Little Things With Big Effects: On the Identification and Interpretation of Tokens for Error Diagnosis in ICALL

Luiz A. Amaral and W. Detmar Meurers

1 Introduction

Intelligent Computer-Assisted Language Learning (ICALL) systems differentiate themselves from traditional CALL systems through their ability to analyze learner input. They can identify language properties and diagnose errors, which in principle allows ICALL systems to provide specific, individualized feedback for a wider range of learner input and activity types.

Error diagnosis can be conceived as a process abstracting from the learner's production to a set of linguistic features that best describe the learner's (mis)-conceptions of the linguistic structures represented in a given input string.[1] This process may comprise several steps that take into consideration morphological, syntactic, and semantic properties of the input. However, it almost invariably starts with the identification of the basic linguistic units that will serve as the building blocks of the analysis, i.e., the identification and interpretation of tokens.

In this paper, we discuss the identification and interpretation of tokens and the mismatches that can arise in an ICALL context between the learner's conceptualization of a given token and the system's interpretation of its linguistic properties.

The general issue is made concrete using real-life examples from the error diagnosis performed by TAGARELA, an ICALL system for Portuguese. We tested the system with students from introductory Portuguese courses at the Ohio State University in Spring 2007. Analyzing the logs which record what the students

[1] In addition to the linguistic properties, error diagnosis may also benefit from an analysis of extra-linguistic properties (cf. Amaral & Meurers 2008). In the current paper, we focus exclusively on the linguistic aspects.

entered and what the system returned as feedback, we identified several apparent mismatches between how some learners conceptualize tokens and the way the system analyzes them linguistically and represents them.

We focus on two specific cases, the representation of contractions and the interpretation of accented characters. We show that the mismatches arising in such cases can be addressed in a general way by building ICALL systems on an annotation-based Natural Language Processing (NLP) architecture which monotonically enriches the representation of the learner input. As background for the discussion of the two specific cases in section 3, we start with a discussion of the relevant aspects of the TAGARELA system and annotation-based processing.

2 ICALL background

2.1 TAGARELA

TAGARELA (Teaching Aid for Grammatical Awareness, Recognition and Enhancement of Linguistic Abilities) is an intelligent web-based workbook for learners of Portuguese (Amaral & Meurers 2006, 2007, 2008). The system can be used as a pedagogical complement in traditional classroom settings, as well as in distance learning or individualized instruction programs. It includes six activity types: *listening comprehension*, *reading comprehension*, *picture description*, *fill-in-the-blanks*, *rephrasing*, and *vocabulary*. These activities provide opportunities for students to practice their listening, reading, and writing skills. The expected input consists of words, phrases or sentences.

Different from paper-based workbooks, TAGARELA offers on the spot individualized feedback on orthographic errors (non-words, spacing, capitalization, punctuation), syntactic errors (verbal and nominal agreement), and semantic errors (missing concepts, extra concepts, word choice). In contradistinction to traditional CALL exercises, specific, individualized feedback can be provided even for activities which allow a wide range of variation in the vocabulary, the morphological form, the word order, and the syntactic constructions used by the learner. For all activity types, the answers are checked by the system, i.e., the generation of feedback is completely automated.

