Generating Learner-Like Morphological Errors in Russian

Markus Dickinson
Indiana University
md7@indiana.edu

Abstract

To speed up the process of categorizing learner errors and obtaining data for languages which lack error-annotated data, we describe a linguistically-informed method for generating learner-like morphological errors, focusing on Russian. We outline a procedure to select likely errors, relying on guiding stem and suffix combinations from a segmented lexicon to match particular error categories and relying on grammatical information from the original context.

1 Introduction

Work on detecting grammatical errors in the language of non-native speakers covers a range of errors, but it has largely focused on syntax in a small number of languages (e.g., Vandeventer Faltin, 2003; Tetreault and Chodorow, 2008). In more morphologically-rich languages, learners naturally make many errors in morphology (Dickinson and Herring, 2008). Yet for many languages, there is a major bottleneck in system development: there are not enough error-annotated learner corpora which can be mined to discover the nature of learner errors, let alone enough data to train or evaluate a system. Our perspective is that one can speed up the process of determining the nature of learner errors via semi-automatic means, by generating plausible errors.

We set out to generate linguistically-plausible morphological errors for Russian, a language with rich inflections. Generating learner-like errors has practical and theoretical benefits. First, there is the issue of obtaining training data; as Foster and Andersen (2009) state, "The ideal situation for a grammatical error detection system is one where a large amount of labelled positive *and* negative evidence is available." Generated errors can bridge this gap by creating realistic negative evidence (see also Rozovskaya and Roth, 2010). As for evaluation data, generated errors have at least one advantage over real errors, in that we know precisely what the correct form is supposed to be, a problem for real learner data (e.g., Boyd, 2010).

By starting with a coarse error taxonomy, generating errors can improve categorization. Generated errors provide data for an expert—e.g., a language teacher—to search through, expanding the taxonomy with new error types or subtypes and/or deprecating error types which are unlikely. Given the lack of real learner data, this has the potential to speed up error categorization and subsequent system development. Furthermore, error generation techniques can be re-used, adjusting the errors for different learner levels, first languages, and so forth.

The error generation process can benefit by using linguistic properties to mimic learner variations. This can lead to more realistic errors, a benefit for machine learning (Foster and Andersen, 2009), and can also provide feedback for the linguistic representation used to generate errors by, e.g., demonstrating under which linguistic conditions certain error types are generated and under which they are not.

We are specifically interested in generating Russian morphological errors. To do this, we need a knowledge base representing Russian morphology, allowing us to manipulate linguistic properties. After outlining the coarse error taxonomy (section 2), we discuss enriching a part-of-speech (POS) tagger lexicon with segmentation information (section 3). We then describe the steps in error generation (section 4), highlighting decisions which provide insight for the analysis of learner language, and show the impact on POS tagging in section 5.

2 Error taxonomy

Russian is an inflecting language with relatively free word order, meaning that morphological syntactic properties are often encoded by affixes. In (1a), for example, the verb начина needs a suffix to indicate person and number, and ет is the third person singular form.[1] By contrast, (1b) illustrates a paradigm error: the suffix ит is third singular, but not the correct one. Generating such a form requires having access to individual morphemes and their linguistic properties.

(1) a. начина+ет [*nachina+et*]
begin-3s

b. *начина+ит [*nachina+it*]
begin-3s (diff. verb paradigm)

This error is categorized as a suffix error in figure 1, expanding the taxonomy in Dickinson and Herring (2008). Stem errors are similarly categorized, with *Semantic errors* defined with respect to a particular context (e.g., using a different stem than required by an activity).

For formation errors (#3), one needs to know how stems relate. For instance, some verbs change their form depending on the suffix, as in (2). In (2c), the stem and suffix are morphologically compatible, just not a valid combination. One needs to know that мож is a variant of мог.

(2) a. мог+ут [*mog+ut*]
can-3p

b. мож+ет [*mozh+et*]
can-3s

c. *мож+ут [*mozh+ut*] (#3)
can-3p (wrong formation)

Using a basic lexicon without such knowledge, it is hard to tell formation errors apart from lexicon incompleteness (see section 4.2.2). If можут (2c) is generated and is not in the lexicon, we do not know whether it is misformed or simply unattested. In this paper, we group together such cases, since this allows for a simpler and more quickly-derivable lexicon.

