NEALT Proceedings

Northern European Association for Language Technology

Proceedings of the Second Workshop on NLP for Computer-assisted Language Learning

NODALIDA 2013

May 22-24, 2013 • Oslo, Norway

Linköping Electronic Conference Proceedings

Proceedings of the

second workshop on NLP for computer-assisted language learning

at NODALIDA 2013

edited by

Elena Volodina
Lars Borin
Hrafn Loftsson

NEALT Proceedings Series 17 • ISBN 978-91-7519-588-9
Linköping Electronic Conference Proceedings 86
ISSN 1650-3740 (Online) • ISSN 1650-3686 (Print) 2013

Preface

ICALL – Intelligent Computer-Assisted Language Learning – is an interdisciplinary field whose aim is implementing and deploying applications for language learning based on Language Resources and Natural Language Processing (NLP), thereby opening the way for inclusion of open-ended language analysis and generation functionality in such applications.

Existing NLP tools and resources do not tend to find their way into the language learning classroom, despite their obvious potential uses in language learning. The reasons may be twofold. On the one hand, there is a lack of interested sponsors. On the other hand, there is a general lack of interest in the NLP community in CALL applications. While this situation arguably may have started to change for English, and a small number of other languages in the past ten years, it still holds true for the Nordic languages.

It seems that the few systems that have been developed for ICALL are either copyrighted and restricted by high licensing fees – and hence too expensive for universities and schools – or fall short of the required quality in linguistic or pedagogical functionality.

It is obvious though that ICALL holds a potential for applying NLP tools and NL resources in real-life conditions as opposed to laboratory tests and academic research. ICALL can help popularize NLP tools and NL resources among many users. At the same time, NLP technologies and resources can support teachers, relieving them from tedious tasks that can be modelled and carried out by computers.

This situation calls for a change and the successful first workshop on NLP for CALL (`http://spraakbanken.gu.se/eng/Research/icall/NLP4CALL`) organized in connection with the Swedish Language Technology Conference 2012 in Lund, as well as the recent establishment of the Special Interest Group at North European Association of Language Technology, NEALT SIG-ICALL (`http://spraakbanken.gu.se/eng/Research/icall/SIG-ICALL`), have shown that there is a need for a forum where these issues can be discussed.

In view of that, we took the initiative to gather interested researchers together and discuss experiences, challenges and successes in the area of ICALL development. In the call for papers we invited submissions on topics such as the following:

- research directly aimed at ICALL,
- actual or potential use of existing NLP tools or resources for language learning,
- ongoing development of resources and tools with potential usage in ICALL, either directly in interactive applications, or indirectly in materials, application or curriculum development, e.g. collecting and annotating learner corpora; developing tools and algorithms for readability analysis, selecting optimal corpus examples, etc.

We were especially interested in submissions describing work for Nordic languages.

We received a total of 8 papers, that have undergone blind review by three members of the program committee:

- Toni Badia, UPF, Barcelona
- Lars Borin, University of Gothenburg
- Robert Eklund, Linköping University
- Petter Karlström, Stockholm University
- Sofie Johansson Kokkinakis, University of Gothenburg
- Ola Knutsson, Stockholm University

- Hrafn Loftsson, Reykjavik University
- Montse Maritxalar, University of the Basque country
- Detmar Meurers, University of Tübingen
- Martí Quixal, Universty of Texas at Austin
- Mathias Schulze, University of Waterloo
- Joel Tetreault, Nuance Communications
- Cornelia Tschichold, Swansea University
- Elena Volodina, University of Gothenburg

Following the reviewers' recommendations, 5 submissions were accepted for presentation at the workshop and inclusion into the workshop proceedings volume, subject to revisions as recommended by the reviewers.

The workshop was opened by an invited talk on *Challenges in ICALL* given by Cornelia Tschichold (Swansea University, UK), followed by two sessions with oral presentations where a range of topics have been introduced. A general discussion concluded the workshop.

The workshop organizers:
Elena Volodina
Lars Borin
Hrafn Loftsson

WS website: `http://spraakbanken.gu.se/eng/forskning/ICALL/2ndNLP4CALL`

Acknowledgements: Financial support for the organization of the workshop has come in part from the NordPlus Sprog program (the project *A System Architecture for Intelligent CALL*), and in part from the University of Gothenburg, through its support of the *Centre for Language Technology*: `http://www.clt.gu.se`

Contents

Natural Language Processing for the Translation Class

Lars Ahrenberg[1] *and Ljuba Tarvi*[2]

(1) Department of Computer and Information Science, Linköping University
(2) Helsinki University

lars.ahrenberg@liu.se, ljuba.tarvi@welho.com

ABSTRACT

We propose a system for use in translation teaching with automatic support for alignment and comparative assessment of different translations. A primary use of this system is for discussion in class and comparison of student translations from a given source text, but it may also be used to study and compare differences between published translations. We describe the intended functions of the system and give suggestions on its design and architecture. We also discuss the degree of automation that can be expected and report results from a small indicative study focused on word alignment performance.

KEYWORDS: natural language processing, translation teaching, translation assessment, alignment

Lars Ahrenberg and Ljuba Tarvi 2013. Natural language processing for the translation class. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 1–10.

1 Introduction

With the advent of computational aids for translators, such as translation memory systems, terminology management systems and corpus search tools, the need to teach the use of such tools in translator training has also been recognized. From the perspective of natural-language processing, however, these tools are not very sophisticated. In particular, the technologies that have propelled the fast developments in machine translation have not been used as much as they could be.

In this paper we give a proposal for a system that can support the assessment and classroom discussion of students' translations. Crucial to both aims is having the translations aligned at the word and segment levels with the source text. From this alignment global metrics of the student translations can be computed, helping them to understand the style of their translation in relation to other translations, including published ones. By using quantitative measures that have been shown to correlate well with qualitative judgements, it also helps the grading of students' translations. Moreover, the alignment can support different kinds of visualizations of the students' work. Our hypothesis is that a sufficient alignment quality can be obtained by using a combination of automatic and interactive methods.

In the following, we first, in section 2, report on related work and then proceed, in section 3, to give an overview of the design and functions of the proposed system. In section 4 we describe preliminary results of a small experiment on alignment of texts that have been, or could be used for class instruction. Section 5, finally, holds our conclusions.

2 Related work

Translation memory systems and other CAT (Computer-Aided Translation) tools are increasingly being used in translator training, and the creation and use of corpora has been a common interest for translation studies, translator training and computational linguistics for several years (e.g. Zanettin et al., 2003). In translator training the corpora are mostly seen as resources for the student to use when translating (Lopez-Rodriguez and Tercedor-Sanchez, 2008; Pastor and Alcina, 2009). Proposals have also been made to use e-learning environments for specialized translation courses, where students' translations can be collected and compared by all course participants (Fictumová, 2007).

Also in our proposal immediate comparisons and assessments of students' translations are offered as a class-based activity. A system with some similarity is reported in Shei and Pain (2002: 323) who describe an "intelligent tutoring system designed to help student translators learn to appreciate the distinction between literal and liberal translation". Their system allows students to compare their own translations with reference translations and have them classified in terms of categories such as literal, semantic, and communicative. The comparisons are made one sentence at a time, using the Dice coefficient, i.e., by treating the sentences as bags of words. To contrast, our proposal uses more advanced computational linguistics tools, offers more teacher involvement, and provides text level assessment based on token alignments.

Our proposal relies heavily on recent advances within computational linguistics. In particular, it can be viewed as a test bed for current alignment technology. In addition it draws on the Token Equivalence Method (TEM; Tarvi, 2004) where the idea that translation correspondence at the token level is useful for the characterization and assessment of translations is developed (see section 3).

Alignment technology has advanced considerably over the years and is still an active area of research (Tiedemann, 2011). Word alignment is an input technology for a range of bilingual or multilingual NLP tasks. Most prominent of these is perhaps statistical machine translation (SMT; Koehn, 2009) but also many others such as terminology extraction, lexicon generation, and the creation of parallel corpora and treebanks. As far as we are aware, however, there is no published work reporting on alignment technology for use in the translation class.

3 System overview

The proposed environment has a central system for the teacher and a number of client systems for the students. The students' systems can be designed somewhat similar to a translation memory, allowing alternative views of the source and target texts and, if need be, enforcing alignment of a student's work with the source text at an appropriate segment level (sentence or paragraph). When a student has finished a translation task, she will save her translation in an XML-based exchange format such as an XLIFF extension and make it available for the teacher, say, by uploading it through a web interface. For the rest of this section we will focus on the teacher system.

The teacher's system is equipped with several modules for text analysis, including tokenization, lemmatization, part-of-speech tagging, sentence and word alignment, where automatic tools are integrated into an interactive environment. Any output from an automatic component can be reviewed and changed by the teacher. The teacher's system also has components for visualization and joint display of the student translations.

When a text has been selected for a translation exercise, it will be segmented, tokenized and indexed. The teacher can prepare the system dictionary for the new text as required and identify multiword units, including idioms, as units of special interest. When translations are returned, the teacher is acting as a post-editing human agent who can combine both manners of assessment, computer-assisted and manual. After tokenization and indexing the translations can be analyzed in the same way as the source text and be aligned with it. The teacher reviews the alignments and corrects the errors.

Sentence alignment can be enforced for a given translation task, but if the teacher does not want that, sentence aligners usually perform well enough on the kind of short texts that are suitable for a translation class. Word alignment is a different matter. While error rates as low as 5% or less have been reported on some data sets (Liang et al. 2006; Moore et al. 2006), such figures are hard to achieve**.** Only practice can show what level of accuracy is actually required for the system to be useful and requirements may be different for classroom display and for grading purposes.

Alignment of a source text with several translations runs the risk that different translations segment and order the content in different ways so that no single segmentation of the source text can be taken as adequate for all translations. Within a text we can recognize segments, phrases, and tokens. Segments should be big enough to have one-to-one corresponding segments in all translations. Tokens are the smallest text units and phrases are made up of one or more tokens within a segment.

The source text is maintained as a single file. It is connected to the translations via alignment files, one for each translation. Alignments at both segment and token level are represented in the alignment files. Translations of a source phrase can be computed for each translation based on the token alignments. It may of course happen that some part of a selected phrase has not been translated, or that the alignment contains more tokens than necessary. This information can be collected during the retrieval process and be displayed with the retrieved phrase.

3.1 Translation views

We imagine the system to support different views of the translations. A basic view is the **segment view** where a segment from the source text is displayed with one or more corresponding segments from the translations. This is the easiest one to implement as it only requires a correct segment alignment, where a segment may be a sentence, or a short paragraph. Words and phrases of interest in a source segment can be high-lighted, but the corresponding translations have to be recognized by the students without help from the system.

Another view is the **token view**, where a word or phrase at a specific position in the source text is singled out and its different translations are displayed. The display of translations can be restricted to an arbitrary subset of the translations, and the context can also be varied, say, to one or more segments or in terms of bytes. The matching tokens can be high-lighted against the still visible context.

A **type view** of the data is of interest when some word or phrase is used in different parts of the source text. Apart from just listing the different translations and their distribution on the students' texts, frequency tables are also compiled.

In addition, the system can display the outcomes of the different metrics that are described in the following section. These offer a **global view** of the translations, such as the amount of information from the source that are kept in the different translations. Such data can be displayed as a table, like Table 1 below in section 3.3.

3.2 Assessment and grading

There are a number of global metrics that can be computed from a word alignment. Here we follow the TEM framework. In Tarvi (2004) the TEM was used for comparing the classical Russian novel in verse by A. Pushkin *Eugene Onegin* (1837) and its then existing English translations. The quantitative figures calculated on 10% of the text of the novel showed a very good fit with the results obtained elsewhere on the same material by conventional comparative methods. Also, it could answer the question of which one of all the translations is the closest to the original, in both content and form.

Methodologically, the TEM focuses on what has been kept in translation. Two basic analytical planes are considered – content and formal. The lexical content of the original retained in its translation(s) is calculated as a percentage of the former. Several means of comparative assessment, in TEM referred to as 'frames', can be used, with the cumulative result – Translation Quotient (TQ) – calculated as an arithmetic mean of the percentages in all frames. There are also optional frames that focus on other characteristics of the translations that reflect the translator's style. In some analytical frames, the results are calculated as absolute numbers.

To illustrate the method, an eight-word excerpt (One LIX: 1-2) and the following five translations of *Eugene Onegin* are used: the translation by Vladimir Nabokov (1964), and the four latest versions – by Tom Beck (2004), Stanley Mitchell (2008), Henry M. Hoyt (2008), and D.M. Thomas (2011). The source sentence contains three Subject (S) – Predicate (P) groups, one Conjunction (C), and one Attribute (A):

Pushkin:	*1:Proshla*	*2:lyubov,*	*3:yavilas'*	*4:muza,*	*5:i*	*6:projasnilsya*	*7:tyomnyi*	*8:um.*
	[passed]	[love]	[appeared]	[muse]	[and]	[cleared up]	[dark]	[mind]
	P1	S1	P2	S2	C	P3	A	S3

The translations are shown with the alignments in the direction from translation to source inserted (punctuation marks are ignored). Thus, the first link (1-2) associated with Nabokov's translation says that the first word in the translation corresponds to the second word of the original. A zero (0) indicates that a word has no correspondent. For clarity multiword translations have been underlined and tokens with null links are indicated in bold:

Nabokov: Love passed, **the** Muse appeared, and **the** dark mind cleared up.

1-2 2-1 3-0 4-4 5-3 6-5 7-0 8-7 9-8 10-6 11-6

Beck: **Once** love **had** passed, **the** muse **then** surfaced, **the** darkness **in my** mind **had** cleared.

1-0 2-2 3-0 4-1 5-0 6-4 7-0 8-3 9-0 10-7 11-0 12-0 13-8 14-0 15-6

Hoyt: Love past, **the** muse **has** made appearance, and **the** dark mind **has** changed to light;

1-2 2-1 3-0 4-4 5-0 6-3 7-3 8-5 9-0 10-7 11-8 12-0 13-6 14-6 15-6

Mitchell: Love passed, **the** Muse resumed dominion and cleared **the** darkness **from my** mind,

1-2 2-1 3-0 4-4 5-3 6-3 7-5 8-6 9-0 10-7 11-0 12-0 13-8

Thomas: Love **as she** leaves lets in **the** Muse, and clarity **once more I find**.

1-2 2-0 3-0 4-1 5-3 6-3 7-0 8-4 9-5 10-6 11-0 12-0 13-0 14-0

Note the mode of alignment suggested here: only the meaningful denotative tokens are aligned, while added grammar tokens, such as *had* or *the,* are given null alignments. Thus, although token *6:projasnilsya* has been rendered as *cleared up* (Nabokov), *had cleared* (Beck), *changed to light* (Hoyt), *cleared* (Mitchell), and even as *clarity* (Thomas), all these renderings are viewed as retaining the denotative meaning of the original token. The connotative shades of meaning most suitable for the outlined goals can be discussed in class.

