Generating Modular Grammar Exercises with Finite-State Transducers

Lene Antonsen, Ryan Johnson, Trond Trosterud, Heli Uibo

University of Tromsø, Norway

lene.antonsen@uit.no, ryan.txanson@gmail.com,
trond.trosterud@uit.no, heli1401@gmail.com

ABSTRACT

This paper presents an ICALL system for learning complex inflection systems, based upon finite state transducers (FST). Using a FST has several advantages: it makes it possible to generate a virtually unlimited set of exercises with a relatively small amount of work, and it makes it possible to process both input and output according to a wide range of parameters, such as dialect variation, and varying writing conventions. It also makes it possible to anticipate common error types, and give precise feedback both on errors and possible corrections. It shifts the developer's focus from form generation and over to a pedagogically-motivated modelling of the learning task. The system is in active use on the web for two Saami languages, but can be made to work for any inflectional language.

KEYWORDS: ICALL, Morphology, FST, Generating Exercises.

1 Introduction

It has been argued that *Inflectional morphology is the bottleneck to language learning* of morphologically rich languages (Slabakova, 2009). This article presents a web-based ICALL system for learning two Saami languages, both morphologically complex languages. Although the system offers a wide range of learning tasks spanning from date and time expressions via vocabulary training to in-depth correction of free-input dialogues, the tasks targeting word inflection are by far the most popular in terms of actual use. This is a proof that Saami language learners consider our morphology drill programs useful.

The general focus within contemporary CALL development is on vocabulary applications. We felt that they neither provided what the student needed in order to produce target language utterances, nor made use of the linguistic insight which is found within computational linguistics.

Section 2 presents the motivation for our approach, and puts it in a wider context. Section 3 presents the system, Section 4 gives an evaluation of the generated tasks and of the logging popularity, and the last section gives a conclusion.

2 Background

At the outset, the main motivation behind our ICALL approach was a dissatisfaction with existing language-learner programs. These were mainly based upon English as a target language, and the programs did not take morphological complexity into account. For example, all the software listed in the Wikipedia article about CALL (`http://en.wikipedia.org/wiki/Computer-assisted_language_learning`) addresses English, except for a single CALL system for Basque. In the three volumes of the online journal Language Learning & Technology (`http://llt.msu.edu/archives/`) published in 2012 ten out of eleven papers deal with teaching English.

For the two Saami languages presented here comprehensive FSTs were available, and detailed enough to be able to function as an engine for spellchecking, thus covering the whole morphology and lexicon. We had a pedagogical philosophy which holds that "morphology is important", and used this plan as a basis for a turn-taking system in which students could learn to inflect verbs. As an afterthought, we also made the two programs presented here, for training word inflection, one with no context outside of the bare minimum needed to identify the target form, and another that included generated sentences with a question-answering frame.

Text-based ICALL systems for grammar learning can either be based on sentences extracted from a corpus, such as in Killerfiller (Bick, 2005), VIEW (`http://sifnos.sfs.uni-tuebingen.de/VIEW/`) and ESPRIT (Koller, 2005), where the user chooses a web page to extract text; or based on a strongly controlled lexicon and syntactic rules as in (Perez-Beltrachini et al., 2012), or the system presented here. ICALL based on actual texts is suitable for intermediate and advanced language learners, but for beginners, simplified language material with controlled lexicon and syntax are needed. Additionally, for languages with rich morphology, many of the combinations of stem type and inflection forms are infrequent in text. Covering all types requires more text than can be covered in a language course, and often also more text than is electronically available. The learner still needs these forms in order to master the system as a whole, and we thus argue for generated language material for beginner students.

There are some ICALL systems made according to the same principles. One of them is *ArikIturri*. (Aldabe et al., 2006), a grammar learning system for Basque. It can generate different types of questions: fill-in-the-blank, word formation, multiple choice, and error correction. The system makes use of question patterns encoded in XML and NLP tools for generation of exercises.

Another example is *Salama*, a system for learning Swahili, based upon a morphological FST (Hurskainen, 2009). The program is based upon so-called learning *tours*. The system starts out by giving the learner an arbitrary noun, and asks them to add an adjective to it, and then a pronoun, successively building rather complex NPs. The task is implicitly given via the initial word (*put the adjective in the same gender as the noun*), but the feedback put high demands upon the meta-grammatical knowledge of the users.