0. Correct: The word is well-formed.
1. Stem errors:
   (a) Stem spelling error
   (b) Semantic error
2. Suffix errors:
   (a) Suffix spelling error
   (b) Lexicon error:
      i. Derivation error: The wrong POS is used (e.g., a noun as a verb).
      ii. Inherency error: The ending is for a different subclass (e.g., inanimate as an animate noun).
   (c) Paradigm error: The ending is from the wrong paradigm.
3. Formation errors: The stem does not follow appropriate spelling/sound change rules.
4. Syntactic errors: The form is correct, but used in an in appropriate syntactic context (e.g., nominative case in a dative context)

- Lexicon incompleteness: The form may be possible, but is not attested.

Figure 1: Error taxonomy

We have added syntactic errors, whereas Dickinson and Herring (2008) focused on strictly morphological errors. Learners make syntactic errors (e.g., Rubinstein, 1995; Rosengrant, 1987), and when creating errors, a well-formed word may result. In the future, syntactic errors can be subdivided (Boyd, 2010).

This classification is of *possible* errors, making no claim about the *actual* distribution of learner errors, and does not delve into issues such as errors stemming from first language interference (Rubinstein, 1995). Generating errors from the possible types allows one to investigate which types are plausible in which contexts.

[1] For examples, we write the Cyrillic form and include a Roman transliteration (SEV 1362-78) for ease of reading.

It should be noted that we focus on inflectional morphology in Russian, meaning that we focus on suffixes. Prefixes are rarely used in Russian as inflectional markers; for example, prefixes mark semantically-relevant properties for verbs of motion. The choice of prefix is thus related to the overall word choice, an issue discussed under *Random stem generation* in section 4.2.4.

3 Enriching a POS lexicon

To create errors, we need a segmented lexicon with morphological information, as in (3). Here, the word могу (*mogu*, 'I am able to') is split into stem and suffix, with corresponding POS tags.[2]

(3) a. мог,Vm-----a-p,у,Vmip1s-a-p
 b. мож,Vm-----a-p,ет,Vmip3s-a-p
 c. мог,Vm-----a-p,NULL,Vmis-sma-p

The freely-available POS lexicon from Sharoff et al. (2008), specifically the file for the POS tagger TnT (Brants, 2000), contains full words (239,889 unique forms), with frequency information. Working with such a rich database, we only need segmentation, providing a quickly-obtained lexicon (cf. five years for a German lexicon in Geyken and Hanneforth, 2005).

In the future, one could switch to a different tagset, such as that in Hana and Feldman (2010), which includes reflexivity, animacy, and aspect features. One could also expand the lexicon, by adapting algorithms for analyzing unknown words (e.g., Mikheev, 1997), as suggested by Feldman and Hana (2010). Still, our lexicon continues the trend of linking traditional categories used for tagging with deeper analyses (Sharoff et al., 2008; Hana and Feldman, 2010).[3]

3.1 Finding segments/morphemes

We use a set of hand-crafted rules to segment words into morphemes, of the form: if the tag is x and the word ends with y, make y the suffix. Such rules are easily and quickly derivable from a textbook listing of paradigms. For certain exceptional cases, we write word-specific rules. Additionally, we remove word, tag pairs indicating punctuation or non-words (`PUNC`, `SENT`, `-`).

One could use a sophisticated method for lemmatizing words (e.g., Chew et al., 2008; Schone and Jurafsky, 2001), but we would likely have to clean the lexicon later; as Feldman and Hana (2010) point out, it is difficult to automatically guess the entries for a word, without POS information. Essentially, we write precise rules to specify part of the Russian system of suffixes; the lexicon then provides the stems for free.

We use the lexicon for generating errors, but it should be compatible with analysis. Thus, we focus on suffixes for beginning and intermediate learners. We can easily prune or add to the rule set later. From an analysis perspective, we need to specify that certain grammatical properties are in a tag (see below), as an analyzer is to support the provision of feedback. Since the rules are freely available,[4] changing these criteria for other purposes is straightforward.

3.1.1 Segmentation rules

We have written 1112 general morphology rules and 59 rules for the numerals 'one' through 'four,' based on the *Nachalo* textbooks (Ervin et al., 1997). A rule is simply a tag, suffix pair. For example, in (4), `Ncmsay` (Noun, common, masculine, singular, accusative, animate [yes]) words should end in either а (*a*) or я (*ya*).