When employed manually, TEM employs such operations as consecutive numbering of the tokens in the source text; finding correspondences between the source and target tokens, identifying grammar tokens, parts of speech and syntactic positions, and calculating the obtained results as counts, percentages and Translation Quotients (TQ) for the purpose of grading. Therefore, the method generates absolute score (overall estimates) based on relative scores in separate frames (see Table 1).

All of this work can be automated, promising a substantial reduction in the time to perform a TEM analysis. Some of the automatic modules, given the current state-of-the-art will introduce a high number of errors, however, and for this reason, their output needs to be reviewed and corrected. The most critical one is the word alignment.

3.3 TEM Frames

In automatic mode, the (corrected) alignment files are used to calculate how much of the original information has been retained in the translations. Two content frames are used here – one basic, and one optional. The **basic content frame** (BCF) computes the number of source tokens that are part of a non-null alignment. This figure is then rendered as a percentage of the number of content tokens in the original. As is seen Nabokov, Hoyt and Mitchell translated all eight tokens and, hence, scored 100% each, Beck ignored *5:i* (87%), while Thomas has left out *7:tyomnyi 8:um* (75%).

The **optional content frame** (OCF) is a useful tool in additional assessment as it shows what has been added to the translation or that have no counterparts in the source texts. This can be calculated as an absolute number. Nabokov and Hoyt added no excessive content tokens, Mitchell added one (*from*), Beck – three, (*once, then, in*) Thomas – six (*as, she, once, more, I, find*). Note that not all null-aligned tokens are relevant to the OCF; grammar tokens that are required or suggested by the target language grammar are not counted. Thus, the OCF as other formal frames require an explicit recognizer for these tokens.

The formal frames pertains to the formal aspects of the translations in comparison with the original. In this analysis, there is a basic frame and two optional ones.

The **basic formal frame** (BFF), has the grammar tokens, – articles, tense markers, etc. at the centre of attention. Also these (*the, had, has, my*) can be seen to be employed in different quantities in the translations above: Thomas used only one, Nabokov – two, Mitchell – three, Hoyt – four, Beck – five. This frame, like other obliquely source-dependent frames, can say something about the translator's (or student's) individual style.

The **optional formal frame I** (OFF1) monitors another aspect of a translation. It counts the content tokens that are rendered with the same part of speech (PoS) in the translation as in the source. It is expressed as a percentage of all content tokens of the source. It is to be noted, that, like in other optional frames, the results reflect the translator's strategies to render the original rather than the intrinsic qualities of latter. Nabokov used in all eight tokens the same part of speech as in the original, Hoyt in seven (he used a participle, *past*, instead of the verb for 1:*Proshla*, Beck and Mitchell rendered the adjective *7:tyomnyi (dark)* with a noun *darkness*, while Thomas kept the PoS for only the first five tokens of the original.

Another way of gauging the ‘presence’ of the original in its translation is to register the syntactic changes. It is indisputable that there are certain syntactic changes in translations that are inevitable, due to the grammatical requirements of the target language, like, for instance, source tokens 1-2, 3-4, and 6-8 here, which can be translated into English only in a reverse order. However, translators have the option to reformulate and go beyond what is minimally required in rendering the contents of the source text.

If two tokens are rendered in the same sequence as in the original and preserve the same syntactic functions, they are considered kept. The **optional formal frame II** (OFF2) counts the number of such pairs and renders it as a percentage of all pairs. As could be expected, the most dramatic changes happened in the last group of tokens, Sts 6-7-8, with, for instance, St 8, originally Subject 3, rendered as Prepositional Objects (PO) by Beck and Mitchell; or St 7, originally an Attribute (A), rendered as a Direct Object (DO) by the same authors. Only Nabokov and Hoyt managed to have kept the attribute *dark* (St 7) in its original syntactic function.

To compute OFF2 automatically requires a good parser. Simpler measures that register reorderings from the alignments have been proposed in the literature, e.g. Kendall’s tau or the LRscore (Birch and Osborne, 2011). These measures, while not using syntactic functions can still rank different translations with respect to the amount of reordering.

	BCF	OFF1	OFF2	**TQ**	OCF	BFF	**Rank**
Translator	%	%	%	**pp**	count	count	**N**
Nabokov	100	100	25	**75**	0	2	**1**
Beck	87	75	0	**54**	3	5	**4**
Hoyt	100	87	25	**70**	0	4	**2**
Mitchell	100	87	0	**62**	1	3	**3**
Thomas	75	62	0	**45**	2	1	**5**

Table 1. The TEM applied to eight words; assessment and grading.

3.4 Grading

Grading can be based on the frames. The TEM employs a measure called the **translation quotient** (TQ) which is calculated as the arithmetic mean of the percentages obtained in the frames. Moreover, as all translations can be given a rank for each frame, they can also be ranked from the TQ (see Table 1). After class discussion the students can revise their translations and one more monitoring can be carried out. The final grade, which can be an arithmetic mean of the home and class grades, is not only displayed but is registered automatically. If, at the end of class, the final grades are exhibited on screen in their ranking order, it is the best possible motivation for students to work diligently both at home and in class.

4 What word alignment technology to use?

As word alignment is crucial to the proposal, it is of interest to know what performance we can expect from currently available alignment systems, and what work is required from the teacher in order to get data of sufficient quality.

Word alignment systems usually give two kinds of output, token-oriented and type-oriented. The token-oriented alignment connects positions of the parallel texts, while the type-oriented output provides associations of words and phrases from the corpus as a whole, with or without probabilities. In the case of machine translation and the extraction of lexical data, the token-based alignment is not of primary interest; it is rather the word and phrase associations that can be derived from it. In our application both are relevant, but the token-based alignment is primary.

The most widely used word alignment systems, such as Giza++ (Och and Ney 2003) and its relatives, are statistical, learning word translation probabilities from parallel data. The alignment problem that we wish to find a solution for has the following characteristics:

- The source text is usually short, maybe in the range of 500–2000 words
- There are several translations and the parallel corpus to be aligned can be built from all the different translations and repeated versions of the source text
- Source and target languages are known so available resources in the form of dictionaries, SMT phrase tables, morphological analyzers, taggers, named entity recognizers, and parsers can be used

The fact that the texts are short speaks against using a statistical aligner. On the other hand, since the number of different translations can be high, data may still be sufficient for the exploitation of statistical tendencies. Also, we may augment the corpus with relevant portions of free parallel corpora, such as Europarl, based on lexical overlap. As the languages and source text are known in advance, word aligners that are based on generic resources such as dictionaries, syntactic pattern correspondences, and distortion distributions, the latter computed, say, from parallel treebanks, can also be employed. A framework using such resources is the "pressure aligner" of Esplà-Gomis et al. (2012).

We have made initial studies of alignment performance of Giza++ and a pressure aligner for two data sets. The purpose of these experiments is to find out what level can be reached with these systems. In particular we want to study the effect of the number of translations available for the statistical aligner, the effect of text-specific dictionaries for the pressure aligner, and the possibility to combine the two methods.

The first data set we have used is Russian–English; it comprises 17 stanzas (1085 tokens) from Eugene Onegin and eight different translations. We applied Giza++ (model 4) with standard settings to this data varying the size of the training corpus from one translation to all eight. As expected, performance improved with the addition of more translations; for one translation precision and recall are close to 30%, for eight translations it rises to 48%. We have not yet applied a pressure aligner to this data.

The second data set is English–Swedish with student translations from a translation class. The source text is made up of two short text snippets used in translation exercises, altogether 1234 tokens. There are three student translations and one published translation. To augment the corpus, two translations made by Google Translate and Microsoft Translator were added. The test set is the first short text with 452 tokens. We report precision and recall figures for six different set-ups in Table 2. The first two rows shows that more training data helps performance of the statistical system Giza++ (model

4). PA-1 is a pressure aligner with a dictionary for the most common English words and a short list of syntactic pattern correspondences. PA-2 has an added lexicon with words from the source text including correspondences found in the test set. The table also shows performance for the union and intersection of the two best aligners.

Id	**System**	**Corpus size**	**Null links included**		**Null links excluded**	
			Precision	**Recall**	**Precision**	**Recall**
1	Giza++[1 trl]	452	0.68	0.68	0.75	0.65
2	Giza++[5 trl]	2260	0.75	0.74	0.82	0.70
3	PA-1	452	0.50	0.55	0.82	0.49
4	PA-2	452	0.61	0.66	0.89	0.61
5	Union(2,4)	N.A.	0.74	**0.79**	0.78	**0.78**
6	Intersection(2,4)	N.A.	**0.87**	0.54	**0.98**	0.53

Table 2. Word alignment results for different systems. Best values are shown in bold.

5 Conclusions

We have presented an innovative concept for computer-aided translation teaching, based on existing token-based analyses of translations from computational linguistics and translation studies. As word alignment is the most crucial process for the proposal, we have also reported a pilot study on the feasibility of current alignment technologies for use in the system.

While the word alignment evaluation is small-scale, we believe it shows promising results. The statistical aligner improves when more translations are used, and the pressure aligner is able to take advantage of small increments to its dictionaries. In addition, they both find correspondences that the other aligner does not, so results can be further improved by combining them. With these small amounts of data, however, both aligners produce too many null links. That is why performance is better when only non-null links are considered. For post-editing, it is probably better to leave null links out, but for the test corpus at hand, this still means that at least some 200 links need to be added for a complete alignment. This is quite a lot of work, in particular if we consider that it should be multiplied by the number of translations at hand.

Still, we have not exhausted the potential of our word aligners. Performance is likely to improve by extending training data with open parallel resources for the statistical aligner, and using a much larger dictionary and phrase list for the pressure aligner. Also, as interactive word alignment can arguably be said to have some pedagogical value for the analysis of translations, this is work that may sometimes be performed by the students as a class-based activity.

References

Beck, Tom (2004) *Eugene Onegin by Alexander Pushkin*, Sawtry: Dedalus.

Birch, Alexandra and Osborne, Miles (2011) Reordering Metrics for MT. In Proceedings of the 49th Annual Meeting of the ACL, Portland, Oregon, USA.

Esplà-Gomis, Miquel, Sánchez-Martínez, Felipe, and Mikel L. Forcada (2012) Using external sources of bilingual information for in-th-fly word alignment. arXiv:1212.1192v2 [cs.CL] 7 Dec 2012, [last visited on 17/12/2012].

Fictumová, Jarmila (2007). Technology-enhanced Translator Training (Possible Pitfalls and Problems – A Case Study). Eldům, eldům - databáze publikovaných článků, 2007, Neuveden, Neuveden, pp. 1-12.

Hoyt, Henry M. (2008) *Alexander Pushkin. Eugene Onegin. A Novel in Verse,* Indianapolis: Dog Ear Publishing.

Koehn, Philipp (2010). *Statistical Machine Translation*, Cambridge University Press.

Liang, Percy, Taskar, Ben, and Klein, Dan (2006). "Alignment by Agreement." In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 2006, 104-111.

López-Rodríguez, Clara Inés and Tercedor-Sánchez, María Isabel (2008) "Corpora and Students' Autonomy in Scientific and Technical Translation training" *Journal of Specialized Translation* (JoSTrans), Issue 09 (2008) pp. 2-19.

Mitchell, Stanley (2008) *Alexander Pushkin. Eugene Onegin*, London: Penguin.

Moore, Robert C, Yih, Wen-tau, and Bode, Anders (2006). "Improved Discriminative Bilingual Word Alignment." In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 2006, 513-520.

Nabokov, Vladimir (1975 [1964]). Nabokov, Vladimir. Eugene Onegin. Revised Edition. A Novel in Verse by Alexander Pushkin translated from the Russian, with a Commentary, by Vladimir Nabokov. In Four Volumes, New York: Boulinger Foundation.

Och, Franz Josef and Ney, Hermann (2003) A Systematic Comparison of Various Statistical Alignment Models, Computational Linguistics 29(1), 19-51.

Pastor, Verónica, and Alcina, Ampero (2009) Search techniques in corpora for the training of translators. *Proceedings of the workshop on NLP Methods and Corpora in Translation, Lexicography and Language Learning*, Borovets, Bulgaria, pp. 13-20.

Shei, Chi-Chiang and Pain, Helen (2002) "Computer-Assisted Teaching of Translation Methods." *Literary & Linguistic Computing*, Vol, 17, No 3 (2002), pp. 323-343.

Tarvi, Ljuba (2004) *Comparative Translation Assessment: Quantifying Quality*. Helsinki: Helsinki University Press.

Thomas, D.M. (2011) *Alexander Pushkin. Onegin*, London: Francis Boutle Publishers.

Tiedemann, Jörg (2011) *Bitext Alignment*. Morgan & Claypool Publishers.

Zanettin, Federico, Bernardini, Silvia, and Stewart, Dominic (Eds.) (2003). *Corpora in Translator Education*. Manchester: St Jerome Publishing.

Constraints in Free-input Question-Answering Drills

Lene Antonsen
University of Tromsø
lene.antonsen@uit.no

ABSTRACT
This article describes a set of question-answer drills for language learning for a richly inflected language. The drills have been in actual use for some time. They allow for free input and make use of a constraint-grammar-based system, which anticipates a number of grammatical errors and common misspellings and gives certain response types. The interactions between student and computer are recorded, and the log reveals that the free-input approach comes at a price: students tend to avoid complex constructions. In order to force the student to answer with more complex constructions, while still keeping the free-input approach, we implemented a solution with more constraints for the input. The exercise items are generated and each template gives rise to a huge numbers of exercises. Constraint grammar makes it easy to control for both grammar errors and adherence to the constraints given in the task. The evaluation on authentic learner data shows that constraining the user's input with the question itself, makes it possible to analyse the student's free input, with very good precision and recall. But parsing the input is only a part of the challenge of designing real-life ICALL systems. The article discusses other design issues related to question-answering drills.

KEYWORDS: Constraint Grammar, ICALL, Grammar Exercises, Syntax.

Lene Antonsen 2013. Constraints in free-input question-answering drills. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 11–26.

1 Introduction

Two question-answering drills (QA-drills) offering free input with immediate error feedback, have been available since 2009 for people learning North Saami. This article both presents an evaluation of the processing of the students' input, and how we have met some challenges we have seen from the learners' real-life use of the programs.

The QA-drills are a part of an ICALL system for North Saami called Oahpa! ('Learn!')[1] consisting of both word quizzes and morphological exercises with e.g. fill-in-the-blanks. The system was originally designed as a supplement to ordinary text books. Since 2012 the ICALL-programs have been integrated into the university's introductory courses, and integrated into web-based teaching materials[2], which are used together with teacher instruction.