An example of a run of an exercise might go like this: **System:** *Type 'ndugu'!* ("brother"). **User:** `ndugu` **System:** *OK. Combine this noun with adjective 'pole' ("gentle")! (ndugu +N+HUM+9/10-SGndugu)*. **User:** `ndugu mpole` (and so forth, for other parts of speech, such as determinatives, numerals).

In addition to using FSTs to model the morphology, *Salama* also uses them for modelling word order. The user may thus add the NP members in several orders, but only the grammatical orders are accepted. Here, the FST contains an analysis (a path through the transducer) for each possible NP-internal word order pattern. It then returns a success tag, OK, to the grammatical strings, but separate error tags for all the wrong ones. These are then presented as error messages to the user, for example *Please check word order! Adjectives can't come before nouns!*. The system differentiates between spelling errors, which it reports as such, and concordance errors which is identified as *Please check the concordance!*,

Salama is a nice illustration of the possibilities given by FSTs. It is flexible and tolerates a wide range of input, while still being able to give precise feedback to the user, based upon an analysis of the input given. The system is reported to be operative, but no URL is provided. There is also no reference to actual use.

In (Dickinson and Herring, 2008) an idea of a FST-based system of morphology exercises for beginning learners of Russian is proposed. The intended system incorporated an error generating module that generated possible incorrect forms by combining the morphemes in incorrect ways. According to later publications (Dickinson, 2010) the system does not make use of FSTs, though, as initially planned. Unfortunately, there is no demo available of this system either.

3 Presentation of the System

Our system is a part of a larger system, *Oahpa* `http://oahpa.no`, and consists of an FST, a lexicon enriched with grammatical and semantic information, and templates for question-answering drill generation (Antonsen et al., 2009). These are all connected together by use of the programming language Python, and a MySQL database. For the web-specific aspects of this, the application relies on an open-source web framework, Django `http://www.djangoproject.com`. Data for lexica and morphological exercises are stored in XML files, with some morphological settings in plain text files, and these together are installed in the database.

An important point is that the use of FSTs and an XML format moves the focus from task generation to task adjustment, and one does not necessarily need to be a software developer in order to create new lexical entries and questions, but rather have some knowledge of how to edit XML, and run validation tools on the files. This means that the pedagogical idea behind each and every task is found in the lexicon, and in the information stored there, rather than completely stored in Python source code. This also means that the development of lexical data and question sets may be carried out primarily by linguists and specialists in the language, without necessarily having a programmer available to handle all the development.

3.1 Finite State-Transducers

The core of our system is an FST. The source files to the FST list all the stems and affixes, and concatenate them to word forms in a FST file. A separate transducer takes care of non-concatenative morphological processes resembling ablaut in Germanic languages. Note that in the Saami languages, these processes are fully productive, and not restricted to a closed set of common lexemes. Just listing the non-concatenative word forms is thus not an option. Figure 1 shows small parts of the two transducers for North Saami. The leftmost transducer turns *lemma form + grammar tag* into *stem + WG (weak grade marker) + suffix*. The rightmost transducer conducts the consonant gradation operation $vdn : vnn$ in the context of the weak grade operator WG. Cf. (Beesley and Karttunen, 2003) for a detailed explanation.

Each Saami stem combines with inflectional and derivational affixes and pragmatic clitic particles into literally hundreds of forms. In addition to enabling us to generate all these forms, the FST also gives us the possibility to model different versions of the word forms. To take a trivial example, the FST may contain an additional transducer allowing accented letters to be written without accents, but at the same time giving the correctly accented form back as feedback. A case in point is the South Saami *ï* in e.g. *gïele*, 'language', which is often rendered by writers as i. Instead of interrupting the exercise by demanding a correction of the i, the system accepts it, but presents the correct answer with the correct letter. The FST may also model dialect variation, and thus accept a dialectal suffix $-n$ instead of $-s$ for locative, but not n for other instances of s.

Thus, instead on focusing upon generating forms for morphological exercises, we let the FST generate the forms, and concentrate upon the pedagogical aspects of the formal variation of the forms.

