Russian Morphological Processing for ICALL

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

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

Abstract

We outline a method for processing Russian morphology, such that it can detect erroneous words and provide relevant grammatical information. The method is based upon finite-state morphological analysis and allows us to entertain alternative, competing hypotheses about a word's analysis, so that a particular ICALL context can select the best choice.

1 Introduction and Motivation

Intelligent computer-aided language learning (ICALL) systems are ideal for language pedagogy: by providing additional practice outside the classroom, they aid learners in the development of awareness of language forms and rules (see, e.g., Amaral and Meurers, 2006, and references therein). But there are very few ICALL systems in existence today: to the best of our knowledge, the only full systems are for German (Heift and Nicholson, 2001), Portuguese (Amaral and Meurers, 2006, 2007), and Japanese (Nagata, 1995). General techniques have been developed to detect ill-formed learner text (see, e.g., Vandeventer Faltin, 2003, and references therein), but these too have generally focused on a limited set of languages and language types (such as English and French). For processing of (error-filled) learner language to be general, it must account for a variety of language families, including those with more complex morphology than the Romance and Germanic languages.

Before detecting ill-formed structures in learner input, one must have an appropriate representation supporting learner language. That is, we must be able to analyze learner input in a way which can model learner variation and support useful feedback to the learner. Recent methods for processing input have focused on so-called *annotation-based processing* (see Amaral and Meurers, 2007). The idea is that before any error detection or diagnosis is performed, the learner input is annotated with the (potentially ambiguous) linguistic properties which can be automatically determined, feeding the results to a separate error diagnosis module. The first question in analyzing learner language, then, is to determine which linguistic properties are relevant.

For these reasons, we focus our attention on a data representation for Russian words which is appropriate for ICALL systems that must detect ill-formed input. To handle the relatively rich morphology of Russian, we adopt an approach grounded in Finite State Morphology—in the tradition of (Beesley and Karttunen, 2003) and (Clemenceau, 1997)—and constrain the processing by an explicit taxonomy of expected error types. We believe that this approach is easily generalizable to new languages.

We describe the general ICALL context in section 2 before turning to the appropriate linguistic analysis in section 3. In section 3.1, we outline an appropriate data representation, namely a finite-state lexicon, and turn to using this for error detection in section 3.2. We give arguments for the benefit of this representation in section 3.3, and sketch a way that such a lexicon can be semi-automatically constructed in section 4.

2 Background

Designing a morphological processor for Russian learner language depends upon the types of errors which learners will make, which in turn depends upon the context in which they produce language. We outline the exercise content and expected error types here; see Dickinson and Herring (2008) for more details.

2.1 Exercise context

The goal of the system is to cover a range of exercises for students enrolled in an eight-week "survival" Russian course. The exercises must therefore support the basics of grammar, but also be contextualized with situations that a student might encounter. From the processing point of view, each will have its own hurdles, but all require some morphosyntactic analysis of Russian.

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.
vchera	on	__	videt'	fil'm
Yesterday	he	__	(to see)	a film

2.2 Expected error types

When considering the integration of NLP tools for morphological error detection, it is important to consider the nature of learner language. An analyzer in a system designed to aid in the learning process 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. We are starting with verbal exercises, and thus our taxonomy for verbal morphological errors is given in figure 1. Note that these error types have direct corollaries in other parts of speech.

As can be seen, the errors become more complex and require more information about the complete syntax as we progress in the taxonomy. Briefly, error types #1a and #2a are essentially spelling errors. Error type #1b depends upon the exercise, and thus is of less interest for our purposes.

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

Error types #2b, #2c, and #3 are of the most interest for morphological processing, irrespective of context. For correctly-spelled affixes, there are two main ways that they 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 (*nachinaet*).

(2) a. *начина-ев
begin-??

b. *начина-ит
begin-3s

The third type of error is where the stem and affix 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 later.

The fourth error type is the situation when we have a well-formed word appearing in an inappropriate context. In this paper, we do not focus on context and do not discuss these types at length.

3 Morphological Analysis

The relatively free word order of Russian requires that morphological analysis be involved in the annotation process from the earliest stages. As vital clues to syntactic structure are as likely to be found in the morphology as in the word order, it is important that any annotation-based analysis scheme for this language be able to determine the functional features of a lexical item, to the extent that this is possible, independent of its surrounding context. As noted, ICALL systems are responsible for giving feedback to learners on error-laden input, saddling the analyzer with the added burden that it cannot simply reject ungrammatical strings, but must rather form and maintain several competeing hypotheses about the learner's intentions throughout the analysis process, choosing from among them as a basis for feedback generation.

