaknesses, it depends on its subtype. Errors at the lexical level are difficult to classify and most of them must regarded as spelling rather than grammar errors. The number of errors identified in this corpus is 543. These statistics could be regarded as a representative average of the frequency of errors/mistakes occurring in Spanish texts.

[1]The corpus used contains nearly 70,000 words including text fragments from literature, newspapers, technical and administrative documentation. It has been provided to a large extent by GramCheck pilot user, ANAYA, S.A.

Table 1: Statistics of errors

Type of error		Percentage
Non-structural violations as described above		18.5
Structural violations as described above		9.7
Punctuation	Omission	32.2
	Addition	4.8
Errors at the lexical level[a]	At the character level	6.3
	Stress	8.0
Style weaknesses	Structural	3.5
	Lexical	12.0
Others		5.0

[a]Errors at the lexical level include typing errors, word segmentation (*si no* vs *sino*), and cognitive errors (*onceavo* (partitive) vs *undécimo* (ordinal).

A presupposition adopted in the project led to the idea that violations at the feature level can be captured by means of the relaxation of the possibly violated features while violations at the level of configuration may not be relaxed without raising unpredictable parsing results, thus being candidates for the implementation of explicit rules encoding such incorrect structures.

Under this view, a comprehensive grammar checker must make use of both strategies, called in the literature **feature** or **constraint relaxation** and **error anticipation**, respectively.

However, given the relevance of features in the encoding of linguistic information in TFSs, some structural errors can be reanalyzed as agreement errors in a wide sense (as feature mismatching violations rather than structural ones). This allows the implementation of a uniform approach to grammar correction, thus avoiding explicit rules for ill-formed input. This paper describes such implementation for both non-structural and structural violations.

3 Error detection, diagnosis and correction techniques

The overall strategy for detection, diagnosis and correction of grammar and style errors within GramCheck relies on three axes:

- For detection, a combined *feature relaxation* and *error anticipation* approach is adopted. In order to implement the former, extensive use of external CSs is performed in the analysis grammar, whereas for the latter, explicit rules, adequately defined either in the core grammar or in satellite subgrammars, are implemented[2].

[2]GramCheck checks texts belonging to the standard language and to the administrative sublanguage. The analysis module has been conceived as composed by a core grammar and two satellite subgrammars for overlapping cases— that are mutually exclusive. Thus, the activation of one subgrammar implies the

- Diagnosis is performed by the CSs themselves with the aid of a ***heuristic technique*** for those errors where tests should be performed on several elements and a *pattern-related technique* which provides a means to extend feature values with a gradation of correct and posible but incorrect information. The typical case for the former is agreement, thus for signs involving this type of information, both an initial heuristic value is assigned and arithmetical operations are performed on (in)equality tests. As for the latter, head-argument relations where bound prepositions are involved are treated this way. For all grammar errors there is no notion of weak vs. strong diagnosis, being all considered strong errors needing automatic correction.
- Correction is performed by means of tree transduction of Linguistic Structures (LSs) containing errors to a 'language' (actually a 'language use') defined as *correct Spanish*. These synthesized structures are displayed to the user. The overall design is then similar to a transfer-based MT system, where the usual cycle is analysis-transfer-synthesis, being the main differences the addition of the above-mentioned grammar correction devices and the fact that not all, but only incorrect sentences, will be pushed through the complete cycle.

CSs allow the relaxation of certain features in the grammar rules whose unification will be decided upon, in a non-trivial way, within these CSs. Thus, rules do not perform feature value checking, so CSs play a crucial role performing extended variable unification and taking appropriate decisions. Depending on the error type, CSs carry out different operations on features, scores and lists. These operations concern basically the detection and the evaluation of the error, providing a diagnostic on the error and correct value(s) for features involved. The use of CSs favours a one step diagnosis procedure where decisions are only taken once all the relevant information is gathered.

3.1 A heuristic technique to diagnose non-structural errors

CSs are used in GramCheck roughly in the way presented in (Crouch, 1994) and (Ruessink, 1994), developers of CSs for ALEP. Nevertheless, while these reports describe constraint solving as an extension to the expressive power of grammar rules, the novelty of the approach presented here is the use of CSs to allow feature relaxation in rules and boolean, relational and arithmetical operations on relevant features.

deactivation of the other, and in both cases they are added to the core grammar depending of the type of (sub)language selected by the user.