North Saami is a morphologically complex Uralic language, and its orthographic conventions differ substantially from the native language of most of the students. The language demands a lot of practising before the student reaches the level of fluency required for everyday conversation. Since it is a minority language, with only appr. 18,000 speakers altogether in Norway, Sweden and Finland, learners often do not have enough opportunities to practise the language in a natural setting.

Crucial for learners of North Saami is the mastering of the morphological system, and the QA-drills handled in this paper are based on the concept *Focus on Form*, proposed by (Long, 1991). This means that in the context of a communicative interaction, the attention of language learners is drawn to the form of specific language features. This approach is contrasted with *Focus on FormS*, which is limited solely to the explicit focus on language features, and *Focus on Meaning*, which is limited to focus on meaning with no attention paid to form at all. According to a review on available research (Norris and Ortega, 2000), both Focus on Form and Focus on FormS lead to more substantial effects than implicit instruction.

The main goal of the programs was to develop a language tutoring system with error analysis. Immediate error feedback and meta-linguistic advice about morphology and syntax were seen as important requirements for the programs, grounded in research that shows that formal rule teaching is necessary for adult language learners (DeKeyser, 1995, 2000). Another goal was to make the QA-drills as open as possible for the students, imitating real-word communication with a native speaker.

Even if many ICALL systems of this kind are proposed in the literature, not many of them are fully integrated into real-life foreign-language programmes in universities. In addition to the system presented here, reported systems integrated in university instruction are found for Japanese, Portuguese and German (Nagata, 2002; Heift, 2001, 2010; Amaral and Meurers, 2011).

But being able to process ill-formed input is only part of the challenge of designing real-life ICALL systems; other challenges are how to avoid long instructions in the learners' L1 but still constraining the learner input so that it can be analysed well enough, and how to give appropriate feedback to the learner (Amaral and Meurers, 2011). An evaluation of the program's first three months of operation (Antonsen et al., 2009), and later supervising of the learner data, revealed that the learners' avoidance of complex constructions is a challenge in a free-input system, that the learners' misspellings make the human-computer interaction more

[1] http://oahpa.no/davvi/

[2] http://kursa.oahpa.no/

difficult and less interesting for them, and that many learners do not work through the whole dialogues. These challenges are the topics of this article, and they are treated in the following sections: The different QA-drills and their design and how we constrain the input is explained in Section 2. All QA-drills use the same analyser, based on finite-state transducers and constraint grammar, which is described in Section 3. Section 4 presents the human-computer interaction, the feedback to the users and an evaluation based on the log files. Section 4.2 looks at some other aspects grounded in direct responses from students. Both the evaluation and how we meet the challenges, are summarised in the conclusion in Section 5.

2 The Question-Answering Drills

The drills consist of questions, both of yes/no-questions and wh-questions. The pedagogical goal is to let the student exercise verb inflection by answering the question with correct person, tense and mood, and also use correct case on the noun. The students get tutorial feedback on grammatical errors in their input.

The *Dialogue QA-drill* offers six dialogues built up with ready-made sentences, based on real-world scenarios. In each dialogue there are alternative branches, and the navigation between the branches is made dependent upon the student's answers. For example, if the question is whether the student has a car, a positive answer will navigate to a branch with follow-up questions about the car. Some of the information in the student's input, is stored and used in the questions, e.g. the brand of the student's car or what kind of drink she has chosen. Each dialogue has an underlying pedagogical goal, e.g. the shopping-dialogue is for answering with accusative vs. nominative case, helping a friend moving furniture-dialogue is for answering with locative vs. illative case, and looking at prices in a shop dialogue is for comparison of adjectives.

In the *Open Generated QA-drill* the tasks are made by a sentence generator, in order to be able to create a large number of potential tasks. By tuning the generator, one can easily offer variation to the user, instead of tailoring every task with ready-made questions. The questions come randomly, but are grouped by level of difficulty.

```
    <question>
      <text>Mas SUBJ MAINV</text>
      <element id="SUBJ">
<grammar pos="N"/>
<sem class="FAMILY"/>
      </element>
       <element id="MAINV">
<grammar tag="V+Ind+Person-Number"/>
<id>ballat</id>
      </element>
   </question>
```

Figure 1: Example of a question template. The generated question (in the <text> element) consists of the interrogative *mas* 'what.LOC', the verb *ballat* 'fear.INF' and a noun from the FAMILY set. The generated question can be *Mas vieljat ballet?* 'What were the brothers afraid of?' The sentence generator handles the agreement between subject and main verb.

A template question matrix contains two types of elements: constants and grammatical units for words selected from a pedagogical lexicon of about 2,700 words that are considered relevant for the learners of the language, categorised by semantic sets. The sentence generator handles agreement, such as person and number agreement between the subject and the main verb. The format for the question 'What is/are SUBJ afraid of?' is presented in Figure 2. The noun for the variable SUBJ is drawn randomly from the FAMILY semantic set, which consists of 48 members, and it can be generated as either singular or plural. The agreement with the verbal is handled in the sentence generator. This sentence generator is also used for generating both question and answer for morphology-grammar exercises (Antonsen et al., 2013).

The Open Generated QA-drill and the Dialogue QA-drill give few constraints for the answer. The student is encouraged to answer with a full sentence and with the same verb as in the question, which is a natural way of answering a question in North Saami, but the purpose is also to force the student to inflect the particular verb. The logs reveal, however, that the students will not write more complex language than they have to. Some examples are the following: students will not answer with a complex NP if they can answer with just a pronoun or a noun, and they will not write a time-expression by using a PP, when they can answer with an adverb instead. The price we pay for the free-input strategy is thus that the users are not forced to exercise more complex language skills.

In order to get the users to construct more complex phrases, a new design of the Open Generated QA-drill has been introduced, here called *Constrained QA-drill*. It presents 2-4 lemmas, which should be used to construct the complete answer. This drill type is inspired by *e-tutor*, a program for teaching German to foreigners (Heift, 2001), in which the possible input is restricted to a set of lemmas which the user must use to construct a sentence. But unlike the *e-tutor* program, the drill in this paper is made by generated tasks, and it uses the same analyser as the other QA-drills, so it allows the student to add more words to the sentence than the given ones.

The lemmas are drawn from semantic sets so there is variation in the exercise items for the students. The system also offers the student the possibility of varying the answer as long as the given lemmas are a part of it. The question in Figure 2 is *Gean deivet gáffádagas?* 'Who did you meet at the cafe?', and for the answer three lemmas are given: *deaivat* 'meet.V', *suohtas* 'funny.Adj', *skibir* 'friend.N'. The QA-pair is glossed in Examples (1) and (2).

(1) Gean deivet gáffádagas?
who.ACC met.PRT.SG2 at-cafe.LOC
'Who did you meet at the cafe?'

(2) Mun deaivat suohtas skibir
I meet.LEMMA funny.LEMMA friend.LEMMA
'I met a funny friend.'

The system accepts many kinds of answers, as long as the three given lemmas form a correctly inflected NP, as presented in examples (3), (4) and (5):

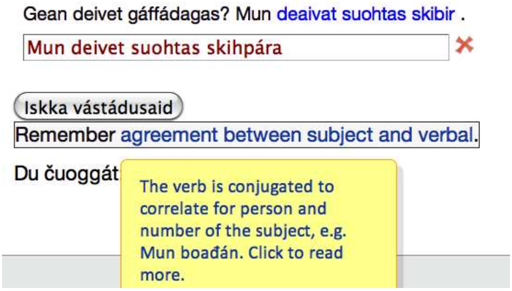
(3) Mun deiven suohtas skihpára.
I met.PRT.SG2 funny.ATTR friend.ACC
'I met a funny friend.'

(4) Mun han deiven iežan suohtas skihpára.
I emph.FOC.PCLE met.PRT.SG2 my.PRON.REFL funny.ATTR friend.ACC
'I (emph) met my funny friend.'

(5) Ikte mun vuot deiven iežan suohtas skihpára.
yesterday I again met.PRT.SG2 my.PRON.REFL funny.ATTR friend.ACC
'Yesterday I again met my funny friend'

The QA-task here is generated from a template with variables:
`<text>Gean MAINV gáffadagas</text> <text>Mun MAINV ADJ NOUN</text>`
The former one is the question and the latter one gives three lemmas for the answer.

The task is thus to inflect the verb and the adjective, which is drawn from a set of 45 suitable adjectives for nouns denoting humans, and the noun from a set of 44 members. Altogether, this template generates 1980 different exercises. Also the verb can be drawn from a set (for this task e.g. *meet, see, know...*), and the number of different exercises expands tremendously.



Gean deivet gáffádagas? Mun deaivat suohtas skibir .

Mun deivet suohtas skihpára ✗

Iskka vástádusaid

Remember agreement between subject and verbal.

Du čuoggát

The verb is conjugated to correlate for person and number of the subject, e.g. Mun boađán. Click to read more.

Figure 2: A example of a QA-task. The lemmas which the student has to inflect, are marked with blue colour. The feedback in the yellow window is a tool-tip, which appears on the student request, and its has a link to the relevant part of an online grammar. The sentences in the task are explained in Section 2.

Common for all three QA-drills is that the tutorial feedback concerning grammar errors is given in a separate window and the user is allowed to correct the answer until it is accepted. The user can choose the meta language (North Saami, Finnish, Swedish, Norwegian or English) because it is important that they understand the meta-linguistic issues reported by the system. The instructions about how to use the system, very limited, no long explanations, are given in North Saami, but the system offers translations in tool-tip, appearing on the user's request.

3 The System and the Analysers

The question-answer pairs are analysed with finite state transducers (FST) for morphology, and a constraint grammar (CG) rule set, which is used both for disambiguating and assigning grammar error tags as triggers for tutorial feedback to the user.

The morphological analyser/generator FST is compiled with the Xerox compilers `twolc` and `lexc` (Beesley and Karttunen, 2003). The lexicon contains 110,000 lemmas – almost half of them are proper nouns. The morphological disambiguator is implemented in the CG-framework (Karlsson et al., 1995).

In order to give better feedback to the students, the FST is enriched with some typical L2 misspellings, which are systematic, and these are marked with error tags (Antonsen, 2012). This makes it possible to some extent to give the student a precise feedback on misspellings, such as 'X should have consonant gradation', and offering more explanations about why. In the following example the wordform *addet* (the lemma is *addit*) gets an additional reading as a misspelling of *áddet*, (the lemma is *áddet*). The misspelling is marked with an error tag, `AErr`. This makes it possible, by means of CG-rules, to respond to the misspelling instead of responding to the verb which is actually written:

```
"<addet>" áddet V TV Ind Prs Sg2 AErr  'to understand'
"<addet>" addit V TV Ind Prt Sg2       'to give'
```

3.1 Analysing with Constraint Grammar

For compilation of CG-rules, `vislcg3` is used, this is a new and improved version of the free/open-source compiler `vislcg` (VISL-group, 2008). The program contains manually written, context-dependent rules, mainly used for selecting the correct analysis in case of homonymy. Each rule adds, removes, selects or replaces a tag or a set of grammatical tags in a given sentential context. Context conditions may be linked to any tag or tag set of any word anywhere in the sentence, either locally (in a fixed subdomain of the context) or globally (in the whole context). Context conditions in the same rule may be linked, so that they are conditioned by each other, negated or blocked by interfering words or tags.

The North Saami syntactic analyser based on constraint grammar has an F-score of 0.99 for part-of-speech (PoS) disambiguation, 0.94 for disambiguation of inflection and derivation, and 0.93 for assignment of grammatical functions (syntax) (Antonsen et al., 2010). CG is feasible for grammar checking, and in use for existing grammar checkers, e.g. of Norwegian, Swedish, Danish and Basque (Johannessen et al., 2002; Arppe, 2000; Birn, 2000; Bick, 2006; Uria et al., 2009). There is a prototype for native speakers of North Saami (Wiechetek, 2012). CG is also used in an ICALL program for annotation of free-user input for seven languages (Bick, 2005).

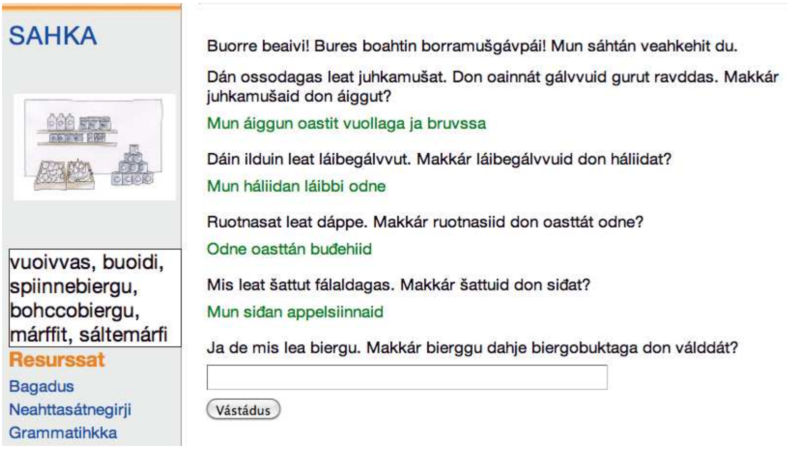
The `vislcg3` rule set used for analysing the input from the drills, consists of two parts. The first part is a set of 872 rules, which disambiguates the user's input only to a certain extent. The rule set is relaxed in comparison to the ordinary disambiguator, in order to be able to detect relevant readings despite grammatical and orthographic errors in the input. The second part of the rule set contains 247 rules for giving feedback to grammatical errors, and for the Dialogue QA-drill there are rules for navigating to the next question or utterance based on the user's answer.

Both the rules for assigning navigation tags and grammar error tags are in the same CG rule set. The advantage of having them in the same file, is saving starting up time, and the flexibility in ordering of the rules. It is e.g. possible to choose dialogue path before commenting on grammar errors. And one can choose to ignore misspellings, which are recognised of the system, in favour to navigate to the next question. To our knowledge, constraint grammar has not been applied for dialogue navigation before.

3.2 Feedback on Input

The system gives two kinds of feedback: If the student's answer is accepted, it turns green and the next question is presented, which in the Dialogue QA-drill can be a follow-up question, see

Figure 3. If the answer is not accepted, there will be feedback about the grammatical problem in the answer, and it can even point out the problematic word. Both feedback types are done by CG-rules.



SAHKA

vuoivvas, buoidi, spiinnebiergu, bohccobiergu, márffit, sáltemárfi

Resurssat

Bagadus

Neahttasátnegirji

Grammatihkka

Buorre beaivi! Bures boahtin borramušgávpái! Mun sáhtán veahkehit du.

Dán ossodagas leat juhkamušat. Don oainnát gálvvuid gurut ravddas. Makkár juhkamušaid don áiggut?

