Developing Online ICALL Exercises for Russian

Markus Dickinson
Department of Linguistics
Indiana University
md7@indiana.edu

Joshua Herring
Department of Linguistics
Indiana University
jwherrin@indiana.edu

Abstract

We outline a new ICALL system for learners of Russian, focusing on the processing needed for basic morphological errors. By setting out an appropriate design for a lexicon and distinguishing the types of morphological errors to be detected, we establish a foundation for error detection across exercises.

1 Introduction and Motivation

Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy, aiding learners in the development of awareness of language forms and rules (see, e.g., Amaral and Meurers, 2006, and references therein) by providing additional practice outside the classroom to enable focus on grammatical form. But such utility comes at a price, and the development of an ICALL system takes a great deal of effort. For this reason, there are only a few ICALL systems in existence today, focusing on a limited range of languages.

In fact, current systems in use have specifically been designed for three languages: German (Heift and Nicholson, 2001), Portuguese (Amaral and Meurers, 2006, 2007), and Japanese (Nagata, 1995). Although techniques for processing ill-formed input have been developed for particular languages (see Vandeventer Faltin, 2003, ch. 2), many of them are not currently in use or have not been integrated into real systems. Given the vast array of languages which are taught to adult learners, there is a great need to develop systems for new languages and for new types of languages.

There is also a need for re-usability. While there will always be a significant amount of overhead in developing an ICALL system, the effort involved in producing such a system can be reduced by reusing system architecture and by adapting existing natural language processing (NLP) tools. ICALL systems to date have been developed largely independently of each other (though, see Felshin, 1995), employing system architectures and hand-crafted NLP tools specific to the languages they target. Given the difficulty involved in producing systems this way for even a single language, multilingual systems remain a distant dream. Rather than inefficiently "reinventing the wheel" each time we develop a new system, however, a sensible strategy is to adapt existing systems for use with other languages, evaluating and optimizing the architecture as needed, and opening the door to eventual shared-component, multilingual systems. Furthermore, rather than hand-crafting NLP tools specific to the target language of individual systems, it makes sense to explore the possibility of adapting existing tools to the target language of the system under construction, developing resource-light technology that can greatly reduce the effort needed to build new ICALL systems. In this light, it is important to determine where and how reuse of technology is appropriate.

In this spirit, we are developing an ICALL system for beginning learners of Russian based on the TAGARELA system for Portuguese, reusing many significant components. The first priority is to determine how well and how much of the technology in TAGARELA can be adapted for efficient and accurate use with Russian, which we outline in section 2.

1

Focusing on Russian requires the development of techniques to parse ill-formed input for a morphologically-rich language. Compared with other languages, a greater bulk of the work in processing Russian is in the morphological analysis. As there are relatively few natural language processing tools freely available for Russian (though, see Sharoff et al., 2008), we are somewhat limited in our selection of components.

In terms of shaping an underlying NLP system, though, the first question to ask for processing learner input is, what types of constructions need to be accounted for? This can be answered by considering the particular context of the activities. We therefore also need to outline the types of exercises used in our system, as done in section 3, since constraining the exercises appropriately (i.e., in pedagogically and computationally sound ways) can guide processing. Based on this design, we can outline the types of errors we expect to find for morphologically-rich languages, as done in section 4. Once these pieces are in place, we can detail the type of processing system(s) that we need and determine whether and how existing resources can be reused, as discussed in section 5.

2 System architecture

Our system is based on the TAGARELA system for learners of Portuguese (Amaral and Meurers, 2006, 2007), predominantly in its overall system architecture. As a starting point, we retain its modularity, in particular the separation of activities from analysis. Each type of activity has its own directory, which reflects the fact that each type of activity loads different kinds of external files (e.g., sound files for listening activities), and that each type of activity could require different processing (Amaral, 2007).

In addition to the modular design, we also retain much of the web processing code - including the programming code for handling things like user logins, and the design of user databases, for keeping track of learner information. In this way, we minimize the amount of online overhead in our system and are able to focus almost immediately on the linguistic processing.