2.2 System Architecture

The TAGARELA architecture shown in Figure 1 consists of six modules: the

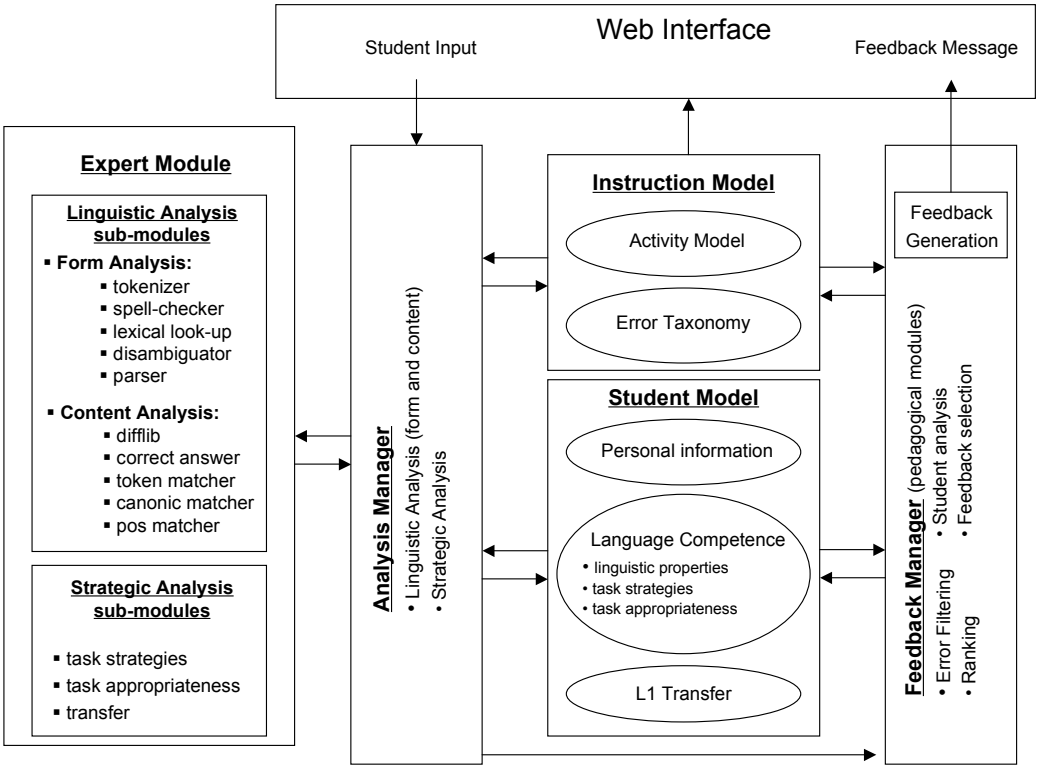

Figure 1: Architecture of the TAGARELA system

Interface, the Analysis Manager, the Feedback Manager, the Expert Module, the Instruction Model, and the Student Model.

After being entered into the web-based interface, the student input is sent to the Analysis Manager for processing, which calls the NLP modules that are part of the Expert Module of the system. The tokenizer takes into account specific properties of Portuguese, such as cliticization, contractions, and abbreviations – issues we turn to in section 3.

After tokenization, the input is checked for non-word spelling errors using a standard spell-checker (Ispell, Kuenning 2005) with Brazilian Portuguese parameter files. Full-form lexical lookup then returns multiple analyses based on the CURUPIRA lexicon (Martins et al. 2006), including detailed morphological information.

In the spirit of Constraint Grammar (Karlsson et al. 1995; Bick 2000, 2004), disambiguation rules are used to narrow down the multiple lexical analyses based

on the local context. Complementing these local disambiguation rules, a simple bottom-up parser using hand-written rules checks agreement, case relations, and some global well-formedness conditions.

In addition to the form-focused processing, content assessment is performed using shallow semantic matching between the student answer and target answers provided by the teacher in the Activity Model, essentially a basic version of the approach discussed in Bailey & Meurers (2008).

Annotation-based processing Following up on on the characterization of the various relevant NLP modules, we can now turn to the general question of how such NLP modules can be combined and how the results of the analyses are represented. Traditionally, in ICALL systems the NLP algorithms and resources have been integrated into a basic pipeline architecture (cf, e.g., Levin & Evans 1995; Heift 2003; Nagata this volume). Such systems call the NLP modules in a predefined order, transforming one data structure into another and terminating when specific conditions are met – for example, when the learner response matches a pre-stored target response, or when spell checking fails. Pipeline architectures work well as long as the system deals with learner input from activity types that are uniform with respect to the required NLP processing.

A pipeline architecture can become problematic, however, when trying to integrate a wider range of activity types, providing learner input of a variable and heterogeneous nature. As we argued in Amaral & Meurers (2007), the use and sequencing of the different NLP modules ideally should then be triggered by demands for particular information based on the activity models of the different activity types. Each NLP module enriches the input with annotations until all information required to evaluate the learner's performance on a particular activity is present. Just like in the annotation of corpora, an annotation-based ICALL system enriches the learner input with layers of information about the input, from general linguistic properties such as the part-of-speech of the tokens, to more specialized analyses of learner language properties or error types. The single algorithmic pipeline is replaced by whatever NLP processing is required to obtain the information needed to provide feedback for a specific activity. It thus becomes possible to provide individualized feedback to a range of activities that differ in the nature and amount of information that needs to be identified by the NLP, complementing whatever is explicitly specified in the activity model.