Mun áiggun oastit vuollaga ja bruvssa

Dáin ilduin leat láibegálvvut. Makkár láibegálvvuid don háliidat?

Mun háliidan láibbi odne

Ruotnasat leat dáppe. Makkár ruotnasiid don oasttát odne?

Odne oasttán buđehiid

Mis leat šattut fálaldagas. Makkár šattuid don siđat?

Mun siđan appelsiinnaid

Ja de mis lea biergu. Makkár bierggu dahje biergobuktaga don válddát?

Vástádus

Figure 3: From the Dialogue QA-drill. The setting is in a grocery and the learner is answering questions about what to buy. The available items show up in the window to the left. The accepted answer turns green, and the next question is presented. The pedagogical goal is to use accusative case in the answers.

The system's question is merged with the student's answer, and given to the analyser as one text string. Instead of a sentence delimiter, the question mark is exchanged with a question delimiter tag (QDL), so that the CG-rules can refer to the question and the answer separately, even if they are merged into one sequence. The question itself constrains the possible analyses of the user input.

The navigation in the Dialogue QA-drill is done by CG-rules, which e.g. assign tags to the string according to whether it contains an affirmative or negative answer, or assign a tag to the target of the question. The rule in Example (6) adds the tag `&dia-target` to the head of the NP, if it is in accusative, and there is no negation to the left of it (*–1), and the interrogative on the left side of the QDL asks for an accusative, defined as the set TARGETQUESTION-ACC. There are exceptions for the possibility of that the targeted word could be a genitive (which is homonymous with accusative) modifying a noun to the right (0 Gen LINK 1 N). The rule is simplified:

(6) ADD (&dia-target) TARGET NP-HEAD + Acc IF (*–1 QDL BARRIER Neg
LINK *–1 TARGETQUESTION-ACC)(NEGATE 0 Gen LINK 1 N) ;

The next question in the dialogue may comment the student's answer by including an inflected form of the `&dia-target`-lemma, or there may be rules navigating to another branch of the dialogue according to the lemma itself, like in the following example. If the student wants tea, she will be offered honey, but sugar if she prefers coffee. The question (the <text> element) is

'Do you want coffee or tea?' and the next question will differ according to the tags mapped by CG-rules to the answer. There will always be a default path if the analyser fails to interpret the answer in any of the predefined ways:

```
<text>Háliidat go gáfe vai deaja?</text>
<alt target="coffee" link="sugar_question"/>
<alt target="tea" link="honey_question"/>
<alt target="negative" link="drink_something_else_question"/>
<alt target="default"> link="next_topic"/>
```

The rules searching for grammatical errors in the input are common for all the QA-drills, and may depend on the morphology and the syntax in the question, for example, which case the interrogative asks for, or the verb tense, or the person-number inflection of the verb. Since the students' sentences are answers to known questions, the error-detection rules can be written accordingly, and problems with long dependencies, often faced by error-detection systems, have not occurred so far. The system also contains rules taking as their scope the right side of the QDL: subject-verbal agreement, NP-internal agreement and the case of nouns and pronouns based on the valency of the verb. The system is conservative and opts for safe error detection rules; false negatives instead of false positives, see Section 4.1.

The error tag in Figure 4 is mapped by a CG rule such as Example (7) (simplified):

(7) MAP (&grm-non-agr-subj-v) TARGET VFIN IF (0 $$PERSON-NUMBER-TAG
LINK –1 (Pers Nom) – $$PERSON-NUMBER-TAG LINK *–1 QDL) ;

This rule maps the error tag to the finite verb (VFIN) if its person-number tag is not the same as for the personal pronoun on the left side. The last constraint, that both the verb and the pronoun are in the answer, is given by asking for a QDL to the left (*–1 QDL).

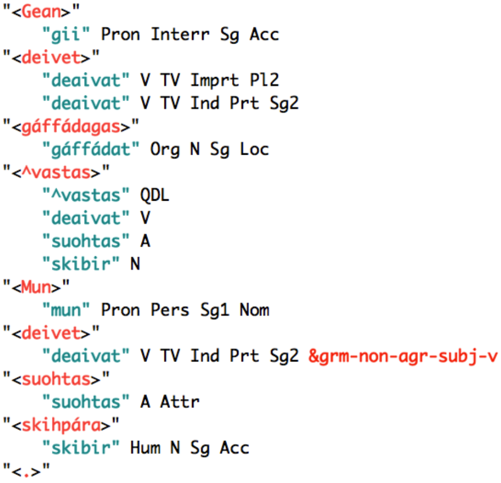
The given lemmas in the Constrained QA-drill are generated from sets in the lexicon and given to the analyser together with the question, stored in the same cohort as the QDL, as in Figure 4. CG-rules map error tags to the string if not all the given lemmas in the QDL cohort are represented in the answer. The handling of the question–answer pair is otherwise the same as for the other drills.

Grammar errors for which there are rules include:

- verbs: finite, infinite, negative form, correct person/tense according to the question
- agreement: subject/verbal, NP-internal
- case of argument and PP based upon the interrogative and valency
- time expressions, some special adverbs, particles according to word order
- comparison of adjectives

The differences between the three types of QA-drills are handled by means of ids assigned by the system to the input, which some of the CG-rules will refer to.

The system gives only one feedback message at a time, even if there are several error tags assigned. The choice of message is decided by the ordering in the message file. The errors based on local context are prioritised, e.g., first giving a message about spelling errors before possible agreement errors, and agreement errors inside the NP are prioritised over agreement between subject and verb, and so on. In some cases two error messages can be triggered by the



```
"<Gean>"
    "gii" Pron Interr Sg Acc
"<deivet>"
    "deaivat" V TV Imprt Pl2
    "deaivat" V TV Ind Prt Sg2
"<gáffádagas>"
    "gáffádat" Org N Sg Loc
"<^vastas>"
    "^vastas" QDL
    "deaivat" V
    "suohtas" A
    "skibir" N
"<Mun>"
    "mun" Pron Pers Sg1 Nom
"<deivet>"
    "deaivat" V TV Ind Prt Sg2 &grm-non-agr-subj-v
"<suohtas>"
    "suohtas" A Attr
"<skihpára>"
    "skibir" Hum N Sg Acc
"<.>"
```

Figure 4: The question and answer pair is given to the analyser as one string. The given lemmas are placed in the same cohort as the QDL (question delimiter) and CG-rules map error tags to the string if they are not represented in the answer. The input is disambiguated to some degree. A CG rule maps the `&grm-non-agr-subj-v` tag to the verb to trigger feedback on the missing subject-verbal agreement. The question-answer pair is translated in Section 2.

same error, and both the error tags will disappear when the student corrects the input after the feedback about the first one.

4 Human-Computer Interaction

The design of the QA-drills should match the students' needs, and it is therefore crucial to study the interaction between the user and the system. The recording of the human-computer interaction makes is possible to evaluate and to improve the system.

Immediate error feedback is an important requirement of the system. Above all the feedback should support and facilitate learning, and the error should be seen as a chance of getting the language learner not only to correct the input, but also understand the reason for her misconception.

In Table 1 is an example of a good interaction. The human-computer interaction in Table 2 is not very good. The student gets misleading feedback on the first answer, and has to correct the input a total of three times, before it is accepted.

The users are not homogenous, and it is a challenge to give feedback on the correct level for each learner. By requiring users to log into the system, it would be possible to give individualised feedback. Based on the stored data about their individual performance one could modulate the instructional feedback according to a student model, in which the user is classified as a novice, an expert on so on, on the particular grammar skill. The student model could also be used to provide exercises to the student which focus on their area of weakness. See (Heift and Schulze, 2007) for more.

Boađát go odne? (‘Do you come today?’)		
user’s input	**System’s judge**	**System’s feedback**
In sáhtte boahtit odne. ‘I don’t "can" come today.’	0	"sáhtte" has wrong form. It comes after the negation verb and should have negation form.
In sáhte boahtit odne. ‘I cannot come today.’	1	

Table 1: An example from the log of a good human-computer interaction. The student corrects her input according to the feedback, and the answer is accepted. System’s judge: <1> means that the answer is accepted.

Lea go dus heasta? (‘Do you have a horse?’)		
user’s input	**System’s judge**	**System’s feedback**
mus ***in*** *leat heasta* ‘I don’t.Sg1 have a horse’	0	Are you confident that you answer in correct person?
mus ***in lean*** *heasta* ‘I don’t.Sg1 have.Sg1=PrtConNeg a horse’	0	Are you confident that you answer in correct tense?
*m*us **in lea** heasta ‘I don’t.Sg1 have.Sg3 a horse’	0	The verb should have negation form.
mus ii leat heasta ‘I don’t.Sg3 have.Sg3 a horse’	1	

Table 2: An example from the log of a human-computer interaction which is not optimal. The problematic words are here marked with bold. In the first input the negation verb should have agreed with the noun, *ii.Sg3* instead of *in.Sg1*. In stead the student corrects the correct infinite form of the main verb to a form which can be interpreted both as Prs.Sg1 and Prt.ConNeg, and therefore the next feedback comments the tense. A better feedback to the first input would have been: ‘Are you confident that "in" is the correct person?’

The system in this paper can be used without logging in, and we have chosen an approach in which the student herself choses how much information she needs. The tutorial feedback is provided to the student on three levels: a short description of the error is always present, on request more information about the grammatical feature is given in a tool-tip, which also contains a link to the relevant part of an online grammar reference (see Figure 2).

Even if the target group for the programs are university students learning North Saami, the programs are freely available on the Internet. One can tell from the usage logs that school pupils use the QA-drills, when they give information about their age and what they do in their answers to the questions in the Dialogue QA-drill. The system’s feedback is targeted at students who know the linguistic terminology, and the logs reveal that many users do not always respond to it, probably the young ones, and they do not use the grammar links in the feedback. Therefore they often write many erroneous answers to the same question.

During the first years of operation the dialogues consisted of up to 28 questions in the longest paths through the branches. We learned from the logs that not many students worked through the whole dialogues. We now offer more, but shorter dialogues, each consisting of 8-14 questions, which seem to be a more appropriate number for the students.

4.1 Evaluation of the User Log

The evaluation is two-fold, both are based on the real use of the QA-drills. In Table 3 is a comparison of the error feedback the system has added to the users' input, for the same period for the Dialogue QA-drill and Open Generated QA-drill with little constraints for the input, and the Constrained QA-drill.

The misspellings make a large part of the error feedback for all QA-drills, even more often for the Constrained QA-drill than for the other ones. The reason is probably that the user is forced to inflect the given lemmas, and cannot choose to answer with simpler words. Most of the misspellings are systematic, and the FST should have been enriched with more erroneous forms marked with error tags. But even if the users to some extent get specific feedback to the misspellings, the log reveals that the young users don't always understand the meta-linguistic feedback.

For the not-constrained QA-drills 17.7 % of the feedback concerns missing finite verb in the answer. A common reason is that the user answers with a single word. Using the Constrained QA-drill it is clearer for the user that the answer has to contain a verb, and this feedback is quite rare, only 1.3 %.

The feedback on semantics in the Constrained QA-drill concerns using the given lemmas. The evaluation revealed that to Sg2-questions about personal information, the users tended to answer intuitively without the given lemmas. In the Dialogue QA-drill the users sometimes in their answers failed to relate to the objects in the dialogue setting, e.g. to which room they will suggest to put a furniture, even if a list of the available rooms are given in the interface.

	Other QA-drills		**Constrained QA-drill**	
CG rule type	**N**	**%**	**N**	**%**
misspellings	307	37.5 %	329	43.7 %
no finite verb	145	17.7 %	10	1.3 %
wrong case/number	102	12.5 %	133	17.7 %
verbal-subject agreement	94	11.5 %	75	10.0 %
comments on semantics	89	10.9 %	119	15.8 %
wrong verb form	35	4.3 %	36	4.8 %
NumP internal agreement	27	3.3 %	22	2.9 %
other	12	1.5 %	3	0.4 %
NP internal agreement	7	0.9 %	26	3.5 %
altogether	818	100.1 %	753	100.1 %

Table 3: Rules in use for a corpus of logged 2834 question-answer pairs.

In the Dialogue QA-drill there are questions about age and family, and a numeral phrase is prohibited in the answer. The feedback about missing agreement inside a numeral phrase makes up for 3.3 % of the feedback, which is almost the same as for the Constrained QA-drill (2.9 %). But the users of the Constrained QA-drill get feedback on NP-internal agreement almost four times as often as the users of the unconstrained QA-drills. The conclusion to Table 3 is that the Constrained QA-drill makes the users form more complex sentences, with finite verb and more complex NPs, than the unconstrained QA-drills.

The other part of the evaluation was to calculate the precision and recall of the system's judgement of the user input. Two parts of the usage log were annotated, for the unconstrained and for the Constrained QA-drill. The results of the annotating are presented in Table 4. Every error-feedback or no-feedback from the system was annotated with true or false.

The precision and recall for the Constrained QA-drill is very good, with the score for precision slightly better than for recall. That means that the system more often slips some errors through, than it flags non-existing errors. For the unconstrained drills precision is not so good: 0.93; the system flags an error which is not there in 7 % of the cases. For 44.7 % of the wrongly flagged errors the reason is a bad CG-rule for accusative in time-expressions. The rule was easy to improve. For some cases there are spelling variants of the word in question, which result in different lemmas in the morphological analysis, and the user gets incorrect feedback on not having used the given lemma or referred to the object in the dialogue. This can be fixed in the FST by unifying variants to one common lemma. In five cases the wordform was missing in the analyser because of limitations done to reduce the compilation time. This proves how important it is to constantly supervise the user logs.

	True pos.	True pos. but not corr. feedback	False pos.	True neg.	False neg.	Prec.	Rec.
Not constrained QA-drills, N=982	493	44 = 8.9 %	36	439	12	0.932	0.976
Constrained QA-drill, N=1114	749	72 = 9.6 %	3	352	9	0.996	0.988

Table 4: Precision and recall for a part of the user log for the Dialogue QA-drill and the Open Generated QA-drill compared the a part of the log from the Constrained QA-drill.

Important for a good human-computer interaction is that the feedback on the error addresses the error the student has made. For 8.9 % and 9.7 % of the true positives the feedback was misleading, like the example in Table 2. In some other cases the problem was caused by two errors in the same word. The first feedback addressed a specific misspelling. When the student had corrected the misspelling, the next feedback informed the user that the inflection nevertheless was wrong in the context. This illustrates that the algorithm explained in Section 3.2 not always is the best one, and in some cases it has to be modified. Another ordering of the error-messages would have made it possible to comment the inflection despite the misspelling.