Finite State Morphology is ideal for this purpose. For one thing, it allows us to use a *fully-specified* lexicon, in the sense that all possible forms of all words are explicitly encoded rather than derived by rules. This has the advantage of reducing the amount of guesswork that has to be done. Rather than applying and reapplying rules to error-laden input to try to come up with a form it recognizes, the analyzer can instead perform simple operations over the stored forms themselves. For another thing, competing hypotheses can be represented compactly as alternate paths over pre-stored strings. Error correction, in some sense, reduces to backtracking. Finally, morphological analysis over a finite-state lexicon allows for easy implementation of activity-specific heuristics. By making use of weighted arcs, lesson designers can easily bias analysis toward forms that are appropriate for the material of the operative lesson without restricting the analyzer's knowledge to such a degree that it is unable to handle unexpected input.

3.1 The nature of the lexicon

To be able to analyze words with morphological errors, the representation of words 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.

In our system, we plan to meet these criteria by using a fully-specified lexicon, implemented as a Finite State Transducer (FST) 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.[1] This provides an efficent way to form parallel hypotheses about word class starting from the sections of the word most likely to be relevant.

Morphological analysis itself will work much as in (Clemenceau, 1997), with morphological endings stored as separate chains and attached to the main chain as appropriate, leveraging the transducer when necessary to account for fused and derived forms (see also Koskenniemi, 1983). The process reads symbols from the input string one at a time, building a set of hypotheses about the proper analysis as it goes. The set of hypotheses at each stage is a set of legal continuations of the current string, plus a set of continuations that can be obtained through the application of one of a set of *repair operations*, as described below. As the analyzer reads in a new symbol from the input string, it transitions to a new state in the lexicon over an arc labeled with that symbol (or else an arc labeled with another symbol rendered legal by one of the repair operations).

As it changes state, the transducer adds information to the current set of analyses. This added information usually involves simply appending the input

[1]See Roark and Sproat (2007) for a general overview of implementational strategies for finite-state morphological analysis.

symbol to the output, i.e., appropriate members of the analysis set. In some cases, where a transition represents the crossing of a morphological boundary, it may additionally add morphological features from a specified set. In other cases, where phonological processes are at work that may have been misapplied by the learner, it may return corrections on the input string as dictated by the process in question.

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 FST, sections shared across forms need be stored only once with diversion represented by different paths from the point where the shared segment ends. This allows regular affixes to be stored as entries in their own right, and stems which require such affixes simply point to them (Clemenceau, 1997). An analyzer operating over an FST lexicon comes with 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).

As mentioned, we 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 FST encodes competing hypotheses, multiple segmentations can be easily maintained.

Consider example (3), 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 дума (*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.

(3) 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 FST 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). It is in this capacity that the error taxonomy described in the previous section finds its most important use. Knowing what kinds of errors are possible is crucial to keeping the whole process workable.

3.2 Error detection

Correctly-spelled morphemes Having established that an FST 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 (4).

(4) a. *начина-ев
begin-??

b. *начина-ит
begin-3s

In (4a) (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 (4b) (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.

Misspelled morphemes Spelling errors within a stem or an affix (error types #1a and #2a) require additional technology in order to find the intended analysis, where it is clear that such spell-checking should be done separately on each morpheme.[2] 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. We outline this process here.

Erroneous input can be handled through simple "repair" operations on chains—including, but not limited to: INSERTION (changing state without consuming an input symbol), DELETION (consuming an input symbol without changing state), and SUBSTITUTION (consuming an input symbol and changing to a state not determined by that symbol). The final operation - SUBSTITUTION - sees the most use when, for example, a learner has correctly added an ending of a certain type to a stem but used the wrong *actual* ending, perhaps putting an ending on a verb that expresses the correct agreement features but is appropriate to verbs of a different class.

By way of illustration, consider the example from (4a), *начина-ев, in more detail. In the first step, the analyzer reading from the left will encounter н (*n*), and the one reading from the right в (*v*). Technically, this means that all words beginning in н and ending in в, including unrelated forms like нарыв (*naryv*, 'abcess, boil'), are in the set of hypotheses.

However, they are not entertained explicitly, at this point, but rather implicitly as members of the set of legal continuations. What is explicitly stored are the characters н and в and any morphological information attached to the transitions that were crossed to output these characters. In the case of в, this will include the features `+noun`, `+masculine` and `+singular` for the path that ends the word нарыв. As further input symbols are read in, the set of entertained hypotheses narrows. On reading in the е from the righthand side, the analyzer will need a repair operation – SUBSTITUTION – to keep the нарыв hypothesis alive. Since substitution for one character is not too costly, it will, in fact, do this – maintaining (5) on its list of hypotheses, but with an associated cost for having applied the repair operation. This cost will become prohibitive over the next few input symbols, and the path through the lexicon corresponding to the hypothesis that the input word is нарыв will soon be dropped.