Agreement errors pose a problem for a grammar checker which parses natural language, because the detection of the error and the diagnosis procedure have to be performed automatically. In inflectional languages, like Spanish, this issue is essential given that in certain contexts it is not possible to give a single correction when performing analysis only at sentence level (i.e. without anaphoric relations). For these cases, the system should be provided with a heuristics for the correction in order to detect and diagnose the place(s) where the error(s) has(have) been produced and to take a decision about the unit(s) to be corrected. For GramCheck, this heuristics relies on a parametrization of two assumptions:

1. the constituent which holds the feature values that in a given error situation control the rest of the feature values in the other constituents,
2. the evaluation of the number of constituents which share and do not share the same values.

Our diagnosis procedure assumes that the gender and number features in the head of a phrase control those in the dependent constituent(s), although, as it will be proved later, this is not necessarily true. In order to do this diagnosis procedure, the CS will contrast those features and leave some clues of this evaluation in phrasal projections in order for these to be available for further operations should they were necessary. These clues are shaped as scores in the approach adopted for agreement errors, and, in this sense, our heuristics is closed to the metric operations performed by other grammar checkers based on NLP techniques (Veronis, 1988), (Bolioli *et al.*, 1992), (Vosse, 1992), (Genthial *et al.*, 1994).

The core of this heuristics is that depending on a set of linguistic principles based on lexico-morphological properties, the values for gender and number in certain lexical units will be promoted over the values in other units, thus, assigning them a higher score.

There are several conditions which have to be taken into account in order to perform the diagnosis procedure. For instance, nouns with inherent gender should control the gender of the rest of the elements in a given NP. However, if the noun does not have inherent gender — it's a noun that shows sex inflection — then the gender value should be controlled by those elements that, sharing the same value, are majority. Hence, a sequence like *el*_masc *casa*_fem (the house) must be corrected into *la*_fem *casa*_fem because this noun has inherent feminine gender in Spanish. On the other hand, an NP like *la*_fem *chic-o*_masc *guap-a*_fem (lit. 'the boy beautiful') should be corrected as *la*_fem *chic-a*_fem *guap-a*_fem ('the girl beautiful'), thus changing the gender value of the head noun in the direction suggested by the other dependent elements. This means that although the system could take the gender value of the head as the value which commands the whole phrase, the number of elements that share the same feature values, if in contrast to those of the head and if the head takes its agreement properties from morphology —ie. are susceptible of keystroke errors-–, can influence the final decision. Finally, for cases where equal scores are obtained, as it happens with a non-inherent masculine noun and a feminine determiner, both possible corrections should be performed, since there is not enough information so as to decide the correct value (unless this can be obtained from other agreeing elements in the sentence —for instance an attribute to this NP).

Basically, the final operation to be performed with the scores is to determine that the higher the score of an element the severer its substitution. Thus, scores are clues for the correction of those elements having the lowest scores.

The initialization steps in order to perform the heuristic technique are related to the assignment of values and scores to lexical projections depending on its inherentness. The values for gender and number of the head of the projection serve as a parameter for the computation of values and scores for the possible modifier which could appear closed to it. Note that agreement in Spanish is based on a binary value system. Thus, the computation of values for the modifier of a given head simply relies on the instantiation of opposite values to those of the head. In the case of underspecification of the head for gender, for instance, the presupposition is that this value is the same as the one of the modifier, if this is not underspecified. Otherwise, both elements remain underspecified. Besides, the weight given to controlling elements (50) ensures that there is no way for modifiers to overpass this score. Note as well that the weight given to inherentless values, as number (10), ensures that there are no promoted elements in this calculation. The following schematic CS illustrates the assignment of scores:

```
and(=(Score'number'head,10),
  and(
   or(and(=(Inherentness'head,yes), =(Score'gender'head,50)),
      and(=(Inherentness'head,no), =(Score'gender'head,10))),
   =(Score'number'mod,0)))
```