Sometimes when the user does not know how to correct the answer, the lemma belongs to a group of stems, which does not follow the general inflectional paradigm. In such cases the feedback should be more specific and address these features for the particular lemma. E.g. even if the main rule is consonant gradation in the stem in certain inflections, there are classes of stems for which this rule does not apply. These words can be recognised by their morphological properties, and it would be possible to give specific comments in the error feedback, e.g. ‘Be aware of that X is a derived agent noun, and therefore it has no consonant gradation.’

4.2 Students’ Responses

In some periods there has been a feedback questionnaire on the web. Some users have asked for an answer key to the questions. For the Dialogue QA-drill this is critical because the user is not able to continue the dialogue before the answer is accepted. We have implemented a solution, which allows the user to request for an example-answer after the second time the answer is not accepted. The user still has to type in the answer herself. This was easy to do for the ready-made questions. We will also consider to generate example-answers for the generated questions.

Some students ask for audio files connected to the dialogues. It will be possible to record all the Dialogue QA-drill with ready-made questions. One could first offer only the audio file for the question, and let the student ask for the text, if needed. For the generated questions one could do the same using a text-to-speech program, if the quality were good enough. Such a program for North Saami is under construction now, and will be available in the future.

The evaluation revealed that the young users do not always understand the meta-linguistic feedback. Even if it would have been easier to give a appropriate feedback to the users if by requiring them to log in, we have hesitated to do that, because of the lack of North Saami teaching materials in the whole educational system. A way of giving them a better feedback, but still keep the possibility of using the programs without logging in, could be asking the users to type their age before they start the exercise, and address adapted feedback for the different age groups.

5 Conclusion

This article has presented an evaluation of QA-drills based on authentic learner data. One of the drills consists of pre-made questions, the other one uses a question generator, which makes a large number of exercises from each template. All the QA-drills give meta-linguistic feedback to the user, in the dialogue QA-drill a correct answer will be followed up by a further question.

Language learners are generally not able to evaluate the feedback in the way a native speaker does, therefore it is crucial that the system gives appropriate feedback and does not flag false errors. The evaluation of authentic learner data shows that constraining the user’s input with the question itself, makes it possible to analyse the student’s free input with very good precision and recall. For 8.9–9.7 % of the true errors the feedback was still misleading, but for most these instances, it was possible to do better by improving the rules or the ordering of the rules.

The learners’ avoidance of complex constructions is a challenge in a free-input system, but constraining the input with given lemmas to build their answer is a way of getting the learners to write more complex language, while still being within the free-input approach. Constraint grammar is very flexible, and can easily be used to check whether the student really has used the given lemmas in the answer, even when the lemmas are generated from large lemma sets.

Studying the authentic human-computer interaction is important in order to see how the system functions as a whole, e.g. whether the learner carry through the whole dialogue or not, or whether she understands the meta-linguistic feedback, so the input can be corrected. From the logs we see that the dialogue QA-drill often attracts users that do not have enough knowledge of the orthography or the linguistic terminology in the feedback. There is a mismatch between the banal content of the questions in the dialogue, and the necessary skills in orthography and grammar for participating in them. A meaning-based dialogue can loose its meaning when the learner to often is interrupted by comments on grammatical errors.

The constrained QA-drill seems to attract the target group, the students, better, because they give the impression of focusing on lemmas to be inflected and put into correct syntax, instead of trying to give the impression of a real-life conversation.

Despite the lack of a coherent semantic content for the constrained QA drill, the CG-parser gives the computer an intelligent behaviour. This makes it possible to give a sophisticated error analysis and the true interaction between student and computer asked for in (Heift and Schulze, 2007).

Acknowledgments

Thanks both to my colleague Heli Uibo who has helped me with implementing the QA-drills, and to Norway Opening Universities for the period 2011–13 funding the project Interactive Saami instruction on the Internet, which has been aimed on integrating the ICALL-programs into the university programme. Thanks also to Trond Trosterud for discussions.

References

Amaral, L. A. and Meurers, D. (2011). On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*, 23(1):4–24.

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). Constraint grammar in dialogue systems. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 8 of *NEALT Proceeding Series*, pages 13–21, Odense, Denmark.

Antonsen, L., Johnson, R., Trosterud, T., and Uibo, H. (2013). Generating modular grammar exercises with finite-state transducers. In *2nd workshop on NLP for computer-assisted language learning, NoDaLiDa 2013*, volume 85 of *Linköping Electronic Conference Proceedings*, Linköping, Sweden.

Antonsen, L., Trosterud, T., and Wiechetek, L. (2010). Reusing grammatical resources for new languages. In *Proceedings of LREC-2010*, Valetta, Malta. ELRA.

Arppe, A. (2000). Developing a grammar checker for Swedish. In *Proceedings of the 12th Nordic Conference of Computational Linguistics, NoDLiDa 1999*, pages 13–27.

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.

Bick, E. (2006). A constraint grammar based spellchecker for Danish with a special focus on dyslexics. In Suominen, M. e. a., editor, *A Man of Measure – Festschrift in Honour of Fred Karlsson*, volume 19, pages 387–396. The Linguistic Association of Finland, Turku.

Birn, J. (2000). Detecting grammar errors with Lingsoft's Swedish grammar checker. In *Proceedings of the 13th Nordic Conference of Computational Linguistics, NoDLiDa 1999*, pages 28–40.

DeKeyser, R. (1995). Learning second language grammar rules: an experiment with a miniature linguistic system. *Studies in Second Language Acquisition*, 17:379–410.

DeKeyser, R. M. (2000). The robustness of critical period effects in second language acquisition. *Studies in Second Language Acquisition*, 22:499–533.

Heift, T. (2001). Intelligent language tutoring systems for grammar practice. *Zeitschrift fur Interkulturellen Fremdsprachenunterricht*, 6(2).

Heift, T. (2010). Developing an intelligent language tutor. *CALICO Journal*, 27:443–459.

Heift, T. and Schulze, M. (2007). *Errors and Intelligence in Computer-Assisted Language Learning. Parsers and Pedagogues*. Routledge, New York and London.

Johannessen, J. B., Hagen, K., and Lane, P. (2002). The performance of a grammar checker with deviant language input. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1223–1227, Taipei, Taiwan.

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

Long, M. H. (1991). Focus on Form: A design feature in language teaching. In *Foreign Language Research in Cross-cultural Perspective*. John Benjamis publishing company, Amsterdam – Philadelphia.

Nagata, N. (2002). BANZAI: An application of natural language processing to web based language learning. *CALICO Journal*, 19(3):583–599.

Norris, J. M. and Ortega, L. (2000). Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis. *Language Learning*, 50(3):417.

Uria, L., Arrieta, B., de Ilarraza, A. D., Maritxalar, M., and Oronoz, M. (2009). Determiner errors in Basque: Analysis and automatic detection. *Procesamiento del Lenguaje Natural*, 43:41–48.

VISL-group (2008). Constraint grammar. `http://beta.visl.sdu.dk/constraint_grammar.html`.

Wiechetek, L. (2012). Constraint grammar based correction of grammatical errors for North Sámi. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SaLTMil 8 – AfLaT2012)*, pages 35–40.

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.

Lene Antonsen, Ryan Johnson, Trond Trosterud and Heli Uibo 2013. Generating modular grammar exercises with finite-state transducers. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 27–38.

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.

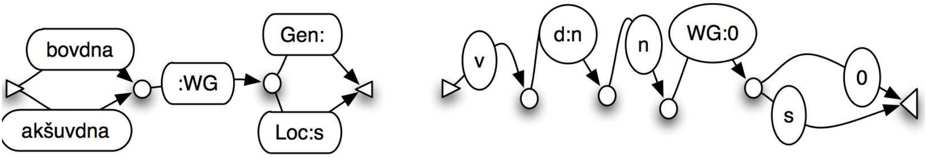


Figure 1: The FST to the left produces the North Saami pairs *bovdna+Loc:bovdnaWGs*, *bovdna+Gen:bovdnaWG* ('tussock'), *akšuvdna+Loc:akšuvdnaWGs*, *akšuvdna+Gen:akšuvdnaWG* ('action'). To the right another transducer produces the pair *vdnaWG : vnna0* . These are composed to give the result *bovdna+Loc:bovnnas,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

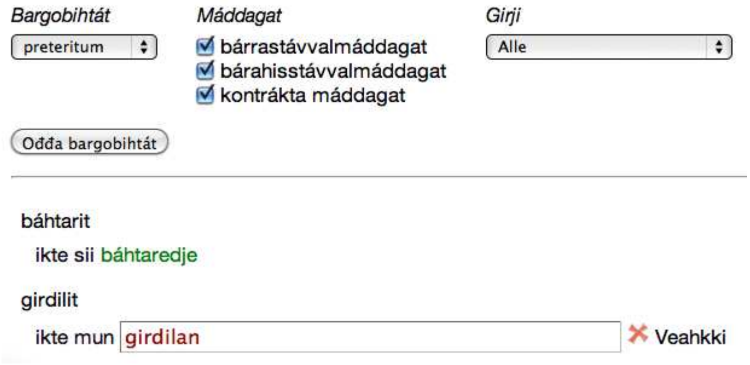
These divisions into tag sets are crucial in the production of morphological exercises.

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 ✗

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.

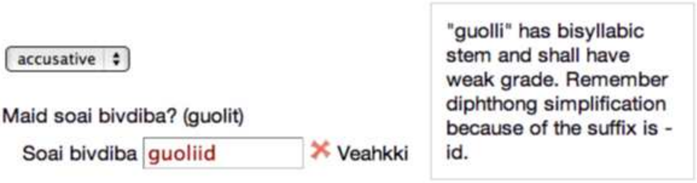
Although it is possible to formulate exercises that make use of more agreement, thus extending simpler question structures to cover a more complex set of sentences, there are some reasons to prefer defining simpler sentence types. First, it is overall a simpler task to produce more exercise definitions 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.



Maid soai bivdiba? (guolit)

Soai bivdiba guoliid ✗

"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.

WordGap - Automatic Generation of Gap-Filling Vocabulary Exercises for Mobile Learning

Susanne Knoop, Sabrina Wilske

University of Bremen
Bremen, Germany

susanne@informatik.uni-bremen.de, wilske@informatik.uni-bremen.de

Abstract

We present a mobile application for learners of English as a second language that instantaneously generates gap filling exercises from a given text. The app provides an opportunity for contextualized vocabulary learning, customized to the learner's interest. Part of the exercise is a multiple choice of the original gap filler plus a set of incorrect distractor items. The key problem to solve in order to automatically generate this type of exercises is the selection of suitable distractor items. For the implementation of the application, we employ strategies proposed in previous work, making use of freely available tools and resources.

Keywords: Vocabulary, ESL, NLP, CALL, cloze exercises, gap filling, Android, informal language learning, mobile learning.

Susanne Knoop and Sabrina Wilske 2013. WordGap – Automatic generation of gap-filling vocabulary exercises for mobile learning. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 39–47.

1 Introduction

In recent years, the rising popularity of smartphones has led to an increase in mobile applications for vocabulary learning. Most of these applications help the user to memorize predefined word lists with the help of digital index cards. However, index cards present a word in an isolated way that does not reflect the incremental and complex process of vocabulary acquisition. Several authors therefore recommend learning from context through extensive reading as the most effective way of vocabulary acquisition (Nerbonne, 2002; Nation and Waring, 1997; Oxford and Scarcella, 1994; Nagy et al., 1985).

A widely used method to test and train word knowledge in context is the "cloze exercise". It consists of a text in which words have been replaced by a gap that has to be filled by the learner (Lee, 2008; Soudek and Soudek, 1983). In the multiple-choice version of the cloze exercise the target word is presented together with several incorrect candidates, called distractors. The quality of a multiple choice cloze exercise is highly dependent on the quality of the distractors. Good distractors should not be totally unlikely, but at the same time not too similar to the target response. If they are too implausible, the exercise would be too easy; if they are too similar, the exercise would not have a clear, unambiguous target response.

In particular if cloze tests are used for language testing, the quality of the distractors is related to (a) their capacity to distract from the correct answer and (b) their ability to discern between learners of different proficiency levels (Goodrich, 1977). Based on an empirical study with Arab learners of English Goodrich recommends words from the context of the text, antonyms, and false synonyms (i.e., synonyms that can not replace the target word in this specific context) as effective distractors for English cloze exercises.

Since it usually requires expert knowledge to select suitable distractors, the creation of individual exercises, based on up-to-date material is expensive and time-consuming (Sumita et al., 2005). In order to avoid the cost of the creation of suitable cloze exercises, there have been several attempts in recent years to automatically generate good distractors using methods from natural language processing (NLP) (Coniam, 1997; Brown et al., 2005; Sumita et al., 2005).

Using the insights of this research and readily available NLP tools, we have built a mobile application for Android smartphones that creates cloze exercises on the fly, based on texts selected by the learner. The application thus provides highly individualized exercises to support contextualized mobile vocabulary learning. Customized reading material that matches the interests of the learner increases the motivation to learn and facilitates learning progress (Heilman et al., 2010; Goto et al., 2010).

In the remainder of the paper we first summarize the strategies used in previous work for selecting suitable distractors (Section 2). We then give an overview of the application in Section 3. Section 4 describes processing steps of the application and the NLP tools that we used for the implementation. In Section 5 we give an outlook on possible further development of the application.

2 Related Work

In this section we describe previous work on generating multiple-choice cloze exercises. As we have discussed above, the challenge in multiple-choice exercises is to select appropriate distractors to the correct target response. The strategies that have been shown to be successful make use of parts of speech, frequency, and distribution of words in the text to select distractors.

One of the first approaches is described by Coniam (1997), who developed a system for the automatic generation of multiple-choice cloze exercises to assist ESL teachers of secondary schools with the preparation of tests. The program chooses distractors with the same part of speech and a similar frequency in the Bank of English Corpus as the target word.

The REAP system, implemented by Brown et al. (2005), develops an individual learner model for each user that encompasses the user's vocabulary and personal interests and chooses a reading text with a distribution of 95% known vocabulary and 5% unknown vocabulary. The knowledge of the newly learned 5% words is then trained and tested through automatically generated vocabulary tests. Afterwards, the program updates the learner model and chooses a new text. The distractors of the automatic multiple choice cloze tests have the same part of speech tag and a similar frequency as the target word and distractors that appear in the original text are preferred. The performance of learners in the automatically generated tests correlates strongly with their performance in manually created tests as well as with the results of standard vocabulary tests.

Sumita et al. (2005) use a corpus of manually created cloze exercises to determine the features of adequate target words, such as their position in the sentence and their part of speech. Distractors are synonyms of the target word that are verified with a Google search: If the sentence together with a distractor candidate yields results, it can be assumed that in this sentence, the distractor could be a valid replacement of the target word and is therefore deleted from the list. Again, the results of the automatically generated tests correlate strongly with the results of the internationally recognized TOEIC (Test of English for International Communication).