In addition to these more "superficial" aspects of TAGARELA, we also carry over the idea of using annotation-based processing (cf. Amaral and Meurers, 2007). Before any error detection or diagnosis is performed, the first step is to annotate the learner input with the linguistic properties which can be automatically determined. From this annotation and from information about, e.g., the activity, a separate error diagnosis module can determine the most likely error.

Unfortunately, the "annotator" (or the *analysis model*) cannot be carried over, as it is designed specifically for Portuguese, which differs greatly from Russian in terms of how it encodes relevant syntactic and morphological information. With an annotation-based framework, the focus for processing Russian is to determine which information can provide the linguistic properties relevant to detecting and diagnosing ill-formed input and thus which NLP tools will provide analyses (full or partial) which have a bearing on detecting the errors of interest.

3 Exercise design

A perennial question for ICALL systems in general is what types of errors are learners allowed to make? This is crucially dependent upon the design of the activities. We want the processing of our system to be general, but we also take as a priority making the system usable, and so any analysis done in an annotation-based framework must be relevant for what learners are asked to do.

The goal of our system is to cover a range of exercises for students enrolled in an eight-week "survival" Russian course. These students start the course knowing nothing about Russian and finish it comfortable enough to travel to Russia. The exercises must therefore support the basics of grammar, but also be contextualized with situations that a student might encounter. To aid in contextualization, we plan to incorporate both audio and video, in order to provide additional "real-life" listening (and observing) practice outside of the classroom.

The exercises we plan to design include: listening exercises, video-based narrative exercises, reading practice, exercises centered around maps and locations, as well as more standard fill-in-the-blank (FIB) exercises. These exercises allow for variability in difficulty and in learner input.

From the processing point of view, each will have

its own hurdles, but all require some morphosyntactic analysis of Russian. To constrain the input for development and testing purposes, we are starting with an FIB exercise covering verbal morphology. Although this is not the ideal type of exercise for displaying the full range of ICALL benefits and capabilities, it is indispensible from a pedagogical point of view (given the high importance of rapid recognition of verbal forms in a morphologically rich language like Russian) and allows for rapid development, testing, and perfection of the crucial morphological analysis component, as it deals with complicated morphological processing in a suitably constrained environment. The successes and pitfalls of this implementation are unlikely to differ radically for morphological processing in other types of exercises; the techniques developed for this exercise thus form the basis of a reusable framework for the project as a whole.

A simple example of a Russian verbal exercise is in (1), where the verb needs to be past tense and agree with third person singular masculine noun.

(1) Вчера   он __ (видеть) фильм.
     Yesterday he __ (to see) a film

4   Taxonomy for morphological errors

When considering the integration of NLP tools for morphological error detection, we need to consider the nature of learner language. In this context, an analyzer cannot simply reject unrecognized or ungrammatical strings, as does a typical spell-checker, for example, but must additionally recognize what was intended and provide meaningful feedback on that basis. Formulating an error taxonomy delineates what information from learner input must be present in the linguistic analysis.

Our taxonomy is given in figure 1. As can be seen at a glance, the errors become more complex and require more information about the complete syntax as we progress in the taxonomy.

To begin with, we have inappropriate verb stems. For closed-form exercises, the only way that a properly-spelled verb stem can be deemed appropriate or inappropriate is by comparing it to the verb that the student was asked to use. Thus, errors of type #1b are straightforward to detect and to provide feedback on; all that needs to be consulted is

1. Inappropriate verb stem

   (a) Always inappropriate
   (b) Inappropriate for this context

2. Inappropriate verb affix

   (a) Always inappropriate
   (b) Always inappropriate for verbs
   (c) Inappropriate for this verb

3. Inappropriate combination of stem and affix

4. Well-formed word in inappropriate context

   (a) Inappropriate agreement features
   (b) Inappropriate verb form (tense, perfective/imperfective, etc.)

Figure 1: Error taxonomy for Russian verbal morphology

the activity model.[1] Errors of type #1a (and #2a) are essentially misspellings and will thus require spell-checking technology, which we do not focus on in this paper, although we discuss it briefly in section 5.3.