The following steps to be performed by CSs are related to the addition of all those scores associated to a given value in the successive rules building the nominal projection and the percolation of morphosyntactic features:

```
or(
  and(or(=(Gender'head'mother,Gender'mod),
        =(Gender'mod,masc'fem)),
     and(num'add(MGEN'SCORE'HEAD,
               MGEN'SCORE'MOD,
               MGEN'SCORE'MOTHER),
        and(=(Gender'mod'head,Gender'mod'mother),
           num'add(HGEN'SCORE'HEAD,
                  HGEN'SCORE'MOD,
                  HGEN'SCORE'MOTHER)))),
  and(=(Gender'mod,Gender'mod'head),
     and(num'add(HGEN'SCORE'HEAD,
               MGEN'SCORE'MOD,
               HGEN'SCORE'MOTHER),
        and(=(Gender'mod'head,Gender'mod'mother),
           num'add(MGEN'SCORE'HEAD,
                  HGEN'SCORE'MOD,
                  MGEN'SCORE'MOTHER)))))
```

The final evaluation performed by CSs is done when categories showing agreement overpass their maximal projection, only if no other inter-syntagmatic agreement must be taken into account (as it is the case with subject-attribute agreement, for instance). Postponing in this way the final evaluation ensures that the CS will take into account all the previous parameters to give an appropriate diagnosis about the complete XP containing the agreement violation. This evaluation is based on the comparison of scores by means of the 'greater than' predicate in order to determine (a) the correct value for the feature(s) checked corresponding to the highest score(s) (`Right_Gender, Right_Number` in the example below), to be used by the transfer module, and (b) the error diagnosis (`gender`, `number` and `gender_number` below), to be used by the error handling module that will display appropriate error information to the user:

```
and(
or(
   or(and(num'gt(HGEN'SCORE'NOUN,MGEN'SCORE'NOUN),
            =(Gender'Noun,Right'Gender)),
      and(num'gt(MGEN'SCORE'NOUN,HGEN'SCORE'NOUN),
            =(Gender'Mod,Right'Gender))),
   =(HGEN'SCORE'NOUN,MGEN'SCORE'NOUN)),
or(
   or(and(num'gt(HNUM'SCORE'NOUN,MNUM'SCORE'NOUN),
           =(Number'Noun,Right'Number)),
      and(num'gt(MNUM'SCORE'NOUN,HNUM'SCORE'NOUN),
           =(Number'Mod,Right'Number))),
   =(HNUM'SCORE'NOUN,MNUM'SCORE'NOUN))),
```

If all elements agree, scores for one of the arguments will always be 0, while if this argument has a value different than 0, this information is considered as an evidence that an error has occurred, the subsequent comparison determining the value for the winning score:

```
or(
  and(=(MNUM'SCORE'MOTHER,0),
     or(=(MGEN'SCORE'MOTHER,0),
        and(num'gt(MGEN'SCORE'MOTHER,0),
           =(ERTYPE,gender)))),
  and(num'gt(MNUM'SCORE'MOTHER,0),
     or(and(=(MGEN'SCORE'MOTHER,0),
           =(ERTYPE,number)),
        and(num'gt(MGEN'SCORE'MOTHER,0),
           =(ERTYPE,gender'number)))))
```

3.2 A pattern-related technique to perform structural error detection/diagnosis

Turning back to the general definitions on error types given at the beginning of this document, structural violations can be seen as special cases of feature mismatching produced by addition, substitution and omission of elements which result in a wrong dependency relation:

Wrong head-argument relations

(i) Substitution of a bound preposition by another one (PP $\mapsto$ PP)

*Los alumnos relacionan la tarea [*a/con] su conocimiento.*

(ii) Omission of a bound preposition resulting in a change of the subcategorized argument (PP $\mapsto$ NP/S)

Se acordó [/de] que tenía una reunión por la mañana.*

(iii) Addition of a preposition resulting in a change of the subcategorized argument (NP/S $\mapsto$ PP)

*Las empresas demandan [*de] métodos.*

In the HPSG-like grammar used, bound prepositions are considered NPs attached to the `subcat` list (ie. the subcategorization feature) of a predicative unit. These NPs have the feature `pform` instantiated to the value of the preposition, if any. If the argument does not have a bound preposition, the value for `pform` is `none`. Thus, the approach adopted within GramCheck is that these error cases have a correct representation of the dependency structure where the only offending information is stored as a feature in the governed element.