Of these examples, Sumita et al. (2005) come closest to the application we present. Like our approach, exercises can be generated based on individually selected web pages. There is also a mobile interface which allows users to download existing exercises stored on a server to a mobile phone. Unlike our application, which focuses on informal, self-paced learning, the system described by Sumita et al. is targeted primarily on language proficiency testing.

3 WordGap: Automatic cloze exercises for smartphone users

In the following, we describe WordGap, an application for the Android platform that allows learners of English to test and train their word knowledge with cloze exercises from any text file or website. The source code of the app and of the server component was published under a GNU General Public License (GPL).[1]

The target group for WordGap are adult and advanced learners of English as a second language who seek to extend their vocabulary by reading texts of personal interest, for example novels, newspaper articles or blog entries. In the context of the mobile application, even short waiting times of 5 to 10 minutes can be used for a short exercise.

The application is of best use to advanced learners because a certain amount of vocabulary has to be known to allow learning new vocabulary from context. Nation and Waring (1997) cite the number of 3,000 words that cover 95% of a text as most efficient for contextual

[1] https://github.com/wordgap/wordgap

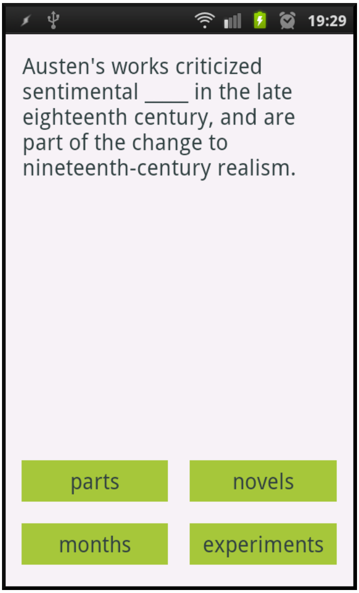

Austen's works criticized sentimental ____ in the late eighteenth century, and are part of the change to nineteenth-century realism.

Figure 1: Screenshot of a cloze exercise with the WordGap app

learning. Less advanced learners, however, could use simplified texts, such as children's and youth literature. Unknown vocabulary can only be guessed if it is sufficiently common, therefore, WordGap is not intendend for the acquisition of terminology, entity names or extremely rare words.

The chosen text can be loaded from a text file saved on the smartphone or from a website. In the latter case, the app will load the website's text when the user selects the app on the "Sent-to" menu of the smartphone's browser. Exercises can be generated for four different parts of speech: nouns, verbs, adjectives or prepositions. The app sends the text and the chosen part of speech to a server that generates the exercise for the learner to carry out on their phone.

WordGap displays the sentences of the exercise sequentially together with the target word and three distractors in random order (see Figure 1 for a screen shot). The user has to choose the correct answer by tapping on it. Correct and incorrect choices are logged for the user's performance statistics that will be displayed after completing or aborting the exercise.

During the exercise, unknown target words can be added to a list and after finishing the exercise, the WordNet definitions of the unknown words can be obtained from the server. This delayed access to the word definitions is motivated by the Depth-of-Processing hypothesis: Guessing the meaning of an unknown word from the context requires a deeper semantic processing than simply looking it up in the dictionary and is therefore supposed to ease long-term memorization (Nerbonne, 2002; Oxford and Scarcella, 1994; Segler et al., 2002). All exercises are saved automatically on the phone's local memory so that they can be repeated at any time even without network connection.

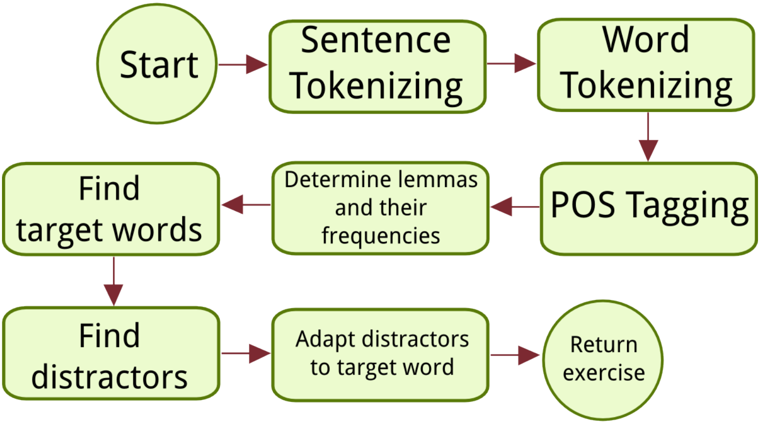


Figure 2: The NLP processing pipeline of the WordGap server

4 System architecture and NLP tools

This section describes the processing steps of the system and presents the tools and resources that we applied.

The implementation consists of a client application that runs on Android devices and a server implementation that can run on any machine. The server implemenation is based on the Django Python web framework[2]. The server and the clients communicate through the JSON data interchange format[3].

When creating an exercise, the WordGap server processes the text according to the pipeline demonstrated in Figure 2. After sentence and word tokenizing, the text is part-of-speech tagged. The tokens tagged with the part of speech chosen by the user (i.e., nouns, verbs, adjectives or prepositions) are then transformed into their base form (i. e., infinitive for verbs and singular for nouns) to determine their frequency. For each sentence, the program attempts to find a target word of the given part of speech tag. If there are multiple candidates in one sentence, the target word is chosen based on its frequency in the text – the more frequent a word, the more likely it is to be selected by a weighted random choice function.

For each target word, adequate distractors have to be determined. These can be words with the same part of speech from the text, or antonyms and false synonyms of the target word. In a post-processing step, they are adapted to the target word in their grammatical form and capitalization to avoid giving the user additional hints about the correct choice.

The following subsections describe how several Natural Language Processing resources support the process of exercise generation.

4.1 WordNet

WordNet is a semantic network that was developed by George Miller and Christiane Fellbaum to study the vocabulary acquisition of infants (Miller and Fellbaum, 2007). It organizes the

[2] www.djangoproject.com
[3] www.json.org

nouns, adjectives, adverbs and verbs of the English language into so-called “synsets” that contain lemmas that can be synonyms in certain contexts. Nouns and verbs are also ordered in a hierarchy of hypernyms and hyponyms.

WordGap uses WordNet to find distractors that are antonyms or false synonyms of the target word. For the latter, we implemented two possibilities: Words with the same hypernym as the target word or synonyms of the synonyms.

A visual and informal inspection of the distractors showed that those taken directly from the text tend to be more useful than the distractors derived from WordNet. One reason for that is that WordNet often lacks entries for antonyms and synonyms that a thesaurus would contain. Besides, many synsets contain rare words that will seem out of place because they do not fit to the text’s genre. The user could in this case just guess the right word by choosing the only word that seems familiar. Because of this shortcoming, the user is offered the option to use or not use WordNet when creating the exercise.

4.2 The Natural Language Tool Kit (NTLK)

The Natural Language Tool Kit (NTLK) is an extensive open-source library for the programming language Python that was first developed by Edward Loper and Ewan Klein at the University of Pennsylvania (Perkins, 2010; Bird et al., 2009). It contains WordNet as well as numerous corpora and dictionaries in different languages.

The WordGap server uses methods of the NLTK for sentence and word tokenizing, as well as part of speech tagging. For the latter, NLTK’s implementation of the Naive Bayes Tagger was trained on the Brown corpus. For the WordGap application we want to value precision over recall because an incorrectly tagged target word would lead to inadequate distractors. Therefore, we trained the Naive Bayes tagger with a cut-off probability of 95%, which means that no token will be tagged with a tag that has a probability of less than 95%.

4.3 Nodebox Linguistics

NodeBox Linguistics[4] is a collection of different open-source libraries for Python. First of all, it contains a more convenient interface for accessing WordNet than the NLTK does. More importantly, it provides methods for generating different grammatical forms of English lemmas. Thus it offers singular and plural forms of nouns and different tenses for verbs. WordGap uses these methods to adapt the distractors to the target word and make their number and/or tense match.

5 Conclusion and Outlook

We have presented an application for smartphones that generates instantaneous cloze exercises based on texts chosen by the user and thus provides contextualized and individualized vocabulary learning. We have shown that it is possible to create such an application with readily available NLP tools and resources.

One possible extension of WordGap would be the adaption to a target language other than English. This would require that the following resources are available in that language: A software library for sentence tokenizing, word tokenizing and part of speech tagging, a

[4] `nodebox.net/code/index.php/Linguistics`

semantic network like WordNet or a thesaurus with synonyms and antonyms, a database or software library to retrieve different grammatical forms of nouns, verbs or adjectives and a source of definitions or translations of unkown target words.

Another possible extension to the app would be multiple choice cloze exercises that focus on grammatical knowledge, similar to the work described by Meurers et al. (2010). For instance, for learning verb tenses, the distractors could be different tense forms of the same verb. For learning the use of articles, the learner would have to choose from a list of definite, indefinite and no article.

So far, the application has only been tested in terms of usability as part of the development process to identify usability issues. In future work, we would like to evaluate the app in terms of the learning gains that it enables. We also have not yet conducted a thorough assessment of the quality of the generated exercises.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Brown, J., Frishkoff, G. A., and Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *HLT/EMNLP*. The Association for Computational Linguistics.

Coniam, D. (1997). A Preliminary Inquiry Into Using Corpus Word Frequency Data in the Automatic Generation of English Language Cloze Tests. *CALICO Journal*, 14(2-4):15–33.

Goodrich, H. C. (1977). Distractor efficiency in foreign language testing. *TESOL Quarterly*, 11(1):pp. 69–78.

Goto, T., Kojiri, T., Watanabe, T., Iwata, T., and Yamada, T. (2010). Automatic generation system of multiple-choice cloze questions and its evaluation. *Knowledge Management & E-Learning : an International Journal*.

Heilman, M., Collins-Thompson, K., Callan, J., Eskenazi, M., Juffs, A., and Wilson, L. (2010). Personalization of reading passages improves vocabulary acquisition. *Int. J. Artif. Intell. Ed.*, 20(1):73–98.

Lee, S. H. (2008). Beyond reading and proficiency assessment: The rational cloze procedure as stimulus for integrated reading, writing, and vocabulary instruction and teacher–student interaction in esl. *System*, 36(4):642 – 660.

Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V., Ott, N., and Tübingen, U. (2010). Enhancing authentic web pages for language learners.

Miller, G. A. and Fellbaum, C. (2007). Wordnet then and now. *Language Resources and Evaluation*, 41(2):pp. 209–214.

Nagy, W. E., Herman, P. A., and Anderson, R. C. (1985). Learning Words from Context. *Reading Research Quarterly*, 20(2):233–253.

Nation, P. and Waring, R. (1997). Vocabulary size, text coverage and word lists. In *Vocabulary: Description, Acquisition and Pedagogy*, pages 6–19. University Press.

Nerbonne, J. (2002). Computer-assisted language learning and natural language processing. In *Handbook of Computational Linguistics*, pages 670–698. University Press.

Oxford, R. and Scarcella, R. C. (1994). Second language vocabulary learning among adults: State of the art in vocabulary instruction. *System*, 22(2):231–243.

Perkins, J. (2010). *Python Text Processing with NLTK 2.0 Cookbook : Over 80 Practical Recipes for Using Python's NLTK Suite of Libraries to Maximize Your Natural Language Processing Capabilities*. Packt, Birmingham, UK.

Segler, T. M., Pain, H., and Sorace, A. (2002). Second Language Vocabulary Acquisition and Learning Strategies in ICALL Environments. *Computer Assisted Language Learning*, 15(4):409–422.

Soudek, M. and Soudek, L. I. (1983). Cloze after thirty years: new uses in language teaching. *ELT Journal*, 37(4):335–340.

Sumita, E., Sugaya, F., and Yamamoto, S. (2005). Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, EdAppsNLP 05, pages 61–68, Stroudsburg, PA, USA. Association for Computational Linguistics.

Towards a gold standard for Swedish CEFR-based ICALL

Elena Volodina[1], Dijana Pijetlovic[1], Ildiko Pilán[1], Sofie Johansson Kokkinakis[1]

(1) Språkbanken, University of Gothenburg, Box 200, 405 30 Göteborg, Sweden

elena.volodina@svenska.gu.se, guspijdi@student.gu.se, ildiko.pilan@gmail.com, sofie.johansson.kokkinakis@svenska.gu.se

Abstract

In qualitative projects on ICALL (Intelligent Computer-Assisted Language Learning), research and development always go hand in hand: development both depends upon the research results and dictates the research agenda. Likewise, in the development of the Swedish ICALL platform *Lärka*, the practical issues of development have dictated its research agenda. With NLP approaches, sooner or later, the necessity for reliable training data becomes unavoidable. At the moment Lärka's research agenda cannot be addressed without access to reliable training data, so-called “gold standard”. This paper gives an overview of the current state of the Swedish ICALL platform development and related research agenda, and describes the first attempts to collect the reference corpus (“gold standard”) coming from course books used in CEFR-based language teaching.

Keywords: ICALL, CEFR, exercise generator, course book corpus compilation

Elena Volodina, Dijana Pijetlović, Ildikó Pilán and Sofie Johansson Kokkinakis 2013. Towards a gold standard for Swedish CEFR-based ICALL. *Proceedings of the second workshop on NLP for computer-assisted language learning at NODALIDA 2013.* NEALT Proceedings Series 17 / Linköping Electronic Conference Proceedings 86: 48–65.

1 Background

The ICALL platform *Lärka* described in this paper is an open-source web-based application that uses principles of Service-Oriented Architecture (Volodina et al., 2012a; Volodina & Borin, 2012). The platform is divided into several modules: an exercise generator with activities for university students of linguistics and second/foreign language (L2) learners; and modules facilitating different aspects of development and research, at the moment consisting of an experimental sentence readability module and an editor for learner-oriented corpora.

The main focus of Lärka is on L2 learners. This sets certain requirements, first of all, on the use of a pedagogical framework. Among different pedagogical theories and approaches, the Common European Framework of Reference for Languages (CEFR) is one of the most influential. CEFR is a document containing guidelines for harmonization of language teaching and assessment across languages and countries (Council of Europe, 2001). It provides a common metalanguage to talk about objectives, assessment and proficiency levels. Further, it offers a descriptive scheme that can help analyze learner’s needs, target communicative competences and define the course curriculum. It is useful for tracking learner progress as well as for designing assessment tests and assigning proficiency levels (Little, 2007, 2011; North, 2007). CEFR defines language competences and skills through “can do” statements at six proficiency levels (A1, A2, B1, B2, C1, C2) which offer flexibility in interpreting them for different languages and target groups. Since the publication of the CEFR guidelines in 2001, a number of countries including Sweden have adopted the system and reorganized language teaching and testing practices to fit into this framework.