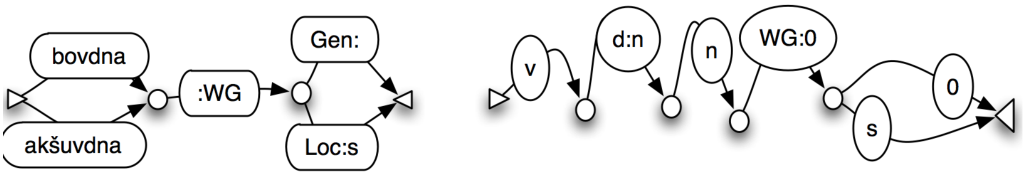


Figure 1: The FST to the left produces the North Saami pairs *bovdna+Loc:bovdnaWGs*, *bovdna+Gen:bovdnaWG* ('tussock'), *akšuvdna+Loc:akšuvdnaWGs*, *akšuvdna+Gen:akšuvdnaWG* ('action'). To the right another transducer produces the pair *vdnaWG : vnna0* . These are composed to give the result *bovdna+Loc:bovnnas, bovdna+Gen:bovnna, akšuvdna+Loc:akšuvnnas, akšuvdna+Gen:akšuvnna*.

3.2 Lexicon Structure

In addition to the FST, the other central resource for our language learning programs is a lexicon. It is a pedagogical lexicon containing the vocabulary of relevant textbooks.

The lexicon is stored in a MySQL database that is generated from XML source files. For each lemma, the data includes semantic classification, phonotactic and morphophonological information, dialect information and translations to pivot languages. The basic XML structure of the lexicon files is simple – each lexical unit is defined as an entity which may have any number of attributes depending on the word. Still there is no problem if some of the attributes are missing, see an example of a lexicon entry in Section 3.5.

While generating the lexical database, the morphological forms of the words are also generated by the FST and saved in database tables. That makes the generation of inflectional tasks quicker as there is no need to generate the forms at runtime. Forms are stored with reference to a morphological tag, and each morphological tag can belong to several tag sets. One can request all tags with a specific tense marking, person-number marking, mood, and so on; and also request general tags with any tense marking, or any person marking. For example, it is possible to retrieve the singular illative forms of all substantives that belong to the semantic category "BUILDING", or present tense, indicative mood, third person plural of the verb *geavahit* (en: *to use*).

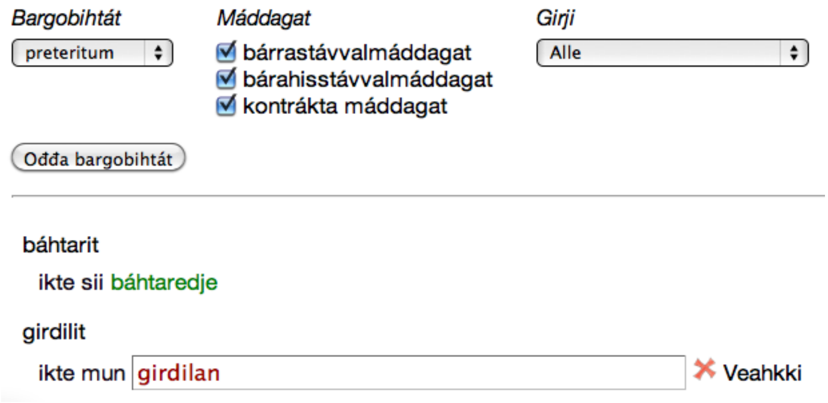
These divisions into tag sets are crucial in the production of morphological exercises.

The meta-information stored in the lexicon is there to select the appropriate words for the exercises. In addition, the morphological properties of words are used when providing detailed feedback on morphological errors.

3.3 Morphological Exercises

The first exercise type for morphology, *Morfa-S*, is purely inflectional, producing exercises with (almost) no accompanying context. The basic inflectional task starts out by giving the user two compulsory choices. First the user chooses which part of speech to inflect (or, in one case, to derive), Then, for the part of speech chosen, the user must choose an inflectional category: for verbs, either present, past, or one of the moods, and for the nominal categories one of the cases. For the verb exercises, the user then must produce the correct person-number form (there are 9 forms, representing 3 persons, and 3 numbers). For a question prompt, the user is given a verb in the infinitive, along with the relevant personal pronoun, for which they must fill in a blank containing the verb in the correct inflectional form for the chosen tense, see Figure 2. The user may also choose stem type (one of the major factors governing most aspects of the morphology of this language).

For the nominal forms, the user chooses case forms, and is then presented with words in either nominative singular or nominative plural. The task is then to give the corresponding singular or plural form for the case in question. Users have an alternative choice of choosing some specific morphophonological categories, such as words with an even number of syllables, words with an odd number of syllables, or contracted stems; which are important categories for determining what the inflectional stem of the word is.