The linguistic principle behind the pattern-related technique is based on the fact that native writers substitute a preposition by another one when certain associations between patterns, showing either the same lexico-semantic and/or syntactic properties, are performed. Thus, this kind of error is not so accidental as it could be imagined.

For instance, Spanish speakers/writers usually associate the argument structure of the comparative adjective *inferior* (lower), which subcategorizes the preposition *a* (to), with the Spanish comparative syntactic pattern (*menos ... que*, less ... than) whose second term is introduced by the conjunction *que*, producing phrases such as **inferior que* instead of *inferior a*. With the verb *relacionar* (to relate), something similar occurs: this verb subcategorizes for the preposition *con*; however, due to the fact that there exists the prepositional multi-word units *en relación* **a** and *en relación* **con**, speakers tend to think that the same prepositional alternation can be performed with the verb (**relacionar a* vs. *relacionar con*).

Following this idea, configurational rules are regarded, for grammar checking, as descriptions of patterns, each of them having associated a wrong pattern linked to the correct pattern. Both patterns are in a complementary distribution. This way, structural errors can be foreseen and controlled, and the system is provided with a mechanism which establishes the way rule constraints must be relaxed.

To cope with this error, a CS operating on lists checks whether the preposition in the constituent attached to the predicative sign belongs to the head of the list or to the tail. If the preposition is member of the tail, the same actions shown for agreement errors are performed — instantiation of the correct value and determination of the error type.

4 Error coverage

The current version of the GramCheck demonstrator is able to deal with the following types of errors:

- Intra- and inter-syntagmatic agreement errors (gender and/or number in active — with both predicative and copulative verbs — and passive sentences).
- Direct objects: omission of the preposition *a* with an animate entity and addition of such a preposition with a non-animate entity.
- Addition, omission and substitution of a bound preposition covering what is called

dequeísmo —the addition of a false bound preposition de with clausal arguments— and *queísmo* —the omission of the bound preposition de with clausal arguments.

- Errors on *portmanteau* words (use *de el, a el* instead of *del, al*).

Regarding style issues, three different types of weaknesses are detected: structural weaknesses, lexical weaknesses and abusive use of passive, gerunds and manner adverbs. While structural weaknesses are detected in the phrase structure rules using CSs (noun + "a" + infinitive), by means of an error anticipation strategy, lexical weaknesses are detected at the lexical level, with no special mechanisms other than simple CSs. Lexical errors currently detected are related with the use of Latin words which it is better to avoid, foreign words with Spanish derivation, cognitive errors, foreign words for which a Spanish word is recommended and verbosity.

5 Further developments

Results obtained with the current demonstrator are very promising. The performance of the system using CSs is similar to that shown without them, hence its use in conjunction with the detection techniques proposed, rather than a burden, may be seen as a means to add robustness to NLP systems. In fact, CSs may provide more natural solutions to grammar implementation issues, like PP-attachment control.

Several directions for further developments have already been defined. These include the integration of these grammar checking techniques into the final release of the LS-GRAM Spanish grammar, which will have a more realistic coverage in terms both of linguistic phenomena and lexicon. Besides, on this new version of the grammar, hybrid techniques will be used, taking advantage of the preprocessing facilities included in ALEP. In particular, while for errors like those presented in this paper the approach adopted is linguistically motivated, for certain punctuation errors (or simply in order to reduce lexical ambiguity) other relatively simple means can be defined that include certain extended pattern matching on regular expressions or the passing of linguistic information gathered in a preprocessing phase to the unification-based parser. It is also foreseen to include a treatment for *cognitive spelling errors*, usually not dealt with by conventional spelling checkers.

References

Adriaens G. 1994. The LRE SECC Project: Simplified English Grammar and Style Correction in an MT Framework, in *Proceedings of Linguistic Engineering Convention*, Paris.