In the TAGARELA architecture, the Analysis Manager is designed to realize this demand-driven annotation process. The current system includes a basic ver-

sion of the Analysis Manager which collects information from the Activity Model (as part of the Instruction Model) to decide on the sequence of NLP modules to call. The annotated input, i.e., the learner input enriched with the results of the NLP analyses, is passed on to the *Feedback Manager*. In addition to the annotated learner input, the Feedback Manager can in principle consult the student model and the activity model. Based on all this information, it filters and prioritizes the errors that should be targeted and decides on the best feedback message to generate. Because the input is annotated with the output of every NLP module that was called, the Feedback Manager can use any of the information provided by any of the input processing to decide on and formulate the feedback message. This will become crucial in sections 3.1 and 3.2.

Finally, the explicit *Instruction Model* and the *Student Model* included in Figure 1 are the repository of information about activities and the student already mentioned above. They complement the information obtained through NLP analysis of the learner input and can be seen as guiding the processing mechanism from linguistic analysis to feedback generation.

3 Little Things with Big Effects

Following a general scaffolding methodology to help the learner develop self-editing skills (cf., e.g., Hyland & Hyland 2006), the TAGARELA system provides feedback that is designed to lead the student to producing the correct answer. Yet when we manually inspected the log files documenting the input by the students and their feedback on the system's response, we identified certain student-system interactions where the feedback of the system did not seem to result in improved student input. Some of those students also used a system option allowing them to report being confused by the system feedback in those cases. Many of these cases involved the use of contractions and accented characters.

We discuss both of these cases in this chapter, in turn, starting with an illustration of the failed learner-system interactions, followed by enough background on Portuguese to understand what goes on in those cases, and ending with the conclusions on how to address the mismatch between the system analyses and the apparent learner conceptualizations. We will see that the question of how foreign language tokens are identified and interpreted by language learners has general implications for the design of ICALL systems.

3.1 Contractions

An instance of the problem The issue around contractions is clearly represented by a learner-system interaction in which the student was answering a reading comprehension question on a text describing the different regions of Brazil. One of the possible ways to answer the question is shown in (1a). Apparently aiming for this target, the student entered the sentence in (1b), which differs only in the use of *no* in place of the correct *na*.

(1) a. O Amazonas fica **na** região norte.
the Amazon lies in the$_{\text{FEM.SG}}$ region North

b. * O Amazonas fica **no** região norte.
the Amazon lies in the$_{\text{MASC.SG}}$ region North

'The Amazon is in the Northern region.'

In response to the student input in (1b), the system feedback reports an agreement error in gender between the determiner *o* and the noun *região* in the sequence *o região norte*. This feedback message was not helpful for the student, who failed to enter the correct answer in a subsequent attempt.

To be able to explain the source and nature of the problem with the feedback message in this case, and the general issue it illustrates, we first need to provide some basic background on contractions in Portuguese.

Linguistic background Contractions are frequent in Portuguese and are used at all levels, including the beginning learners targeted by our system. Contractions occur, for example, between a preposition and an article, as shown in (2).

(2) Preposition + Article

a. do = de + o
of the = of + the$_{\text{MASC.SING}}$

b. numa = em + uma
in a = in + a$_{\text{FEM.PL}}$

The single expression *do* (*of the*) is used in place of the preposition *de* (*of*) followed by the article *o* (*the*). Note that different from English, articles in Portuguese encode gender and number information, and while ordinary prepositions in Portuguese do not encode such distinctions, the contractions of a preposition and an article are specified for gender and number. For example, the contraction *do* can only combine with masculine singular nouns; for feminine singular noun

phrases one must use the contraction *da*, and for masculine plural the appropriate contraction is *dos*.

Other types of contractions found in Portuguese include prepositions plus personal pronouns, such as in (3), and prepositions plus demonstrative pronouns, such as in (4)[2].