Bargobihtát: preteritum

Máddagat:
☑ bárrastávvalmáddagat
☑ bárahisstávvalmáddagat
☑ kontrákta máddagat

Girji: Alle

báhtarit

ikte sii báhtaredje

girdilit

ikte mun girdilan ✗ Veahkki

Figure 2: The user has chosen verbs, and chooses also an inflectional category, and may also choose stem type. Five tasks will be generated each time, here we see two of them. The exercises are presented with the relevant personal pronoun and *ikte* (‘yesterday’) as context. The user is offered morphological feedback, that is described in Section 3.5.

Rather than giving beginner students the whole lexicon as a potential task, we made a controlled vocabulary of 1200 nouns, 750 verbs, 300 adjectives, and a handful of pronouns, and numerals from one to ten, as well as count words such as "many" and "few". The inflectional paradigms for this lexicon added up to approximately 80,000 wordforms. Drawn in sets of five at a time, this gives rise to a virtually unlimited number of tasks.

3.4 Contextual Morphological Exercises

The second exercise, *Morfa-C*, is contextual. In order to construct exercises the contextual system uses tags, tag sets, and semantic classes to fetch words from the lexicon. Exercise patterns are defined in XML source files, which are used to construct the necessary database relationships. The tasks consist of a question-answer pair, with a fill-in-the-blank in the answer. The surrounding context is thus natural language, and not a single pronoun, as for the pure inflectional excercise.

Each question is defined as a set of question elements, each defining either a syntactic function, or a lemma, and optionally with a set of syntactic tags or morphological tags in order to define which words can be used in the question element. Morphological tags can also be specific or general: either requesting a word of a particular part of speech, or a specific inflectional form, or a set of possible inflectional forms, via tag sets, for example a verb inflected in a specific person but with any possible tense. The element that represents the task for the learner is marked:

```
<question>
  <text>Maid SUBJ MAINV luomus</text>
  <element id="SUBJ">
    <grammar pos="Pron"/>
  </element>
  <element id="MAINV">
    <id>bargat</id>
    <grammar tag="V+Cond+Prs+Person-Number"/>
  </element>
</question>
<answer>
  <text>Luomus SUBJ V-COND</text>
  <element game="morfa" id="V-COND" task="yes">
    <sem class="ACTIVITY"/>
    <grammar tag="V+Cond+Prs+Person-Number"/>
    <agreement id="MAINV"/>
  </element>
</answer>
```

The above example is an exercise from a set of conditional mood sentences. The question and answer prompt (see <text> tag) translates to: 'What would PRON do on vacation? On vacation, PRON [....]'. Here, the pronoun in the question is generated together with the corresponding agreeing form of the verb *bargat* 'do'. In the answer the pronoun will agree with the question (explained below), and the task for the learner is to produce a conditional form of the verb with the correct person-number inflection corresponding with the pronoun in the answer sentence.

The user is provided with a lemma from the ACTIVITY-set containing 87 appropriate verbs. Together with 9 person forms of the verb, this would create a total of 783 possible activities.

As noted above, the generation of these activities requires that certain syntactic relationships be represented in the text shown to learners in order to construct natural sentences. For North Saami, the following agreement types are required: (1) subject correspondence between question and answers (e.g., question: "Did you...?", answer: "I did."); (2) main verb and subject agreement; (3) habitive agreement, which is a kind of number agreement between the existential verb and a non-subject argument; (4) reciprocal pronoun agreement with subject person; (5) reflexive agreement with subject person.

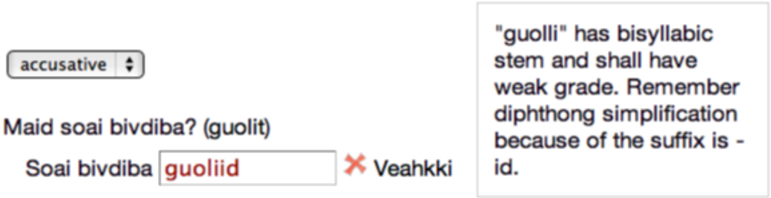
Although it is possible to formulate exercises that make use of more agreement, thus extending simpler question structures to cover a more complex set of sentences, there are some reasons to prefer defining simpler sentence types. First, it is overall a simpler task to produce more exercise definitions instead of fewer, more complex exercises. Second, in order to produce semantically natural sentences, it is better to make several, less complex questions in place of one, because this makes it possible to be very specific in the semantic sets used in question elements.