(5) ыв: `+noun`, `+masculine`, `+singular`

What the system will in fact end up with is a set of hypotheses that proceed from начина (*nachina*), ending in a morpheme boundary. At this point, it will only be possible to apply repair operations to make anything work, and since the two analyzers reading from either end agree on the position of the morpheme boundary after начина, repair operations that simply read in legal affix-based continuations from that point will be less costly than any hypothesis that attempts to make the начина work without a boundary (e.g., начинание, *nachinanie*, 'undertaking'). Any analysis that tries to make the ев (*ev*) ending work without reference to the начина part, such as the aforementioned нарыв, will have been

[2]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.

dropped long before this point as entirely too costly.[3] The analyzer will return a list of hypotheses – not only the "best" ones, but anything that falls within the determined limits of "acceptable cost" – which are then passed on to the next, sentence-level, component for use in further processing. Error feedback is then based on the mismatch between the form that the later module chooses from this list and the form that the learner input originally.

Regular spelling errors are handled similarly. If, for example, a learner gets the form basically right but mistakenly leaves out a letter—perhaps typing in нчинает by mistake—it is a simple matter of applying the repair operation INSERTION to come up with the correct analysis. In such cases, INSERTION is applied when the analyzer finds itself in a state with no legal continuations on the next input symbol. It then simply follows all the exit transitions looking for a subsequent state that contains an exit arc annotated with the next input symbol and INSERTs the symbol required by the transition that takes it there.

DELETION works analogously. When the analyzer finds itself in a state with no exit arcs labeled with the next input symbol, but one labeled with the following input symbol, it may simply pay a cost to skip the next input symbol, following the transition over the one that comes after it.

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 outlined above. In short, we intend a system that entertains competing hypotheses "online" as it processes input words.[4]

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.

3.3 Alternative analyses

In considering a fully-specified lexicon, we could instead explore the use of 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 FST 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, -у 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'

[3]What counts as "too costly" is obviously a heuristic matter that will have to be fine-tuned according to the needs of the system in the final stages of development.

[4]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.

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 an FST 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 FST, 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 a 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 reasonably be expected to be correct. Since an analyzer based on an FST 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).

4 Constructing the Lexicon

Because it remains so tightly connected to actually occurring words, the construction of such a lexicon employing FST chains can be done semi-automatically. Using a freely available corpus, one can use a handful of inflected forms to derive common morphological paradigms. That is, subparts of words (for Russian, generally the beginning or end) will be common to many morphological forms, thus distinguishing themselves as relevant affixes. From these commonalities, one can automatically derive a lexicon with a potentially small amount of manual checking.

The initial challenge is in creating the FST lexicon, given that no such resource exists, as far as we are aware. However, unsupervised approaches to calculating the morphology of a language exist, and these can be directly connected to FSTs (Goldsmith and Hu, 2004). Thus, by using a tool such as Linguistica[5] on a corpus such as the freely available subset of the Russian Internet Corpus (Sharoff et al., 2008),[6] we can semi-automatically construct an FST lexicon, pruning it by hand.

5 Summary and Outlook

We have proposed a method for analyzing the morphology of learner language in Russian, making it clear how this analysis can be optimized for learning environments where the priority is to obtain a correct analysis, over obtaining any analysis.

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 beyond the word level will be provided by separate 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.

Acknowledgments We would like to thank Detmar Meurers and Luiz Amaral for providing us with the TAGARELA sourcecode, as well as for valuable insights into the workings of ICALL systems; and to thank Anna Feldman and Jirka Hana for advice on Russian resources. We also thank two anonymous reviewers for insightful comments that have influenced the final version of this paper. This research was supported by grant P116S070001 through the U.S. Department of Education's Fund for the Improvement of Postsecondary Education.

References

Amaral, Luiz and Detmar Meurers (2006). Where does ICALL Fit into Foreign Lan-

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

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

guage Teaching? Talk given at CALICO Conference. University of Hawaii, `http://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.

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.

Dickinson, Markus and Joshua Herring (2008). Developing Online ICALL Exercises for Russian. In *The 3rd Workshop on Innovative Use of NLP for Building Educational Applications (ACL08-NLP-Education)*. Columbus, OH.

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.

Koskenniemi, Kimmo (1983). Two-level morphology: a general computational model for word-fo rm recognition and production. Ph.D. thesis, University of Helsinki.

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.

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.

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.