Other existent proficiency scales for Swedish language learning include the ones used in SFI (Swedish for immigrants) and SVA (Swedish as a Second Language), both aligned to fit into the CEFR paradigm. SFI, containing levels A, B, C, D correspond to CEFR's A1-/A1, A1/A2, A2/A2+, B1/B1+ respectively according to the recommendations provided by the Swedish National Agency for Education (Skolverket). The language proficiency scale used for SVA, is said to be roughly equivalent to the CEFR level C1 when sva B is reached. Since the CEFR scale combines all the extremes of development of Swedish as L2, and offers interoperability across different countries, we have chosen this scale for our platform.

Ideally, the use of CEFR scales in the context of an ICALL platform should offer a clear-cut possibility to generate exercises and materials adjusted to the proficiency levels. It is, however, a non-trivial task to apply the CEFR descriptors to the practical task of automatic selection of language samples appropriate for different proficiency levels. CEFR’s flexibility, being a positive feature on the one hand, has a reverse side. As a number of Second Language Acquisition (SLA) researchers have mentioned, it is non-specific and therefore it is difficult to associate the different kinds of competences and levels of accuracy that learners would need in order to perform language learning tasks with different CEFR levels (Westhoff, 2007). Milton (2009) says that the lack of objectivity in the CEFR descriptors makes it possible that learners with different amounts and kinds of knowledge can be placed into the same CEFR level; or that performance outweighs competence so that competent but insecure performers can be assigned to a lower CEFR level than they deserve. Among other things, insufficient specifications for vocabulary and grammar competence have been pointed out by Byrnes (2007); Milton (2009); Westhoff (2007); Little (2007, 2011).

Special efforts have been undertaken to interpret CEFR guidelines as sets of Reference Level Descriptions (http://www.coe.int/t/dg4/linguistic/dnr_en.asp) as well as to establish procedures to relate language exams to the CEFR (Council of Europe, 2009), but to the best of our knowledge that has not been done yet for Swedish. Attempts at *aligning texts and tests with CEFR* for a number of other languages are ongoing (e.g. Khalifa et al., 2010; Szabó, 2010; Dávid, 2010) with what could be called a *top-down approach*, i.e. starting from CEFR descriptors and going all the way down to the actual selection of appropriate language samples. We suggest a *bottom-up approach*, where we start from the actual language samples labeled by experienced teachers or coursebook writers for levels, analyze them for linguistic constituents with the help of machine learning approaches and then try to map the identified constituents to the CEFR descriptors. The two approaches should be viewed as complementary of each other.

This is the starting point for our “quest” for data collection, designed to help us interpret CEFR descriptors in a way that can facilitate automatic methods in L2 material generation, among other things: to identify receptive vocabulary scope per level, and to adjust algorithms for sentence readability per proficiency level. Both aspects are described in detail in the following section.

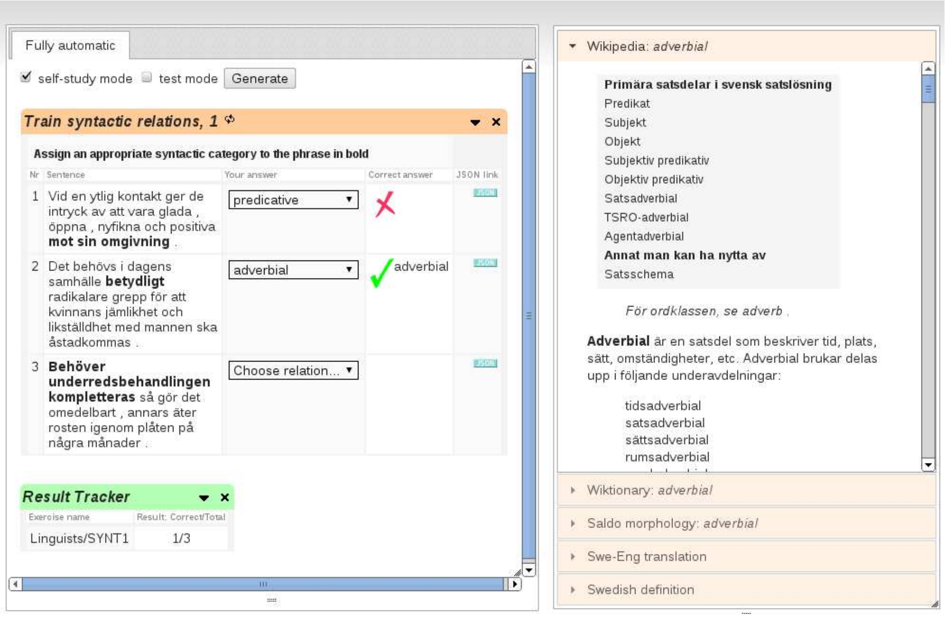
The paper is structured as follows: section 2 reports on the current state of development where the lack of exact interpretation of CEFR scales into linguistic constituents for Swedish has so far hindered implementation of desired exercises or their adjustment to learner proficiency levels. Section 3 describes the compilation of a corpus of CEFR-related course book texts as a way to cope with that obstacle. Section 4 concludes the paper.

2. Current state - in need of a gold standard

Use of NLP for language learning tasks has been pursued in different studies (e.g. Amaral and Meurers, 2011; Amaral et al., 2011; Heift, 2003; Nagata, 2009). Most of the implemented applications generate learning materials, tasks or feedback customized to user interests, needs and proficiency levels. However, the question of automatic classification of authentic language samples (e.g. texts or sentences) into proficiency levels is not always directly addressed. In Meurers et al. (2010) and Knoop & Wilske (2013), the user finds the texts on the web him-/herself, and the exercise is generated on the basis of that text. In Toole & Heift (2002) this issue is solved indirectly through teachers feeding in sample texts containing examples of learning objective. In Aldabe et al (2006) this issue is ignored and only questions for “high language level” are generated. The question of text classification into levels is directly approached in REAP and Choosito applications (Collins-Thompson & Callan, 2007; Heilman et al., 2007; Francois & Miltsakaki, 2012), elaborating on two major factors: vocabulary frequency and a readability measure based on a selection of linguistic parameters.

2.1 Module for university students of Linguistics

An exercise generator for linguists comes with two exercises: training syntactic relations and training parts of speech (Figure 1).



Fully automatic

☑ self-study mode □ test mode Generate

Train syntactic relations, 1

Assign an appropriate syntactic category to the phrase in bold

Nr	Sentence	Your answer	Correct answer	JSON link
1	Vid en ytlig kontakt ger de intryck av att vara glada , öppna , nyfikna och positiva **mot sin omgivning** .	predicative ▼	✗	JSON
2	Det behövs i dagens samhälle **betydligt** radikalare grepp för att kvinnans jämlikhet och likställdhet med mannen ska åstadkommas .	adverbial ▼	✓ adverbial	JSON
3	**Behöver underredsbehandlingen kompletteras** så gör det omedelbart , annars äter rosten igenom plåten på några månader .	Choose relation... ▼		JSON

Result Tracker

Exercise name	Result: Correct/Total
Linguists/SYNT1	1/3

Wikipedia: *adverbial*

Primära satsdelar i svensk satslösning
Predikat
Subjekt
Objekt
Subjektiv predikativ
Objektiv predikativ
Satsadverbial
TSRO-adverbial
Agentadverbial
Annat man kan ha nytta av
Satsschema

För ordklassen, se adverb .

Adverbial är en satsdel som beskriver tid, plats, sätt, omständigheter, etc. Adverbial brukar delas upp i följande underavdelningar:

tidsadverbial
satsadverbial
sättsadverbial
rumsadverbial

Wiktionary: *adverbial*

Saldo morphology: *adverbial*

Swe-Eng translation

Swedish definition

FIGURE 1. EXAMPLE OF AN ITEM FOR TRAINING SYNTACTIC RELATIONS. INTENDED USERS: LINGUISTS

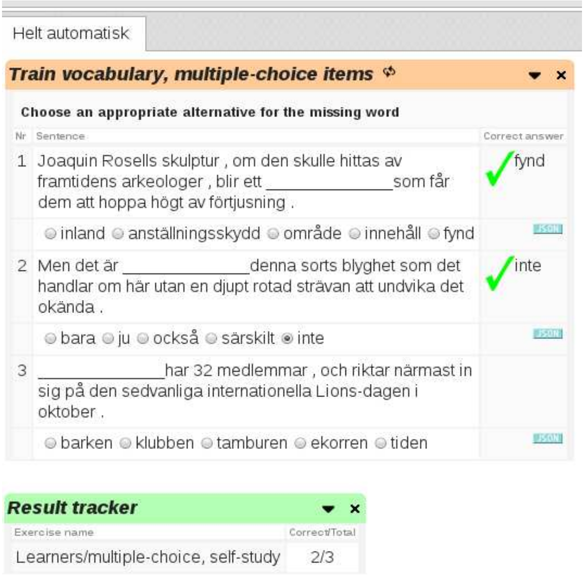
Both exercises use multiple-choice model and are based on sentences randomly selected from several manually checked corpora of Swedish: Stockholm Umeå Corpus (Källgren et.al., 2006), Talbanken (Teleman, 1974; Einarsson, 1976; Nivre et al., 2006) and Läsbart (Heimann Mühlenbock, 2013). The user is offered support in the form of Wikipedia and lexicon entries, as well as feedback in the form of correct-incorrect answers and a result tracker. Once the item is answered, another one is generated.

The system has been tested in real-life setting with students of Linguistics and the first feedback has revealed the general acceptance of the exercises.

However, teachers have expressed their reservation against the use of Wikipedia instead of reference sources of higher quality/reliability. Among other desired improvements a better sentence selection has been mentioned. “Better” sentences should be understood as non-eliptic well-formed simple sentences (as opposed to complex ones). The problem of selection of “appropriate” sentences is described under “Sentence readability” below.

2.2 Multiple-choice vocabulary items for L2 learners

An exercise generator for language learners comprises at the moment multiple choice exercise items for vocabulary training, see Figure 2.



Train vocabulary, multiple-choice items

Choose an appropriate alternative for the missing word

Nr	Sentence	Correct answer
1	Joaquin Rosells skulptur , om den skulle hittas av framtidens arkeologer , blir ett ______________ som får dem att hoppa högt av förtjusning .	✓ fynd
	○ inland ○ anställningsskydd ○ område ○ innehåll ○ fynd	JSON
2	Men det är ______________ denna sorts blyghet som det handlar om här utan en djupt rotad strävan att undvika det okända .	✓ inte
	○ bara ○ ju ○ också ○ särskilt ● inte	JSON
3	______________ har 32 medlemmar , och riktar närmast in sig på den sedvanliga internationella Lions-dagen i oktober .	
	○ barken ○ klubben ○ tamburen ○ ekorren ○ tiden	JSON

Result tracker

Exercise name	Correct/Total
Learners/multiple-choice, self-study	2/3

FIGURE 2. MULTIPLE-CHOICE ITEMS FOR LANGUAGE LEARNERS

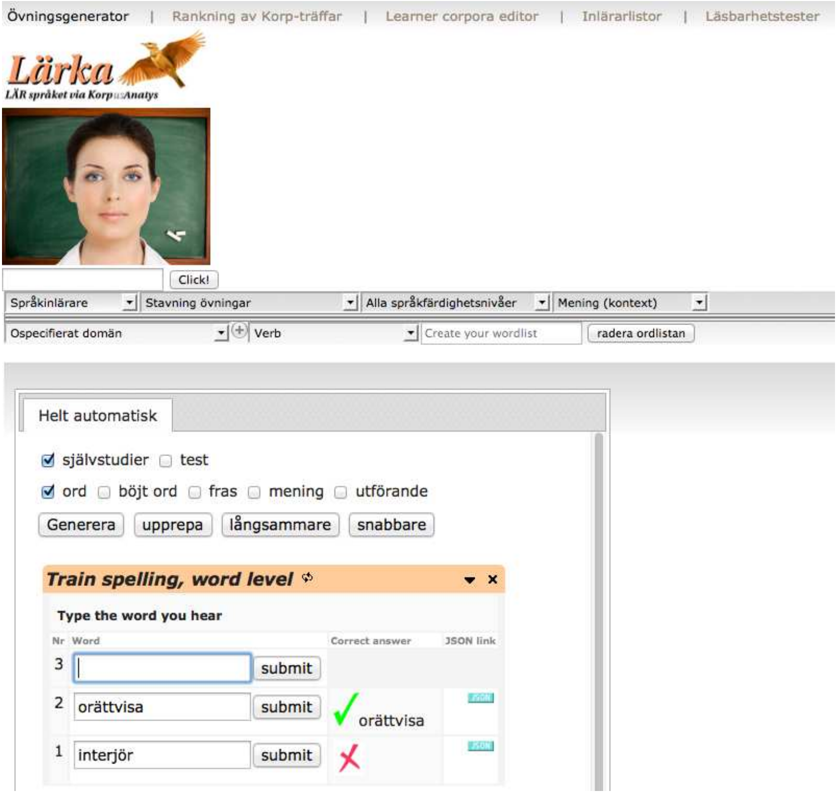
The target vocabulary for training is at the moment selected randomly from the Swedish Kelly list (Volodina & Johansson Kokkinakis, 2012), a frequency-based vocabulary list for language learners. A sentence containing the target vocabulary is then randomly selected from SUC (Stockholm Umeå Corpus, Källgren et.al., 2006) guided by the principle of maximum sentence length limited to 15 tokens. Distractors to the correct answer are selected based on the principle of the shared frequency band with the correct answer, the same part of speech and shared morpho-syntactic tag.

However, to generate exercise items appropriate at different learner proficiency levels, selection of target vocabulary should be aligned with the CEFR levels. The latter means the need to study the vocabulary used in the CEFR-based courses, both receptively in course books and productively in written essays, per proficiency level. Addressing this problem without reference data labelled for CEFR levels is however impossible.

Another problem arising in connection with vocabulary training is the appropriateness of the language samples where the target item is used in its context. For copyright reasons, the usual context in Lärka is limited to sentences. Selection of appropriate sentences for language training at different proficiency levels needs a reliable method to classify available sentences by CEFR levels. This, in turn, cannot be studied without an extensive collection of appropriate sentences labelled for proficiency levels, which again points to the need of a corpus of CEFR-related texts.

2.3 Dictation and spelling items for L2 learners

The dictation and spelling items have been recently implemented, but the development is still in progress (Pijetlovic & Volodina, forthcoming).