(4) a. Ncmsay, а
 b. Ncmsay, я

A program consults this list and segments a word appropriately, requiring at least one character in the stem. In the case where multiple suffixes match (e.g., ени (*eni*) and и (*i*) for singular neuter locative nouns), the longer one is chosen, as it is unambiguously correct.

We add information in 101 of the 1112 rules. All numerals, for instance, are tagged as `Mc-s` (Numeral, cardinal, [unspecified gender], singular). The tagset in theory includes properties such as case; they just were not marked (see footnote 6, though). Based on the ending, we add all

[2] POS tags are from the compositional tagset in Sharoff et al. (2008). A full description is at: `http://corpus.leeds.ac.uk/mocky/msd-ru.html`.

[3] This lexicon now includes lemma information, but each word is not segmented (Erjavec, 2010).

[4] `http://cl.indiana.edu/~boltundevelopment/`

possible analyses. Using an optional output tag, in (5), Mc-s could be genitive (g), locative (l), or dative (d) when it ends in и (*i*). These rules increase ambiguity, but are necessary for learner feedback.

(5) a. Mc-s, и, Mc-sg
b. Mc-s, и, Mc-sl
c. Mc-s, и, Mc-sd

In applying the rules, we generate stem tags, encoding properties constant across suffixes. Based on the word's tag (e.g., Ncmsay, cf. (4)) a stem is given a more basic tag (e.g., Ncm--y).

3.2 Lexicon statistics

To be flexible for future use, we have only enriched 90% of the words (248,014), removing every 10th word. Using the set of 1112 rules results in a lexicon with 190,450 analyses, where *analyses* are as in (3). For these 190,450 analyses, there are 117 suffix forms (e.g., я, *ya*) corresponding to 808 suffix analyses (e.g., <я, Ncmsay>). On average 3.6 suffix tags are observed with each stem-tag pair, but 22.2 tags are compatible, indicating incomplete paradigms.

4 Generating errors

4.1 Basic procedure

Taking the morpheme-based lexicon, we generate errors by randomly combining morphemes into full forms. Such randomness must be constrained, taking into account what types of errors are likely to occur.

The procedure is given in figure 2 and detailed in the following sections. First, we use the contextually-determined POS tag to restrict the space of possibilities. Secondly, given that random combinations of a stem and a suffix can result in many unlikely errors, we guide the combinations, using a loose notion of likelihood to ensure that the errors fall into a reasonable distribution. After examining the generated errors, one could restrict the errors even further. Thirdly, we compare the stem and suffix to determine the possible types of errors. A full form may have several different interpretations, and thus, lastly, we select the best interpretation(s).

1. Determine POS properties of the word to be generated (section 4.2.1).
2. Generate a full-form, via *guided* random stem and suffix combination (section 4.2.4).
3. Determine possible error analyses for the full form (section 4.2.2).
4. Select the error type(s) from among multiple possible interpretations (section 4.2.3).

Figure 2: Error generation procedure

By trying to determine the best error type in step 4, the generation process can provide insight into error analysis. This is important, given that suffixes are highly ambiguous; for example, ой (*-oj*) has at least 6 different uses for adjectives. Analysis is not simply generation in reverse, though. Importantly, error generation relies upon the context POS tag for the *intended* form, for the whole process. To morphologically analyze the corrupted data, one has to POS tag *corrupted* forms (see section 5).

4.2 Corruption

We use a corpus of 5 million words automatically tagged by TnT (Brants, 2000) and freely available online (Sharoff et al., 2008).[5] Because we want to make linguistically-informed corruptions, we corrupt only the words we have information for, identifying the words in the corpus which are found in the lexicon with the appropriate POS tag.[6] We also select only words which have inflectional morphology: nouns, verbs, adjectives, pronouns, and numerals.[7]

4.2.1 Determining word properties (step 1)

We use the POS tag to restrict the properties of a word, regardless of how exactly we corrupt it. Either the stem and its tag or the suffix and its tag

[5] See http://corpus.leeds.ac.uk/mocky/.

[6] We downloaded the TnT lexicon in 2008, but the corpus in 2009; although no versions are listed on the website, there are some discrepancies in the tags used (e.g., numeral tags now have more information). To accommodate, we use a looser match for determining whether a tag is known, namely checking whether the tags are compatible. In the future, one can tweak the rules to match the newer lexicon.