The general set of steps taken in generating an exercise are the following: (1) the system selects a question at random within the set of activities that the learner wishes to work on; (2) the system iterates through the question elements, selecting words that correspond to the grammar tags and semantic sets defined in the question; (3) agreement relationships are checked; (4) the expected correct forms are chosen for a particular answer, and then this is presented to the user.

The user is then presented with a set of generated questions, and prompted for input. After this is sent back to the server, it is checked against the correct answer or answers, if there are alternatives, as well as potential dialectical variants and orthographic "relaxed" variant. The user then sees two types of feedback from the system: whether or not they were correct, and whether their correct answer included non-standard forms, and then if they did not provide a correct answer, they are given morphological feedback to work on a correct answer. The user may repeat this process as many times as she likes, until she has filled out all answers correctly, or she may alternatively choose to see all the correct answers.

3.5 Feedback

Together with word forms, we also generate a set of relationships between forms and feedback messages, such that any given word in the system has a feedback to learners containing what they need to do to get the answer correct, as in Figure 3.



accusative

Maid soai bivdiba? (guolit)

Soai bivdiba guoliid ✗ Veahkki

"guolli" has bisyllabic stem and shall have weak grade. Remember diphthong simplification because of the suffix is -id.

Figure 3: The question is 'What do the two persons catch? They catch two __.' The task is to write the accusative form of the plural noun *guolit* 'fish'. The correct form is *guliid*. The feedback message consists of four separate parts (concerning *stem, grade, diphthong simplification and suffix*) put together.

Information about morphophonological features of the lemma *guolli* is stored in the lexicon:

```
<l diphthong="yes" gradation="yes" pos="n" finis="0" stemvowel="i"
stem="2syll">guolli</l>
```

This information combined with the information about the task itself implies tags that trigger messages in the chosen user interface language. For example:

```
<l stem="2syll" diphthong="yes" stemvowel="i">
<msg case="Acc" number="Pl">diphthongsimplification</msg>
```

This produces a tag, triggering the message "Remember diphthong simplification because of ". Another combination of the morphophonological information of the lemma and the task gives a tag which triggers the message "the suffix is -id".

Today, the feedback is the same regardless of the student's input, as long as it is recognised as incorrect. The language learner's errors can be accidental mistypings, but more often they are incorrect word forms due to misconceptions of the target language, and these misconceptions are therefore predictable.

The FST models the language in question by producing the correct word forms, but the FST can also model these kind of systematic misspellings with specific error tags in the upper level (Antonsen, 2012). In that way, the analyser identifies the nature of the erroneous form, and the feedback can instead contain general information about the nature of the lemma, as in Figure 3, where the feedback recognises the user's input and comments on the nature of the misspelling, such as: "*guoliid* lacks diphthong simplification caused by the stem vowel -i- plus the suffix -id." A more in depth survey of the logged incorrect forms provided by students would tell what kind of erroneous forms to generate in the FST.

3.6 Comparison to Other Systems

In the following section, some of the systems described in Section 2 that, similarly to our system, have the generation-based approach (in contrast to a corpus-based approach) are pointed out.

Our system is simpler than *GramEx*, presented in (Perez-Beltrachini et al., 2012), as the sentence patterns are defined in XML files that are easy to master for a linguist, whose job it is to formalise new exercise types. It is a straightforward procedure to retrieve words from the database that fit into the slots of these variables, based on the semantic and grammatical attributes of the word forms. In GramEx, there are complex algorithms for implementing grammar generation rules and constraints. It seems like the sentences are presented randomly and isolated. In our system they are presented as a question and answer pair, to give the student some context.

ArikIturri (Aldabe et al., 2006) has similarities with parts of our system: question patterns are also used in combination with meta-information in the lexicon, and NLP tools are used for form generation and analysis. Differently from our contextual morphological program, ArikIturri can have several blanks, which are to be filled in in one sentence. In addition, ArikIturri can generate different types of questions: fill-in-the-blank, word formation, multiple choice, and error correction. However, we do not share the pedagogical goals of using all of these, especially as concerns presenting incorrect forms to the students.