(3) Preposition + Personal Pronoun

a. dela = de + ela
of her = of + her$_{\text{FEM.SING.OBL}}$

b. neles = em + eles
in them = in + them$_{\text{MASC.PL.OBL}}$

(4) Preposition + Demonstrative Pronoun

a. daquele = de + aquele
of that = of + that$_{\text{MASC.SING}}$

b. nestas = em + estas
in these = in + these$_{\text{FEM.PL}}$

Given that these contractions combine several elements of different parts-of-speech and with distinct syntactic functions, it is necessary to decompose them to be able to accurately analyze the syntactic structure when parsing the sentence.

Explaining the mismatch While the system analyzes the contraction in terms of two separate elements to perform the syntactic analysis, students tend to interpret contractions as a single, atomic entity. In our pilot study, students complained about system feedback that pointed to components of a contraction as the source of an error. For example, whenever the system identified a problem with the use of the article *o* in the contraction *do* mentioned in (2a), it reported to students that the wrong article *o* had been used. Students complained by saying that they had not used any article *o* in their answers, showing that they were not fully aware of the compositional nature of contractions in Portuguese. The system feedback messages thus were pedagogically ineffective; they did not help the learner understand the nature of the error, and consequently learners were not able to reformulate their answers.

[2]While in spoken language the use of the contracted forms is obligatory, in written language there are infrequent cases where both the contracted or the non-contracted form are possible. Beginning learners, such as those using TAGARELA, should always use the contracted forms.

3.2 Accents

An instance of the problem The second issue, the interpretation of accented characters, in our system logs frequently involved the use of the conjunction *e* (*and*) instead of *é* (*is*), the third person singular of the verb *ser* (*to be*). Example (5a) shows the system's target answer and (5b) shows the student's input.

(5) a. Marcos **é** brasileiro.
Marcos is Brazilian.

b. * Marcos **e** brasileiro.
Marcos and Brazilian.

For the student input (5b), the system feedback reported that there is a verb missing in the sentence and that it contains unnecessary words. This feedback message was not understood by learners, who generally failed to provide better input in subsequent attempts; some students also reported being confused by this message using a user interface option that allowed them to send feedback about the system performance.

Just as in the case of contractions, we first provide some facts about Portuguese needed to be able to fully appreciate what is going on in such cases.

Linguistic background Portuguese uses 12 accented characters (*à, á, â, ã, é, ê, í, ó, ô, õ, ú, ü*, plus their corresponding uppercase versions) and one additional character *ç*. Accents in Portuguese can be used to indicate the stressed syllable of a word, or to mark differences in vowel pronunciation. This phonological distinction may affect not only the meaning of the word, but often its syntactic properties as well. As we can see from the examples below, in (6) the difference in the accent changes the gender of the noun; *avô* (grandfather) in (6a) is masculine, while *avó* (grandmother) in (6a) is feminine. Similarly, (7) shows an example where the part of speech is affected by the inclusion of the accent; in (7a) *próspero* is the adjective *prosperous*, while (7b) *prospero* is the third person singular of the verb *to prosper* in the present indicative.

(6) a. vovô
grandfather

b. vovó
grandmother

(7) a. próspero
prosperous$_{\text{ADJECTIVE}}$

b. prospero
   $\text{prosper}_{\text{VERB.FIRST.SING}}$

Explaining the mismatch In the student-system interaction logs we looked at, problems often arose when the misuse of an accent in a word changed its part-of-speech, such as in the example (5b) we started with, where TAGARELA reported to the student that a verb was missing from the sentence. As mentioned, most students did not understand this feedback message and some of them complained that the verb *ser* was in fact already part of their answer and correspondingly were unable to identify and fix the reported error. The problem occurs because students interpreted the word *e* as a form of the verb *ser* instead of an entirely different word, the conjunction *and*. Language learners seem to interpret the presence or absence of an accent as a minor variation of the intended character, instead of as two distinct characters.

3.3 Addressing the problem

The problem in both cases, the identification of tokens in contractions and the interpretation of accented characters, is that the system represents and analyzes language in a way that differs from the surface form of the input and the ways learners conceptualize it. This results in feedback which is difficult or impossible to understand for the student.

The solutions we propose for this problem are based on the annotation-based NLP architecture we introduced and motivated in section 2.2. The crucial aspect of such an architecture for our purposes here is that it monotonically enriches the student input instead of transforming one representation into another as is the case in a traditional pipeline architecture. This means that the diagnosis and feedback modules can refer both to the original student's input and to the annotated, linguistic analyses at the same time. The error diagnosis module can take into account the tokenization and linguistic interpretation performed by the system that as such is not visible in the student input, while the feedback module can generate messages making reference to the material visible in the student input.