[7] Adverbs inflect for comparative forms, but we do not consider them here.

can be used as an invariant, to guide the generated form (section 4.2.4). In (6a), for instance, the adjective (`Af`) stem or plural instrumental suffix (`Afp-pif`) can be used as the basis for generation.

(6) a. Original: серыми (*serymi*, 'gray')
$\mapsto$ сер/Af+ыми/Afp-pif
b. Corrupted: сер+ой (*seroj*)

The error type is defined in terms of the original word's POS tag. For example, when we generate a correctly-formed word, as in (6b), it is a syntactic error if it does not match this POS tag.

4.2.2 Determining error types (step 3)

Before discussing word corruption in step 2 (section 4.2.4), we need to discuss how error types are determined (this section) and how to handle multiple possibilities (section 4.2.3), as these steps help guide step 2. After creating a corrupted word, we elucidate all possible interpretations in step 3 by comparing each suffix analysis with the stem. If the stem and suffix form a legitimate word (in the wrong context), it is a syntactic error. Incompatible features means a derivation or inherency error, depending upon which features are incompatible. If the features are compatible, but there is no attested form, it is either a paradigm error—if we know of a different suffix with the same grammatical features—or a formation/incompleteness issue, if not.

This is a crude morphological analyzer (cf. Dickinson and Herring, 2008), but bases its analyses on what is known about the invariant part of the original word. If we use ыми (*ymi*) from (6a) as an invariant, for instance, we know to treat it as a plural instrumental adjective ending, regardless of any other possible interpretations, because that is how it was used in this context.

4.2.3 Selecting the error type (step 4)

Corrupted forms may have many possible analyses. For example, in (6b), the suffix ой (*oj*) has been randomly attached to the stem сер (*ser*). With the stem fixed as an adjective, the suffix could be a feminine locative adjective (syntactic error), a masculine nominative adjective (paradigm error), or an instrumental feminine noun (derivation error). Given what learners are likely to do, we can use some heuristics to restrict the set of possible error types.

First, we hypothesize that a correctly-formed word is more likely a correct form than a misformed word. This means that correct words and syntactic errors—correctly-formed words in the wrong context—have priority over other error types. For (6b), for instance, the syntactic error outranks the paradigm and derivation errors.

Secondly, we hypothesize that a contextually-appropriate word, even if misformed, is more likely the correct interpretation than a contextually-inappropriate word. When we have cases where there is: a) a correctly-formed word not matching the context (a syntactic error), and b) a malformed word which matches the context (e.g., a paradigm error), we list both possibilities.

Finally, derivation errors seem less likely than the others (a point confirmed by native speakers), giving them lower priority. Given these heuristics, not only can we rule out error types after generating new forms, but we can also split the error generation process into different steps.

4.2.4 Corrupting selected words (step 2)

Using these heuristics, we take a known word and generate errors based on a series of choices. For each choice, we randomly generate a number between 0 and 1 and choose based on a given threshold. Thresholds should be reset when more is known about error frequency, and more decisions added as error subtypes are added.

Decision #1: Correct forms The first choice is whether to corrupt the word or not. Currently, the threshold is set at 0.5. If we corrupt the word, we continue on to the next decision.

Decision #2: Syntactic errors We can either generate a syntactic or a morphological error. On the assumption that syntactic errors are more common, we currently set a threshold of 0.7, generating syntactic errors 70% of the time and morphological form errors 30% of the time.

To generate a correct form used incorrectly, we extract the stem from the word and randomly select a new suffix. We keep selecting a suffix until we obtain a valid form.[8] An example is given in (7): the original (7a) is a plural instrumental adjective, unspecified for gender; in (7b), it is singular nominative feminine.

(7)	a.	серыми	глазами	.
		gray	eyes	.
		Afp-**pi**f	Ncmpin	SENT
	b.	серая	глазами	.
		Afp**fsn**f	Ncmpin	SENT

One might consider ensuring that each error differs from the original in only one property. Or one might want to co-vary errors, such that, in this case, the adjective and noun both change from instrumental to nominative. While this is easily accomplished algorithmically, we do not know whether learners obey these constraints. Generating errors in a relatively unbounded way can help pinpoint these types of constraints.

While the form in (7b) is unambiguous, syntactic errors can have more than one possible analysis. In (8), for instance, this word could be corrupted with an -ой (*-oj*) ending, indicating feminine singular genitive, instrumental, or locative. We include all possible forms.