Secondly, there are inappropriate verb affixes, which are largely suffixes in Russian. Other than misspellings (#2a), there are two ways that affixes can be incorrect, as shown in example (2). In example (2a), we have the root for 'begin' (pronounced *nachina*) followed by an ending (*ev*) which is never an appropriate ending for any Russian verb, although it is a legitimate nominal suffix (#2b). The other subtype of error (#2c) involves affixes which are appropriate for different stems within the same POS category. In example (2b), a third person singular verb ending was used (*it*), but it is appropriate for a different conjugation class. The appropriate form for 'he/she/it begins' is начинает.

(2) a. *начина-ев
         begin-??

     b. *начина-ит
         begin-3s

The third type of error is where the stem and affix

---

[1]Note that if one were allowing free input, this error type could be the most difficult, in that the semantics of the sentence would have to be known to determine if a verb was appropriate.

may both be correct, but they were put together inappropriately. In a sense, these are a specific type of misspelling. For example, the infinitive мочь (*moch*, 'to be able to') can be realized with different stems, depending upon the ending, i.e., мог-у (*mogu* 'I can') мож-ем (*mozhem* 'we can'). Thus, we might expect to see errors such as *мож-у (*mozhu*), where both the stem and the affix are appropriate—and appropriate for this verb—but are not combined in a legitimate fashion. The technology needed to detect these types of errors is no more than what is needed for error type #2, as we discuss in section 5.

The final type of error is the one which requires the most attention in terms of NLP processing. This is the situation when we have a well-formed word appearing in an inappropriate context. In other words, there is a mismatch between the morphological properties of the verb and the morphological properties dictated by the context for that verb.

There are of course different ways in which a verb might display incorrect morphological features. In the first case (#4a), there are inappropriate agreement features. Verbs in Russian agree with the properties of their subject, as shown in example (3). Thus, as before, we need to know the morphological properties of the verb, but now we need not just the possible analyses, but the best analysis in this context. Furthermore, we need to know what the morphological properties of the subject noun are, to be able to check whether they agree. Access to the subject is something which can generally be determined by short context, especially in relatively short sentences.

(3) a. Я думаю
        I think-1sg

    b. Он думает
        He think-3sg

    c. *Я думает
        I think-3sg

In the second case (#4b), the verb could be in an inappropriate form: the tense could be inappropriate; the verbal form (gerund, infinitive, etc.) could be inappropriate; the distinction between perfective and imperfective verbs could be mistakenly realized; and so forth. Generally speaking, this kind of contextual information comes from two sources: 1) The

activity model can tell us, for example, whether a perfective (generally, a completed action) or an imperfective verb is required. 2) The surrounding sentence context can tell us, for example, whether an infinitive verb is governed by a verb selecting for an infinitive. Thus, we need the same tools that we need for agreement error detection.

By breaking it down into this taxonomy, we can more clearly delineate when we need external technology in dealing with morphological variation. For error types #1 through #3, we make no use of context and only need information from an activity model and a lexicon to tell us whether the word is valid. For these error types, the processing can proceed in a relatively straightforward fashion, provided that we have a lexicon, as outlined in section 5. Note also that our error taxonomy is meant to range over the space of logically possible error types for learners from any language background of any language's morphological system. In this way, it differs from the more heuristic approaches of earlier systems such as Athena (Murray, 1995), which used taxonomies tailored to the native languages of the system's users.

That leaves category #4. These errors are morphological in nature, but the words are well-formed, and the errors have to do with properties conditioned by the surrounding context. These are the kind for which we need external technology, and we sketch a proposed method of analysis in section 5.4.

Finally, we might have considered adding a fifth type of error, as in the following:

5. Well-formed word appropriate to the sentence, used inappropriately

    (a) Inappropriate position
    (b) Inappropriate argument structure

However, these issues of argument structure and of pragmatically-conditioned word order variation do not result in morphological errors of the verb, but rather clearly syntactic errors. We are currently only interested in morphological errors, given that in certain exercises, as in the present cases, syntactic errors are not even possible. With an FIB design, even though we might still generate a complete analysis of the sentence, we know which word has

the potential for error. Even though we are not currently concerned with these types of errors, we can note that argument structure errors can likely be handled through the activity model and through a similar analysis to what described is in section 5.4 since both context-dependent morphological errors (e.g., agreement errors) and argument structure errors rely on relations between the verb and its arguments.