4 Evaluation

We first evaluate the generated question-answer frames, and thereafter we look at log data collected from the usage of the system.

4.1 Evaluating the Generated Tasks

For the contextual morphological exercises, there are altogether 330 templates for 34 different types of tasks with nouns, verbs, adjectives, pronouns, numerals and verb derivations. Factoring in the possible types of variation in each, they generate a total of 711,454 different exercises.

We randomly selected 10 generated question-answer-pairs of each task type from the North Saami system and asked two annotators to give a score from 1 to 3 for grammaticality and meaningfulness, to each question-answering pair. 3 was the best score. We also had an instructor give scores for the question-answering-pairs' appropriateness for the students. For appropriateness, 3 meant that she could have made a similar kind of exercise herself, 2 meant "not very good, but still possible to give to the students", and 1 meant that she would not have given it to her students at all. As we see from the results in Table 1, the results are good. The sentences marked as having bad grammaticality were partly due to errors in the database, and partly due to too sloppy restrictions on the sets. To take one example, in some cases, predicates put a restriction on their subject, demanding them to be plural, without this being reflected in the sentence frame. Sentences with low meaningfulness score typically violate selectional criteria. For users with a large vocabulary they might be amusing, at best, but for beginners they are mostly confusing. Sentences scoring low on appropriateness are mostly sentences scoring low on one or both of the other criteria.

	Grammaticality			Meaningfulness			Appropriateness		
Scores	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
Number of q.a.-pairs	30	17	308	31	33	281	23	42	295
Distribution in per cent	8.5	4.8	86.8	9.0	9.6	81.4	6.4	11.7	81.9
	average: 2.9			average: 2.8			average: 2.9		

Table 1: Evaluation of 340 randomly selected question-answer-pairs, from 34 different task types. The best score is 3 for each evaluation goal.

4.2 Logging User Activity

The morphological programs are part of set of 8 different exercises, spanning from date, number and vocabulary training to advanced dialogues, with in-depth comments upon the learner input. Usage statistics for North Saami during the period January 1st 2012 through March 8th 2013 (N=116,069) still shows that the overwhelmingly most popular exercises are the ones targeting morphological inflection. The two morphological exercises represent 52.9 % of student input (43.8 % and 9.1 %, respectively), as compared to 39.8 % for the lexicon exercises (vocabulary, clock, dates, numerals) and 7.3 % for (partly) free input dialogue exercises.

Usage data thus show that word form generation is seen as the most critical factor in Saami language learning according to adult learners.

We log all interactions between users and the system. For the morphology drill games, the exercise type and the student's score is saved together with the date and time and the student's username (if she has logged in to the course). Based on this data we can see which of the exercises are most difficult for the students —then the respective topics should get more attention in the course. We can also track the progress of individual students over the time.

Looking at this data, the correct percentage of the morphological exercises is 51.4 %, as compared to 58.7 % for the lexical exercises and 46.6 % for the free input exercises. There is thus no direct correlation between correct percentage and popularity.

In addition to logging data using the server itself, *Google Analytics* provides another kind of usage data, as well as demographic data about users collected from users' web browsers. With Google Analytics, it is easy to find where people are, what languages they are likely to speak (but mainly only majority languages from countries of origin), and also ways that users discover sites and the typical paths that they follow within them.

With these programs, Google Analytics was only taken into use on the 22nd of October, 2012, and has been available since. From this date to March 8th, 2013, Google Analytics tracked 3,676 unique visitors (in terms of uniquely identifiable web browsers) who visited both the North and South Saami sites a total of 5,301 times; and together all of these visits generated 53,751 individual page views. During the course of these visits, Google Analytics determined that on average, users would view 10.14 pages during their visit.

Google Analytics is also highly visual and as such provides another way of displaying demographic data. On the map, Saami regions in Scandinavia are strongly highlighted. Though there is not enough visit data from Russia to compare on the same level, the Murmansk Oblast has the most visits: 4, of a total of 12.

One of the more exciting pieces of data in Google Analytics is that it is clear that a fairly large set of "power users" are responsible for a large percentage of total page views, even though there are many more visits by other users who view less pages per visit. 12.8 % of total visits to these programs are responsible for 69.5 % of total page views per visit (these users on average also viewed upwards of 20 pages per visit), while the remaining 87.2 % are responsible for only 30.3 % of total page views (and these users instead viewed 19 or less pages per visit).