(8)	серой	глазами	.
	Afpfsg.Afpfsi.Afpfsl	Ncmpin	SENT

Likewise, considering the heuristics in section 4.2.3, generating a syntactic error may lead to a form which may be contextually-appropriate. Consider (9): in (9a), the verb-preposition combination requires an accusative (Ncns**a**n). By changing -o to -e, we generate a form which could be locative case (Ncns**l**n, type #4) or, since -e can be an accusative marker, a misformed accusative with the incorrect paradigm (#2c). We list both possibilities.

(9)	a.	. . . смотрел	в	небо
		. . . (he) looked	into	the sky
		. . . Vmis-sma-p	Sp-a	Ncnsan
	b.	. . . в	небе	
		. . . Sp-a	Ncns**a**n+2c.Ncns**l**n+4	

Syntactic errors obviously conflate many different error types. The taxonomy for German from Boyd (2010), for example, includes selection, agreement, and word order errors. Our syntactic errors are either selection (e.g., wrong case as object of preposition) or agreement errors (e.g., subject-verb disagreement in number). However, without accurate syntactic information, we cannot divvy up the error space as precisely. With the POS information, we can at least sort errors based on the ways in which they vary from the original (e.g., incorrect case).

Finally, if no syntactic error can be derived, we revert to the correct form. This happens when the lexicon contains only one form for a given stem. Without changing the stem, we cannot generate a new form which is verifiably correct.

Decision #3: Morphological errors The next decision is: should we generate a true morphological error or a spelling error? We currently bias this by setting a 0.9 threshold. The process for generating morphological errors (0.9) is described in the next few sections, after which spelling errors (0.1) are described. Surely, 10% is an underestimate of the amount of spelling errors (cf. Rosengrant, 1987); however, for refining a morphological error taxonomy, biasing towards morphological errors is appropriate.

Decision #4: Invariant morphemes When creating a context-dependent morphological error, we have to ask what the unit, or morpheme, is upon which the full form is dependent. The final choice is thus to select whether we keep the stem analysis constant and randomize the suffix or keep the suffix and randomize the stem. Consider that the stem is the locus of a word's semantic properties, and the (inflectional) suffix reflects syntactic properties. If we change the stem of a word, we completely change the semantics (error type #1b). Changing the suffix, on the other hand, creates a morphological error with the same basic semantics. We thus currently randomly generate a suffix 90% of the time.

Random suffix generation Randomly attaching a suffix to a fixed stem is the same procedure used above to generate syntactic errors. Here, however, we force the form to be incorrect, not allowing syntactic errors. If attaching a suffix re-

[8] We ensure that we do not generate the original form, so that the new form is contextually-inappropriate.

sults in a correct form (contextually-appropriate or not), we re-select a random suffix.

Similarly, the intention is to generate inherency (#2bii), paradigm (#2c), and formation (#3) errors (or lexicon incompleteness). All of these seem to be more likely than derivation (#2bi) errors, as discussed in section 4.2.3. If we allow any suffix to combine, we will overwhelmingly find derivation errors. As pointed out in Dickinson and Herring (2008), such errors can arise when a learner takes a Russian noun, e.g., душ (*dush*, 'shower') and attempts to use it as a verb, as in English, e.g., душу (*dushu*) with first person singular morphology. In such cases, we have the wrong stem being used with a contextually-appropriate ending. Derviation errors are thus best served with random stem selection, as described in the next section. To rule out derivation errors, we only keep suffix analyses which have the same major POS as the stem.

For some stems, particular types of errors are impossible to generate. a) Inherency errors do not occur for underspecified stems, as happens with adjectives. For example, нов- (*nov-*, 'new') is an adjective stem which is compatible with any adjective ending. b) Paradigm errors cannot occur for words whose suffixes in the lexicon have no alternate forms; for instance, there is only one way to realize a third singular nominative pronoun. c) Lexicon incompleteness cannot be posited for a word with a complete paradigm. These facts show that the generated error types are biased, depending upon the POS and the completeness of the lexicon.

Random stem generation Keeping the suffix fixed and randomly selecting a stem ties the generated form to the syntactic context, but changes the semantics. Thus, these generated errors are firstly semantic errors (#1b), featuring stems inappropriate for the context, in addition to having some other morphological error. The fact that, given a context, we have to generate two errors lends weight to the idea that these are less likely.