5 Linguistic analysis

Given the discussion of the previous section, we are now in a position to discuss how to perform morphological analysis in a way which supports error diagnosis.

5.1 The nature of the lexicon

In much syntactic theory, sentences are built from feature-rich lexical items, and grammatical sentences are those in which the features of component items agree in well-defined ways. In morphologically-rich languages like Russian, the heavy lifting of feature expression is done by overt marking of words in the form of affixes (mainly prefixes and suffixes in the case of Russian). To be able to analyze words with morphological errors, then, we need at least partially successful morphological analysis of the word under analysis (as well as the words in the context).

The representation of words, therefore, must be such that we can readily obtain accurate partial information from both well-formed and ill-formed input. A relatively straightforward approach for analysis is to structure a lexicon such that we can build up partial (and competing) analyses of a word as the word is processed. As more of the word is (incrementally) processed, these analyses can be updated. But how is this to be done exactly?

In our system, we plan to meet these criteria by using a fully-specified lexicon, implemented as a Finite State Automaton (FSA) and indexed by both word edges. Russian morphological information is almost exclusively at word edges—i.e., is encoded in the prefixes and suffixes—and thus an analysis can proceed by working inwards, one character at a time, beginning at each end of an input item.[2]

By *fully-specified*, we mean that each possible form of a word is stored as a separate entity (path). This is not as wasteful of memory as it may sound. Since the lexicon is an FSA, sections shared across forms need be stored only once with diversion represented by different paths from the point where the shared segment ends. In fact, representing the lexicon as an FSA ensures that this process efficiently encodes the word possibilities. Using an FSA over all stored items, regular affixes need to be stored only once, and stems which require such affixes simply point to them (Clemenceau, 1997). This gives the analyzer the added advantage that it retains explicit knowledge of state, making it easy to simultaneously entertain competing analyses of a given input string (Ćavar, 2008), as well as to return to previous points in an analysis to resolve ambiguities (cf., e.g., Beesley and Karttunen, 2003).

We also need to represent hypothesized morpheme boundaries within a word, allowing us to segment the word into its likely component parts and to analyze each part independently of the others. Such segmentation is crucial for obtaining accurate information from each morpheme, i.e., being able to ignore an erroneous morpheme while identifying an adjoining correct morpheme. Note also that because an FSA encodes competing hypotheses, multiple segmentations can be easily maintained.

Consider example (4), for instance, for which the correct analysis is the first person singular form of the verb *think*. This only becomes clear at the point where segmentation has been marked. Up to that point, the word is identical to some form of ду-ма (*duma*), 'parliament' (alternatively, 'thought'). Once the system has seen дума, it automatically entertains the competing hypotheses that the learner intends 'parliament,' or any one of many forms of 'to think,' as these are all legal continuations of what it has seen so far. Any transition to ю after дума carries with it the analysis that there is a morpheme boundary here.

(4) дума|ю
     think-1sg

Obviously this bears non-trivial resemblance to spell-checking technology. The crucial difference

---

comes in the fact that an ICALL morphological analyzer must be prepared to do more than simply reject strings not found in the lexicon and thus must be augmented with additional, morphological information. Transitions in the lexicon FSA will need to encode more information than just the next character in the input; they also need to be marked with possible morphological analyses at points where it is possible that a morpheme boundary begins.

Maintaining hypothesized paths through a lexicon based on erroneous input must obviously be constrained in some way (to prevent all possible paths from being simultaneously entertained), and thus we first developed the error taxonomy above. Knowing what kinds of errors are possible is crucial to keeping the whole process workable.

5.2 FSAs for error detection

But why not use an off-the-shelf morphological analyzer which returns all possible analyses, or a more traditional paradigm-based lexicon? There are a number of reasons we prefer exploring an FSA implementation to many other approaches to lexical storage for the task of supporting error detection and diagnosis.

First, traditional mophological analyzers generally assume well-formed input. And, unless they segment a word, they do not seem to be well-suited to providing information relevant to context-independent errors.