5 Conclusion

The use of FSTs and standardised XML formats to store lexicon and question templates allows for an easy, precise and efficient way to create a variety of complex morphological drills for learners of morphologically complex languages. These are in part built on already existing language resources, which are already in use in spellchecking and machine translation.

The exercises that these resources generate provide students an opportunity to not only learn how to produce specific words, but to produce them in context, as well as to learn the contexts which require the specific word forms. These exercises are also quite popular, reflecting that one of the language-learning tasks that learners identify as necessary is morphology.

Using FSTs we are able to manipulate both input and output according to a wide range of criteria. We may accept a larger range of user inputs based on dialect variation, relaxed spelling constraints, and in order to provide a precise-feedback system. The variation may cut across lexical categories, so that in one operation, we may allow for these kinds of variation across the lexicon as a whole.

The system has proven to be popular among students, and the most popular part of the program is the context-free inflection program. This is a clear indication that the students agree with Slabakova's claim that inflectional morphology plays a key role in language acquisition (Slabakova, 2009).

Combining the inflected forms with lexicon and template files, we are able to make tailored tasks, and also to let the user tailor her own tasks, such as practising past tense inflection only for even-syllabic word stems, and so on. Because developers do not need to focus on the form of the exercises, the pedagogic experience lies in how learners use the system, and how linguists generate the content.

Our proposed method is highly efficient for under-resourced languages, as it is not a requirement to have an extremely detailed and large morphological tool, or lexicon, in order to produce a useful amount of exercises for students. Resources like the one presented here also may be built by using existing resources, rather than needing to create completely new resources to function as the linguistic components for the system; finite-state transducers are typically made for spellchecking and machine translation applications, and lexica are made for dictionaries and for teaching materials. The infrastructure is portable, and given an available FST and vocabularies from existing learning material, an ICALL system like the one presented here may be built in a relatively short time.

Acknowledgments

Thanks to Norway Opening Universities for funding the project Interactive Saami instruction on the Internet, and to Saara Huhmarniemi for inspiring programming during the formative years of Oahpa.

References

Aldabe, I., de Lacalle, M. L., Maritxalar, M., Martinez, E., and Uria, L. (2006). Arikiturri: An automatic question generator based on corpora and NLP techniques. In *Proceedings of the 8th international conference on Intelligent Tutoring Systems, ITS'06*, pages 584–594.

Antonsen, L. (2012). Improving feedback on L2 misspellings – an FST approach. In *NLP for computer assisted language learning, SLTC 2012*, volume 80 of *Linköping Electronic Conference Proceedings*, Linköping, Sweden.

Antonsen, L., Huhmarniemi, S., and Trosterud, T. (2009). Interactive pedagogical programs based on constraint grammar. In *Proceedings of the 17th Nordic Conference of Computational Linguistics. Nealt Proceedings*, 4, Odense, Denmark.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI publications in Computational Linguistics, USA.

Bick, E. (2005). Live use of corpus data and corpus annotation tools in CALL: Some new developments in VISL. In Holmboe, H., editor, *Nordic Language Technology, Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004*, pages 171–185. Museum Tusculanums Forlag, København.

Dickinson, M. (2010). Generating learner-like morphological errors in russian. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 259–267, Beijing, China. Tsinghua University Press.

Dickinson, M. and Herring, J. (2008). Developing Online ICALL Exercises for Russian. In *Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9, Columbus, Ohio, USA. Association for Computational Linguistics.

Hurskainen, A. (2009). Intelligent computer-assisted language learning: Implementation to Swahili. *Technical Reports in Language Technology*, Report No 3:1–29.

Koller, T. (2005). Development of web-based plurilingual learning software for French, Spanish and Italian. In Cristina Mourón Figueroa, T. I. M. G., editor, *Studies in Contrastive Linguistics. Proceedings of the 4th International Contrastive Linguistics Conference (ICLC4)*. University of Santiago de Compostela Press.

Perez-Beltrachini, L., Gardent, C., and Kruszewski, G. (2012). Generating grammar exercises. In *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 127–156. Association for Computational Linguistics.

Slabakova, R. (2009). What is easy and what is hard to acquire in a second language? In Bowles, M., Ionin, T., Montrul, S., and Tremblay, A., editors, *Proceedings of the 10th Generative Approaches to Second Language Acquisition Conference (GASLA 2009)*, Somerville, MA. Cascadilla Proceedings Project.