A randomly-generated stem will most likely be of a different POS class than the suffix, resulting in a derivation error (#2bi). Further, as with all morphological errors, we restrict the generated word not to be a correctly-formed word, and we do not allow the stem or the suffix to be closed class items. It makes little sense to put noun inflections on a preposition, for example, and derivation errors involve open class words.[9]

Spelling errors For spelling errors, we create an error simply by randomly inserting, deleting, or substituting a single character in the word.[10] This will either be a stem (#1a) or a suffix (#2a) error. It is worth noting that since we know the process of creating this error, we are able to compartmentalize spelling errors from morphological ones. An error analyzer, however, will have a harder time distinguishing them.

5 Tagging the corpus

Figure 3 presents the distribution of error types generated, where *Word* refers to the number of words with a particular error type, as opposed to the count of error type+*POS* pairs, as each word can have more than one POS for an error type (cf. (9b)). For the 780,924 corrupted words, there are 2.67 error type+POS pairs per corrupted word. Inherency (#2bii) errors in particular have many tags per word, since the same suffix can have multiple similar deviations from the original (cf. (8)). Figure 3 shows that we have generated roughly the distribution we wanted, based on our initial ideas of linguisic plausibility.

Type	Word	POS	Type	Word	POS
1a	19,661	19,661	1b-2bi	11,772	11,772
2a	6,560	6,560	1b-2bii	5,529	5,529
2bii	150,710	749,292	1b-2c	279	279
2c	94,211	94,211	1b-3+	1,770	1,770
4	524,269	721,051			
3+	83,763	208,208	1b-all	19,350	19,350

Figure 3: Distribution of generated errors

Without an error detection system, it is hard to gauge the impact of the error generation process. Although it is not a true evaluation of the error generation process, as a first step, we test a POS

[9] Learners often misuse, e.g., prepositions, but these errors do not affect morphology. Future work should examine the relation between word choice and derivation errors, including changes in prefixes.

[10] One could base spelling errors on known or assumed phonological confusions (cf. Hovermale and Martin, 2008).

tagger against the newly-created data. This helps test the difficulty of tagging corrupted forms, a needed step in the process of analyzing learner language. Note that for providing feedback, it seems desirable to have the POS tagger match the tag of the corrupted form. This is a different goal than developing POS taggers which are robust to noise (e.g., Bigert et al., 2003), where the tag should be of the original word.

To POS tag, we use the HMM tagger TnT (Brants, 2000) with the model from `http://corpus.leeds.ac.uk/mocky/`. The results on the generated data are in figure 4, using a lenient measure of accuracy: a POS tag is correct if it matches any of the tags for the hypothesized error types. The best performance is for uncorrupted known words,[11] but notable is that, out of the box, the tagger obtains 79% precision on corrupted words when compared to the generated tags, but is strongly divergent from the original (no longer correct) tags. Given that 67% ($\frac{524{,}269}{780{,}924}$) of words have a syntactic error—i.e., a well-formed word in the wrong context—this indicates that the tagger is likely relying on the form in the lexicon more than the context.

	Gold Tags		
	Original	Error	# words
Corrupted	3.8%	79.0%	780,924
Unchanged:			
Known	92.1%	92.1%	965,280
Unknown	81.9%	81.9%	3,484,909
Overall	72.1%	83.4%	5,231,113

Figure 4: POS tagging results, comparing tagger output to *Original* tags and *Error* tags

It is difficult to break down the results for corrupted words by error type, since many words are ambiguous between several different error types, and each interpretation may have a different POS tag. Still, we can say that words which are syntactic errors have the best tagging accuracy. Of the 524,269 words which may be syntactic errors, TnT matches a tag in 96.1% of cases. Suffix spelling errors are particularly in need of improvement: only 17.3% of these words are correctly tagged (compared to 62% for stem spelling errors). With an ill-formed suffix, the tagger simply does not have reliable information. To improve tagging for morphological errors, one should investigate which linguistic properties are being incorrectly tagged (cf. sub-tagging in Hana et al., 2004) and what roles distributional, morphological, or lexicon cues should play in tagging learner language (see also Díaz-Negrillo et al., 2010).

6 Conclusions and Outlook

We have developed a general method for generating learner-like morphological errors, and we have demonstrated how to do this for Russian. While many insights are useful for doing error analysis (including our results for POS tagging the resulting corpus), generation proceeds from knowing grammatical properties of the original word. Generating errors based on linguistic properties has the potential to speed up the process of categorizing learner errors, in addition to creating realistic data for machine learning systems. As a side effect, we also added segmentation to a wide-coverage POS lexicon.