Secondly, we need to readily have access to alternative analyses, even for a legitimate word. With phonetically similar forms used as different affixes, learners can accidentally produce correct forms, and thus multiple analyses are crucial. For example, -y can be either a first person singular marker for certain verb classes or an accusative marker for certain noun classes. Suppose a learner attempts to make a verb out of the noun душ (*dush*), meaning 'shower' and thus forms the word душу. It so happens that this incorrect form is identical to an actual Russian word: the accusative form of the noun 'soul.' A more traditional morphological analysis will likely only find the attested form. Keeping track of the history from left-to-right records that the 'shower' reading is possible; keeping track of the history from right-to-left records that a verbal ending is possible. Compactly representing such ambiguity—especially

when the ambiguity is not in the language itself but in the learner's impression of how the language works—is thus key to identifying errors.

Finally, and perhaps most importantly, morphological analysis over a FSA lexicon allows for easy implementation of activity-specific heuristics. In the current example, for instance, an activity might prioritize a 'shower' reading over a 'soul' one. Since entertained hypotheses are all those which represent legal continuations (or slight alterations of legal continuations) through the lexicon from a given state in the FSA, it is easy to bias the analyzer to return certain analyses through the use of weighted paths. Alternatively, paths that we have strong reason to believe will not be needed can be "disconnected." In the verbal morphology exercise, for example, suffix paths for non-verbs can safely be ignored.

The crucial point about error detection in ICALL morphological analysis is that the system must be able to speculate, in some broadly-defined sense, on what learners *might have meant* by their input, rather than simply evaluating the input as correct or incorrect based on its (non)occurrence in a lexicon. For this reason, we prefer to have a system where at least one component of the analyzer has 100% recall, i.e., returns a set of all plausible analyses, one of which can reasonbly be expected to be correct. Since an analyzer based on an FSA lexicon has full access to the lexicon at all stages of analysis, it efficiently meets this requirement, and it does this without anticipating specific errors or being tailored to a specific type of learner (cf., e.g., Felshin, 1995).

5.3 Error detection

Having established that an FSA lexicon supports error detection, let us outline how it will work. Analysis is a process of attempting to form independent paths through the lexicon - one operating "forward" and the other operating "backward." For grammatical input, there is generally one unique path through the lexicon that joins both ends of the word. Morphological analysis is found by reading information from the transitions along the chain (cf. Beesley and Karttunen, 2003). For ungrammatical input, the analyzer works by trying to build a connecting path based on the information it has.

Consider the case of the two ungrammatical verbs in (5).

(5) a. *начина-ев
        begin-??

    b. *начина-ит
        begin-3s

In (5a) (error type #2b) the analysis proceeding from the end of the word would fail to detect that the word is intended to be a verb. But it would, at the point of reaching the е in ев, recognize that it had found a legitimate nominal suffix. The processing from the beginning of the word, however, would recognize that it has seen some form of *begin*. We thus have enough information to know what the verbal stem is and that there is probably a morpheme boundary after начина-. These two hypotheses do not match up to form a legitimate word (thereby detecting an error), but they provide crucial partial information to tell us how the word was misformed.

Detecting the error in (5b) (type #2c) works similarly, and the diagnosis will be even easier. Again, analyses proceeding from each end of the word will agree on the location of the morpheme boundary and that the type of suffix used (third person singular) is a type appropriate to verbs, just not for this conjugation class. Having a higher-level rule recognize that all features match, merely the form is wrong, is easily achieved in a system with an explicit taxonomy of expected error types coded in.

Errors of type #3 are handled in exactly the same fashion: information about which stem or which affix is used is readily available, even if there is no complete path to form a whole word.

Spelling errors within a stem or an affix (error types #1a and #2a) require additional technology in order to find the intended analysis—which we only sketch here—but it is clear that such spell-checking should be done separately on each morpheme.[3] In the above examples, if the stem had been misspelled, that should not change the analysis of the suffix. Integrating spell-checking by calculating edit distances between a realized string and a morpheme in the lexicon should be relatively straightforward, as that technology is well-understood (see, e.g., Mitton, 1996) and since we are already analyzing subparts of words.