Bolioli A., L. Dini, G. Malnati. 1992. JDII: Parsing Italian with a Robust Constraint Grammar, in *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*:1003-1007.

Carbonell J. and P. Hayes. 1983. Recovery strategies for parsing extragrammatical language, in *American Journal of Computational Linguistics*, 9(3-4):123-146.

Charniak E. 1983. A parser with something from everyone, in M. King (ed.), *Parsing Natural Language*, Academic Press, London.

Crouch R. 1994. *A prototype ALEP Constraint Solver*, SRI International, Cambridge Research Centre.

Pulman, S. G. (ed.). 1991. *EUROTRA ET6/1: Rule Formalism and Virtual Machine Design Study*, SRI International, Cambridge Computer Science Research Centre, ©Commission of European Communities.

Genthial D., J. Courtin *et al.*. 1992. From Detection/Correction to Computer Aided Writing, in *Proceedings of the 15th International Conference on Computational Linguistics (COLING-92)*:1013-1018.

Genthial D., J. Courtin, J. Ménézo. 1994. Towards a more user-friendly correction, in *Proceedings of the 16th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan.

Granger R. 1983. The NOMASD system: expectation-based detection and correction of errors during understanding of syntactically and semantically ill-formed text, *American Journal of Computational Linguistics*, 9(3-4):188-196.

Hayes P. and G. Mouradian. 1981. Flexible Parsing, *American Journal of Computational Linguistics*, 7(4):232-244.

Heidorn G. E., K. Jensen, L. A. Miller, R. J. Byrd and M. S. Chodorow. 1982. Developing a Natural Language Interface to Complex Data, *ACM Trans. on Database Sys.*, 3(2):105-147.

Hendrix G. G, E. D. Sacerdoti, D. Sagalowicz and J. Slocum. 1977. Developing a Natural Language Interface to Complex Data, *ACM Trans. on Database Sys.*, 9(2):105-147.

Jensen K., G. E. Heidorn, L. A. Miller, Y. Rabin. 1983. Parse fitting and prose fixing: getting a hold on ill-formedness, *American Journal of Computational Linguistics*, 9(3-4):147-160.

Johnson T. 1985. *Natural Language Computing: The Commercial Applications*, Ovum, London.

Kwasny S. and N. Sondheimer. 1981. Relaxation techniques for parsing grammatically ill-formed input in natural language understanding systems, *American Journal of Computational Linguistics*, 7(2):99-108.

Parkinson R. C, K. M. Colby and W. S. Faught. 1977. Conversational Language Comprehension Using Integrated Pattern-Matching and Parsing , in *Artificial Intelligence*, 9:111-134.

Ruessink H. 1994. *Manual to the RGR extensions for ALEP*, STT/OTS, preliminary version.

Simpkins N. K. 1994. *An Open Architecture for Language Engineering: The Advanced Language Engineering Platform (ALEP)*. Proceedings of LE2, Paris.

Thurmair G. 1990. Parsing for grammar and style checking, in *Proceedings of the 14th International Conference on Computational Linguistics (COLING-90)*:365-370.

Translator's Workbench. 1992. Final Report of the Esprit Project 2315.

Veronis J. 1988. Morphosyntactic correction in natural languages interfaces, in *Proceedings of the 13th International Conference on Computational Linguistics (COLING-88)*:708-713.

Vosse, T. 1992. Detecting and Correcting Morphosyntactic Errors in Real Texts, *Proceedings of the 3rd Conference on Applied Natural Language Processing (ACL-92)*:111-118.

Weischedel R. and J. Black. 1980. Responding intelligently to unparseable inputs, in *American Journal of Computational Linguistics*, 6(2):97-109.

Weischedel R. and N. Sondheimer. 1983. Metarules as a basis for processing ill-formed input, in *American Journal of Computational Linguistics*, 9(3-4):161-177.

Weischedel R. and L. A. Ramshaw. 1987. Reflections on Knowledge needed to process ill-formed language. In S. Nirenburg (ed.) *Machine Translation. Theoretical and Metholodogical Issues*, Cambridge, Cambridge University Press, pp. 155-337.