Contractions Making the solution concrete for the contraction issue, in an annotation-based ICALL system architecture tokenization can annotate the stretch of the input string corresponding to the contraction with the multiple possibilities: as a single token for the surface form of the contraction and, at the same time, as the

two tokens linguistically encoded in the contraction. In addition, one can mark the former annotation as resulting from a surface-oriented tokenization strategy based on whitespace, while the latter annotation can be marked as resulting from linguistic decomposition.

As a result, the NLP modules performing the error diagnosis can be based on the linguistically decomposed tokens, while the feedback manager can report the results of the analysis in terms of the surface-oriented tokens more immediately apparent to the student.

For the example (1b) we discussed in the introduction of the contraction issue in section 3.1, this means that the diagnosis can use the two token representation *de+o* of the contraction *do* to determine the agreement error between the article *o* and the noun *região*. The Feedback Manager, on the other hand, formulates the feedback message reporting the agreement error in terms of the contraction *do* as a single token and the noun *região*.

Before turning to the second issue, there is another aspect of annotation-based architectures worth mentioning which surfaces in connection with the tokenization of contractions: the ability of each NLP module to contribute whatever information can be determined. In the contraction case at hand, tokenization turns out to support unambiguous part-of-speech assignment for otherwise ambiguous tokens. For example, the token *a* in Portuguese can be a preposition (*to*), a pronoun (*her*, clitic direct object), or an article (*the*, feminine singular). But when the contraction *da* is tokenized in terms of its two parts *de* and *a*, the *a* is known to be the article so that the tokenizer can already assign this part-of-speech as an annotation of this token. For the contraction *vê-la*, on the other hand, the tokenizer can determine the two tokens *ver* and *a* and annotate the latter as a clitic pronoun. As a final example, tokenization of *à* results in a token *a* which can be annotated to be a preposition and another token *a* which can be specified to be an article. For these tokens, the part-of-speech thus can already be determined by the tokenizer and a latter part-of-speech annotator can assign the part-of-speech of the other tokens based on this enriched representation.

Accents The solution for mismatches in the interpretation of accented characters is again based on the possibility of encoding multiple possible annotations for the same interval of the input string. Different from the contraction case, where the multiple annotations resulted from applying both a surface-oriented as well as a deeper token analyses, in the accent case we need to encode multiple possible misconceptualizations of the string which are common with language learners.

In the annotation-based architecture, we can just add alternative token interpretations of the string, with each one corresponding to one (mis)conception the learner had in writing the string. This idea is related to the use of word graphs (or lattices) in speech recognition (cf., e.g., Oerder & Ney 1993), where word graphs encode alternative possible interpretations of a given speech signal. In the ICALL domain, Lee & Seneff (2006) also pursued the idea of generating word lattices of candidate corrections for erroneous learner sentences.

For our accented character issue, the multiple possible token annotations of the string arise from adding, changing or dropping accents. This is motivated by our observation that language learners tend to view a character with different accents as the same character with small variations. Rather than adding all possible accents to all possible characters, since we are trying to find existing words of Portuguese which the learner was aiming for but missed in terms of accenting, we can limit the annotation of alternative tokens to those de- or re-accented words which exist in a full-form lexicon of Portuguese. Finally, one can limit the number of potential alternatives to be annotated further in case the likely confusions depend on the L1 of the learner. This can be realized by allowing the alternative token annotator to make reference to the learner model.

Based on the multiple token annotations, the system can determine whether one of the alternative-accent tokens matches a token in the target answer specified in the activity model. If this is the case, the system can directly proceed to reporting this accent mismatch to the learner. This is parallel to the detection of a non-word spelling mistake, in which the TAGARELA system already short-circuits the further annotation process and directly proceeds to providing feedback on this mistake.

4 Conclusion

In conclusion, based on student-learner logs of a pilot study where beginning learners of Portuguese used the intelligent web-based ICALL system TAGARELA, we identified two examples of interaction where the learner's conceptualization of the language to be learned and the system's underlying linguistic model diverge, resulting in inappropriate feedback. The mismatches we discussed result from the different ways contractions and words with accents can be represented as tokens.