Obviously, in many cases there will be lingering ambiguity, either because there are multiple grammatical analyses in the lexicon for a given input form, or because the learner has entered an ungrammatical form, the intention behind which cannot entirely be determined from the input string alone. It is for such cases that the morphological analyzer we propose is most useful. Instead of returning the most likely path through the analyzer (e.g., the GPARS system of Loritz, 1992), our system proposes to follow *all plausible* paths through the lexicon *simultaneously*—including those that are the result of string edit "repair" operations.[4] In short, we intend a system that entertains competing hypotheses "online" as it processes input words.[5]

This results in a set of analyses, providing sentence-level syntactic and semantic analysis modules quick access to competing hypotheses, from which the the analysis most suitable to the context can be chosen, including those which are misspelled. The importance of this kind of functionality is especially well demonstrated in Pijls et al. (1987), which points out that in some languages—Dutch, in this case—minor, phonologically vacuous spelling differences are syntactically conditioned, making spell checking and syntactic analysis mutually dependent. Such cases are rarer in Russian, but the functionality remains useful due to the considerable interdependence of morphological and syntactic analysis.

5.4 Morphological analysis in context

For the purposes of the FIB exercise currently under development, the finite-state morphological analyzer we are building will of course be sufficient, but as exercises grow in complexity, it will be necessary to use it in conjunction with other tools. It is worth briefly sketching how the components of this integrated system will work together to provide useful error feedback to our learners.

If the learner has formed a legitimate word, the task becomes one of determining whether or not it

---

[3]Clearly, we will be able to determine whether a word is correctly spelled or not; the additional technology is needed to determine the candidate corrections.

[4]These include transitions to states on no input symbol (INSERTION), transitions to states on a different symbol from the next input symbol (SUBSTITUTION), and consumption of an input symbol without transition to a new state (DELETION).

[5]It is worth noting here that GPARS was actually a sentence-level system; it is for the word-level morphological analysis discussed here that we expect the most gain from our approach.

is appropriate to the context. The FSA analyzer will provide a list of possible analyses (i.e., augmented POS tags) for each input item (ranked, if need be). We can explore using a third-party tagger to narrow down this output list to analyses that make sense in context. We are considering both the Hidden Markov Model tagger TnT (Brants, 2000) and the Decision Tree Tagger (Schmid, 1997), with parameter files from Sharoff et al. (2008). Both of these taggers use local context, but, as they provide potentially different types of information, the final system may use both in parallel, weighing the output of each to the degree which each proves useful in trial runs to make its decision.

Since POS tagging does not capture every syntactic property that we might need access to, we are not sure how accurate error detection can be. Thus, to supplement its contextual information, we intend to use shallow syntactic processing methods, perhaps based on a small set of constraint grammar rules (cf, e.g., Bick, 2004). This shallow syntactic recognizer can operate over the string of now-annotated tags to resolve any remaining ambiguities and point out any mismatches between the items (for example, a noun-adjective pair where the gender does not match), thereby more accurately determining the relations between words.

6 Summary and Outlook

We have outlined a system for Russian ICALL exercises, the first of its kind for a Slavic language, and we have specifically delineated the types of errors to which need to be analyzed for such a morphologically-rich language. In that process, we have proposed a method for analyzing the morphology of learner language and noted where external NLP tools will be useful, making it clear how all these tools can be optimized for learning environments where the priority is to obtain a correct analysis, over obtaining any analysis.

The initial challenge is in creating the FSA lexicon, given that no such resource exists. However, unsupervised approaches to calculating the morphology of a language exist, and these can be directly connected to FSAs (Goldsmith and Hu, 2004). Thus, by using a tool such as Linguistica[6] on a cor-

pus such as the freely available subset of the Russian Internet Corpus (Sharoff et al., 2008),[7] we can semi-automatically construct an FSA lexicon, pruning it by hand.

Once the lexicon is constructed—for even a small subset of the language covering a few exercises—the crucial steps will be in performing error detection and error diagnosis on top of the linguistic analysis. In our case, linguistic analysis is provided by separate (levels of) modules operating in parallel, and error detection is largely a function of either noticing where these modules disagree, or in recognizing cases where ambiguity remains after one has been used to constrain the output of the other.