There are several directions to pursue. The most immediate step is to properly evaluate the quality of generated errors. Based on this analysis, one can refine the taxonomy of errors, and thereby generate even more realistic errors in a future iteration. Additionally, building from the initial POS tagging results, one can work on generally analyzing the morphology of learner language, including teasing apart what information a POS tagger needs to examine and dealing with multiple hypotheses (Dickinson and Herring, 2008).

Acknowledgements

I would like to thank Josh Herring, Anna Feldman, Jennifer Foster, and three anonymous reviewers for useful comments on this work.

[11] *Known* here refers to being in the enriched lexicon, as these are the cases we specificaly did not corrupt.

References

Bigert, Johnny, Ola Knutsson and Jonas Sjöbergh (2003). Automatic Evaluation of Robustness and Degradation in Tagging and Parsing. In *Proceedings of RANLP-2003*. Borovets, Bulgaria, pp. 51–57.

Boyd, Adriane (2010). EAGLE: an Error-Annotated Corpus of Beginning Learner German. In *Proceedings of LREC-10*. Malta.

Brants, Thorsten (2000). TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of ANLP-00*. Seattle, WA, pp. 224–231.

Chew, Peter A., Brett W. Bader and Ahmed Abdelali (2008). Latent Morpho-Semantic Analysis: Multilingual Information Retrieval with Character N-Grams and Mutual Information. In *Proceedings of Coling 2008*. Manchester, pp. 129–136.

Díaz-Negrillo, Ana, Detmar Meurers, Salvador Valera and Holger Wunsch (2010). Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum* .

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*. Columbus, OH, pp. 1–9.

Erjavec, Tomaž (2010). MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Proceedings of LREC-10*. Malta.

Ervin, Gerard L., Sophia Lubensky and Donald K. Jarvis (1997). *Nachalo: When in Russia . . . .* New York: McGraw-Hill.

Feldman, Anna and Jirka Hana (2010). *A Resource-light Approach to Morpho-syntactic Tagging*. Amsterdam: Rodopi.

Foster, Jennifer and Oistein Andersen (2009). GenERRate: Generating Errors for Use in Grammatical Error Detection. In *The 4th Workshop on Innovative Use of NLP for Building Educational Applications*. Boulder, CO, pp. 82–90.

Geyken, Alexander and Thomas Hanneforth (2005). TAGH: A Complete Morphology for German Based on Weighted Finite State Automata. In *FSMNLP 2005*. Springer, pp. 55–66.

Hana, Jirka and Anna Feldman (2010). A Positional Tagset for Russian. In *Proceedings of LREC-10*. Malta.

Hana, Jirka, Anna Feldman and Chris Brew (2004). A Resource-light Approach to Russian Morphology: Tagging Russian using Czech resources. In *Proceedings of EMNLP-04*. Barcelona, Spain.

Hovermale, DJ and Scott Martin (2008). Developing an Annotation Scheme for ELL Spelling Errors. In *Proceedings of MCLC-5 (Midwest Computational Linguistics Colloquium)*. East Lansing, MI.

Mikheev, Andrei (1997). Automatic Rule Induction for Unknown-Word Guessing. *Computational Linguistics* 23(3), 405–423.

Rosengrant, Sandra F. (1987). Error Patterns in Written Russian. *The Modern Language Journal* 71(2), 138–145.

Rozovskaya, Alla and Dan Roth (2010). Training Paradigms for Correcting Errors in Grammar and Usage. In *Proceedings of HLT-NAACL-10*. Los Angeles, California, pp. 154–162.

Rubinstein, George (1995). On Case Errors Made in Oral Speech by American Learners of Russian. *Slavic and East European Journal* 39(3), 408–429.

Schone, Patrick and Daniel Jurafsky (2001). Knowledge-Free Induction of Inflectional Morphologies. In *Proceedings of NAACL-01*. Pittsburgh, PA.

Sharoff, Serge, Mikhail Kopotev, Tomaž Erjavec, Anna Feldman and Dagmar Divjak (2008). Designing and evaluating Russian tagsets. In *Proceedings of LREC-08*. Marrakech.

Tetreault, Joel and Martin Chodorow (2008). The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of COLING-08*. Manchester.

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