An annotation-based NLP architecture provides a flexible, general framework for encoding the multiple token representations, be they the result of linguistic analysis of contractions by the system or potentially confused tokens differing

only in accents. Given multiple, monotonically added annotation layers about the input string, the system can select and compare representations of different nature, representing linguistic analysis or a misanalysis typical for certain types of language learners. This makes it possible to provide feedback based on a representation close to the surface form of the student's input instead of the system's representation of the input. And it also supports diagnosis and feedback generation based on multiple alternative token representation encoding potential learner misconceptualizations.

Based on this analysis of the logs of the learner-system interaction related to the interpretation of tokens, we intend to extend the TAGARELA system to make use of the richer token annotation proposed in this paper. By supporting reference to an explicit encoding of surface-oriented, linguistically-motivated, and learner-misconception based tokenization options, the feedback can become more effective, essentially by prioritizing the student's understanding of the targeted construction.

References

Amaral, L. & D. Meurers (2006). Where does ICALL Fit into Foreign Language Teaching? CALICO Conference. May 19, 2006. University of Hawaii.

Amaral, L. & D. Meurers (2007). Putting activity models in the driver's seat: Towards a demand-driven NLP architecture for ICALL. In *EUROCALL 2007, Symposium on NLP in CALL*. University of Ulster, Coleraine Campus, Ireland. Talk slides, `http://purl.org/icall/handouts/eurocall07-amaral-meurers.pdf`.

Amaral, L. & D. Meurers (2008). From Recording Linguistic Competence to Supporting Inferences about Language Acquisition in Context: Extending the Conceptualization of Student Models for Intelligent Computer-Assisted Language Learning. *Computer-Assisted Language Learning* 21(4), 323–338. URL `http://purl.org/dm/papers/amaral-meurers-call08.html`.

Bailey, S. & D. Meurers (2008). Diagnosing meaning errors in short answers to reading comprehension questions. In J. Tetreault, J. Burstein & R. D. Felice (eds.), *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications, held at ACL 2008*. Columbus, Ohio: Associa-

tion for Computational Linguistics, pp. 107–115. URL http://aclweb.org/anthology-new/W/W08/W08-0913.pdf.

Bick, E. (2000). *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus, Denmark: Aarhus University Press.

Bick, E. (2004). PaNoLa: Integrating Constraint Grammar and CALL. In H. Holmboe (ed.), *Nordic Language Technology, Arbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004 (Yearbook 2003)*, Copenhagen: Museum Tusculanum, pp. 183–190.

Heift, T. (2003). Multiple learner errors and meaningful feedback: A challenge for ICALL systems. *CALICO Journal* 20(3), 533–548.

Hyland, K. & F. Hyland (2006). Feedback on Second Language Students' Writing. *Language Teaching* 39(2), 1–46.

Karlsson, F., A. Voutilainen, J. Heikkilä & A. Anttila (eds.) (1995). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Berlin and New York: Mouton de Gruyter.

Kuenning, G. (2005). International Ispell Web, Version 3.3.02. URL http://ficus-www.cs.ucla.edu/geoff/ispell.html.

Lee, J. & S. Seneff (2006). Automatic Grammar Correction for Second-Language Learners. In *INTERSPEECH 2006 – ICSLP*. URL http://groups.csail.mit.edu/sls/publications/2006/IS061299.pd%f.

Levin, L. & D. Evans (1995). ALICE-chan: A Case Study in ICALL Theory and Practice. In V. Holland, J. Kaplan & M. Sams (eds.), *Intelligent Tutoring Systems. Theory Shaping Technology*, New Jersey: Lawrence Erlbaum Associates, Inc., pp. 77–97.

Martins, R., R. Hasegawa & M. das Graças Nunes (2006). Curupira: a functional parser for Brazilian Portuguese. In *Computational Processing of the Portuguese Language, 6th International Workshop, PROPOR. Lecture Notes in Computer Science 2721*. Faro, Portugal: Springer.

Nagata, N. (this volume). Robo-Sensei NLP-based Error Detection and Feedback Generation. *CALICO Journal* .

Oerder, M. & H. Ney (1993). Word graphs: an efficient interface between continuous-speechrecognition and language understanding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993 (ICASSP-93). 27–30 April 1993*. vol. 2, pp. 119–122.