We have also tried to advance the case that this and future ICALL systems do better to build on existing technologies, rather than building from the bottom up for each new language. We hope that the approach we are taking to morphological analysis will prove to be just such a general, scalable system, one applicable—with some tweaking and to various levels—to morphologically-rich languages and isolating languages alike.

References

Amaral, Luiz (2007). Designing Intelligent Language Tutoring Systems: integrating Natural Language Processing technology into foreign language teaching. Ph.D. thesis, The Ohio State University.

Amaral, Luiz and Detmar Meurers (2006). Where does ICALL Fit into Foreign Language Teaching? Talk given at CALICO Conference. University of Hawaii, `http:`

---

[6]`http://linguistica.uchicago.edu/`

[7]`http://corpus.leeds.ac.uk/mocky/`

//purl.org/net/icall/handouts/
calico06-amaral-meurers.pdf.

Amaral, Luiz and Detmar Meurers (2007). Putting activity models in the driver's seat: Towards a demand-driven NLP architecture for ICALL. Talk given at EUROCALL. University of Ulster, Coleraine Campus, http://purl.org/net/icall/handouts/eurocall07-amaral-meurers.pdf.

Beesley, Kenneth R. and Lauri Karttunen (2003). *Finite State Morphology*. CSLI Publications.

Bick, Eckhard (2004). PaNoLa: Integrating Constraint Grammar and CALL. In Henrik Holmboe (ed.), *Nordic Language Technology*, Copenhaguen: Museum Tusculanum, pp. 183–190.

Brants, Thorsten (2000). TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*. Seattle, WA, pp. 224–231.

Ćavar, Damir (2008). The Croatian Language Repository: Quantitative and Qualitative Resources for Linguistic Research and Language Technologies. Invited talk, Indiana University Department of Lingistics, January 2008.

Clemenceau, David (1997). Finite-State Morphology: Inflections and Derivations in a Singl e Framework Using Dictionaries and Rules. In Emmanuel Roche and Yves Schabes (eds.), *Finite State Language Processing*, The MIT Press.

Felshin, Sue (1995). The Athena Language Learning Project NLP System: A Multilingual System for Conversation-Based Language Learning. In *Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, chap. 14, pp. 257–272.

Goldsmith, John and Yu Hu (2004). From Signatures to Finite State Automata. In *Midwest Computational Linguistics Colloquium (MCLC-04)*. Bloomington, IN.

Heift, Trude and Devlan Nicholson (2001). Web delivery of adaptive and interactive language tutoring. *International Journal of Artificial Intelligence in Education* 12(4), 310–325.

Loritz, D. (1992). Generalized Transition Network Parsing for Language Study: the GPARS system

for English, Russian, Japanese and Chinese. *CALICO Journal* 10(1).

Mitton, Roger (1996). *English Spelling and the Computer*. Longman.

Murray, Janet H. (1995). Lessons Learned from the Athena Language Learning Project: Using Natural Language Processing, Graphics, Speech Processing, and Interactive Video for Communication-Based Language Learning. In V. Melissa Holland, Michelle R. Sams and Jonathan D. Kaplan (eds.), *Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, chap. 13, pp. 243–256.

Nagata, Noriko (1995). An Effective Application of Natural Language Processing in Second Language Instruction. *CALICO Journal* 13(1), 47–67.

Pijls, Fieny, Walter Daelemans and Gerard Kempen (1987). Artificial intelligence tools for grammar and spelling instruction. *Instructional Science* 16, 319–336.

Roark, Brian and Richard Sproat (2007). *Computational Approaches to Morphology and Syntax*. Oxford University Press.

Schmid, Helmut (1997). Probabilistic part-of-speech tagging using decision trees. In D.H. Jones and H.L. Somers (eds.), *New Methods in Language Processing*, London: UCL Press, pp. 154–164.

Sharoff, Serge, Mikhail Kopotev, Tomaž Erjavec, Anna Feldman and Dagmar Divjak (2008). Designing and evaluating Russian tagsets. In *Proceedings of LREC 2008*. Marrakech.

Vandeventer Faltin, Anne (2003). Syntactic error diagnosis in the context of computer assisted language learning. Thèse de doctorat, Université de Genève, Genève.