Övningsgenerator | Rankning av Korp-träffar | Learner corpora editor | Inlärarlistor | Läsbarhetstester
Lärka
LÄR språket via KorpusAnalys
Click!
Generera
upprepa
långsammare
snabbare
submit

Språkinlärare | Stavning övningar | Alla språkfärdighetsnivåer | Mening (kontext)

Ospecifierat domän | Verb | Create your wordlist | radera ordlistan

Helt automatisk

☑ självstudier □ test

☑ ord □ böjt ord □ fras □ mening □ utförande

Train spelling, word level

Type the word you hear

Nr	Word	Correct answer	JSON link
3			
2	orättvisa	✓ orättvisa	JSON
1	interjör	✗	JSON

FIGURE 3. DICTATION AND SPELLING ITEM

The goal of this module is to offer web services for automatic generation of spelling exercises using Text-To-Speech technology for Swedish, thus facilitating training of listening and spelling competences. The exercise is planned to be “adaptive” in the sense that once the users are confident with spelling single words, they are offered the target word in inflected forms, in phrases, and finally in sentences (Figure 3).

Spelling errors can be distinguished between performance-based and competence-based. To account for a more fine-grained distinction between errors, a collection of real-life spelling mistakes needs to be consulted in order to give a useful feedback to the user. Due to the lack of Swedish spelling error corpora, one part of this module involves collecting spelling errors through online dictation&typing exercises with both Swedish native and non-native speakers.

The success of this exercise type depends upon the two factors mentioned before: selection of vocabulary and sentences appropriate for learner level.

2.4 Current research agenda

From the short description above, it is clear that the immediate research agenda contains, among other things, (1) the issue of identifying receptive vocabulary scope per proficiency level and (2) the issue of finding a reliable algorithm for sentence readability assessment. Both issues depend on the availability of reference data, which we are now actively collecting.

2.4.1 Receptive vocabulary scope

According to the CEFR document, there are four main sources of vocabulary that potentially can constitute the vocabulary scope of a CEFR-based course, namely: (1) words typical for the topics required for the learners’ communication (domain-specific vocabulary); (2) vocabulary that is based on lexical-statistical principles of selection (highest frequency words); (3) words randomly coming from texts that are selected as learning material by teachers, and finally (4) words learnt in response to the communicative needs that arise. (Council of Europe, 2001:150-151) The users of CEFR guidelines are encouraged to define *what specific/particular lexical elements the learner might need and how they have been selected.*

To identify the scope of receptive vocabulary for exercise generation needs, we intend to collect a frequency-based vocabulary list from the CEFR-related texts labelled for levels. The lists will be ordered by lemmas and their parts-of-speech as a unique unit in the list. Previous attempts at generating learner-oriented frequency-based word lists have been made in the Kelly project (2009-2011, http://www.kellyproject.eu/), an EU-funded project on building learner-oriented frequency-based monolingual and bilingual word lists for 9 languages intended to be used in a commercial language learning tool (Volodina & Johansson Kokkinakis 2012; Kilgarriff et al., forthcoming; keewords.com). In the Kelly project, target vocabulary has been collected from a large web-corpus of written language used on the web. The basis of the Kelly list is the general-purpose vocabulary, providing the range of both lexical and grammatical elements as specified in the CEFR (Council of

Europe, 2001:110-111). However, during the post-Kelly period we have observed the need for additional modifications: (1) the list needs to be validated against the reading materials used in the CEFR-based courses, to make sure that vocabulary in the list is correctly streamed into CEFR levels; (2) we need to fill in the gaps in relevant vocabulary, for example, missing lexical items like “toothpaste”, “toothbrush”, etc. that clearly need to be present in the learner-oriented vocabulary lists, but do not gain any prominent place in the frequency lists generated from written native speaker corpora. We thus need to analyze which vocabulary should be added, removed or relocated in the list with regard to the CEFR guidelines based on the evidence of materials used in the real-life CEFR-based courses; and (3) we need to look specifically into the domain-specific vocabulary according to the CEFR themes – which words, which levels, how many per level – and evaluate if domain vocabulary should be included into the Kelly list or should be available as a “satellite list” following the implicit indications in the CEFR (Council of Europe, 2001:52-53).

The suggested approach will help us identify (concrete) lexical curriculum for CEFR-based courses in Swedish, both in terms of *what* words and *how many* per level a student of each level should acquire. The resulting list will be used primarily as an instrument for training vocabulary in the Lärka-based exercise generator. Apart from this, the list can be used for testing authentic examples (e.g. texts and sentences) for appropriateness for learners of different proficiency levels; for assessment of language proficiency in L2 learner language production, etc. The crucial prerequisite for this sub-project is access to an *annotated corpus* containing texts labeled for proficiency levels, the gold standard described in the next session.

2.4.2 Sentence readability

The degree to which a text can be understood by a human reader is referred to as its *readability* (Kate et al., 2010). In the second language context, readability corresponds to the extent to which learners are able to understand a text at a certain proficiency level.

Texts and sentences can be mapped to a corresponding level with the use of a measure based on statistical information about different linguistic properties of a text. Traditional readability measures, such as Flesh-Kincaid, Dale-Chall etc. for English, and LIX (Läsbarthetsindex) for Swedish, however, are limited to surface text features such as sentence length and the number of syllables (Heilman et al., 2007; Heimann Mühlenbock, 2013). Moreover, they consider text readability from a first language point of view and focus at the text level, and thus have shortcomings when used on very short passages (Kilgarriff et al., 2008) or when applied to L2 contexts (Beinborn et al., 2012).

Second language teachers and writers of teaching materials often need to make human judgments about readability at both text and sentence level, but recent NLP research started to explore automated techniques for this task, which combine syntactic and lexical information with *machine learning methods*. An important first step in most machine learning-based readability methods is a sufficient amount of *annotated training data* containing texts labeled with a corresponding level. Then, a number of features, i.e. information and characteristics of the text that one wishes to take into consideration, should be selected (Collins-Thompson & Callan, 2005; Tanaka-Ishii et al., 2010). Finally, these features need to be mapped to a readability level (or score) with a machine learning algorithm. Hybrid approaches combining rule-based, statistic and machine learning methods are also explored in the area of text readability in L2 context (François & Miltsakaki, 2012).

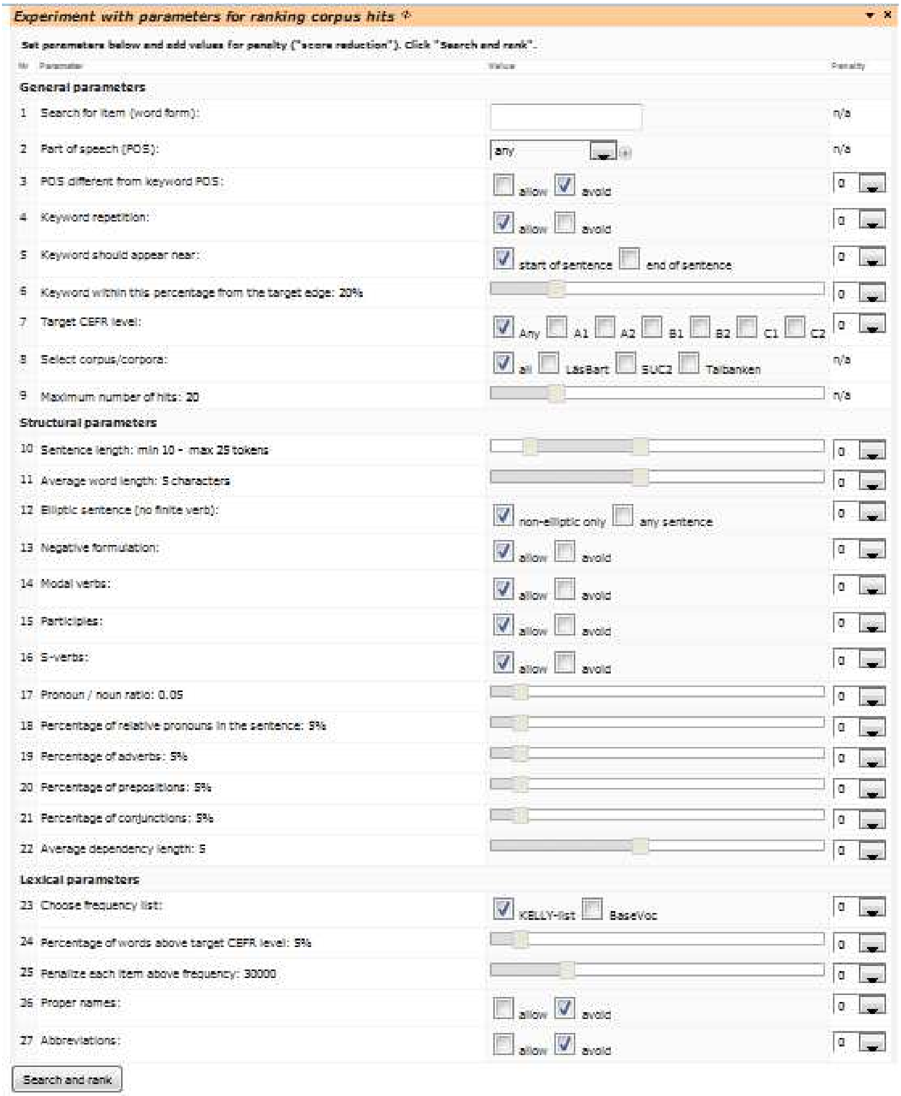
The sentence readability project for Swedish is currently under development (Volodina et al., 2012; Pilán et al., forthcoming). It has arisen in response to the need for a reliable algorithm for classification of sentences into appropriate CEFR-levels in Lärka context.

This project was initially focused on general ranking of corpus hits according to their “appropriateness” (Volodina et al., 2012b). The aim has gradually evolved and eventually crystallized into finding an NLP-based algorithm to predict which lexical, morpho-syntactic and possibly other linguistic elements which students are able to understand at a certain language learning level (Pilán et al., forthcoming).

This project builds upon experimenting with both manually weighted heuristic rules, as well as with machine learning techniques. During the selection of parameters and features not only superficial readability criteria such as sentence and word lengths are taken into consideration, but also deeper linguistic aspects from a second language teaching perspective (part-of-speech, depth of dependencies etc.). The manually set parameters are tested with different thresholds and weights until optimized for a certain CEFR level (see Figure 4). However, to know that the parameter setting is optimal, we need access to experienced teachers who can assess the result (a kind of crowdsourcing), or an open-source collection of sentences labelled for levels to test the prediction accuracy of heuristic rules.

The machine learning part involves supervised techniques to classify the difficulty level of sentences, the training data being a corpus based on second language teaching materials, labeled with CEFR levels, currently available only for B1 and B2 levels. Depending on the outcome of the experiments and users’ preferences, the sentence retrieval process could be fully automatic (based only on the trained model), semi-automatic (with a combination of manual parameters and the trained model) or only manual, so that selection of sentences can be fully customized according to specific needs of teachers and students.

The collection of texts labelled for CEFR levels provides, thereby, a number of opportunities to solve the challenges we face. Moreover, the availability of the training data in question labelled for additional text variables, i.e. not only for CEFR levels but also for topics, genres, etc. can facilitate other research projects relevant for ICALL, for example automatic selection of appropriate texts for the target proficiency level, automatic retrieval of topical texts, automatic question generation, to name just a few.



Experiment with parameters for ranking corpus hits

Set parameters below and add values for penalty ("score reduction"). Click "Search and rank".

Nr	Parameter	Value	Penalty
	General parameters		
1	Search for item (word form):		n/a
2	Part of speech (POS):	any	n/a
3	POS different from keyword POS:	□ allow ☑ avoid	0
4	Keyword repetition:	☑ allow □ avoid	0
5	Keyword should appear near:	☑ start of sentence □ end of sentence	0
6	Keyword within this percentage from the target edge: 20%		0
7	Target CEFR level:	☑ Any □ A1 □ A2 □ B1 □ B2 □ C1 □ C2	0
8	Select corpus/corpora:	☑ all □ LäsBart □ SUC2 □ Talbanken	n/a
9	Maximum number of hits: 20		n/a
	Structural parameters		
10	Sentence length: min 10 - max 25 tokens		0
11	Average word length: 5 characters		0
12	Elliptic sentence (no finite verb):	☑ non-elliptic only □ any sentence	0
13	Negative formulation:	☑ allow □ avoid	0
14	Modal verbs:	☑ allow □ avoid	0
15	Participles:	☑ allow □ avoid	0
16	S-verbs:	☑ allow □ avoid	0
17	Pronoun / noun ratio: 0.05		0
18	Percentage of relative pronouns in the sentence: 5%		0
19	Percentage of adverbs: 5%		0
20	Percentage of prepositions: 5%		0
21	Percentage of conjunctions: 5%		0
22	Average dependency length: 5		0
	Lexical parameters		
23	Choose frequency list:	☑ KELLY-list □ BaseVoc	0
24	Percentage of words above target CEFR level: 5%		0
25	Penalize each item above frequency: 30000		0
26	Proper names:	□ allow ☑ avoid	0
27	Abbreviations:	□ allow ☑ avoid	0

Figure 4. Linguistic parameters for sentence readability, heuristic rules

3. Towards a corpus of CEFR-related course book texts

It is known to be rather controversial to break down CEFR “can-do” statements into concrete constituents, partly due to the “human factor”. Course material producers and teachers often go by their subjective “expert judgements” and intuitions, not necessarily agreeing with each other.

However, we take it for granted that teachers' interpretations of CEFR guidelines, subjective when taken individually, present an objective ground for generalizations and approximations about language complexity and level-wise content, when taken collectively. Therefore, we assume that, given texts used for CEFR-based courses from different authors and publishers, we can perform empirical evidence-based studies of a number of linguistic aspects expected of learners at different levels, for example vocabulary scope, most common grammar per level, text complexity, sentence complexity. Apart from that, we are interested in studying typical linguistic features for texts of different CEFR-based themes (topical domains).

Texts related to language learning fall into two categories: (1) “input” or normative texts provided by course book writers or selected by teachers; and (2) “output” or learner produced texts showing learner performance at the studied level. While learner output texts (not necessarily linked to CEFR levels, though) have been the object of study in different projects for both Swedish (Johansson Kokkinakis & Magnusson, 2011; Hultman & Westman, 1977; Nyström, 2000; Östlund-Stjärnegårdh, 2002) and other languages (Carlsten, 2012; Hawkins & Buttery, 2009), the study of normative course book texts from L2 perspective is rather rarely pursued (Lindberg & Johansson Kokkinakis, 2007, 2009; François & Miltsakaki, 2012). The main (hypothetical) reason for that is absence of accessible digitized data. In the project described in this section we describer our initial efforts at collecting normative texts to fill in the gap and to form the ground for CEFR-based text research for Swedish.

3.1 Collecting corpus materials

To identify relevant course materials, a number of teachers of CEFR-related courses have been interviewed and the relevant publishers have subsequently been contacted for electronic materials. However, texts in electronic format have proven to be rather difficult to obtain. Of all the contacted publishers only *Liber* has shown understanding and provided files for our research. To tackle the problem of lacking texts, we opted for an optical scanning approach subcontracting the relevant digitizing centre. The total amount of course books in pages is 3187; which corresponds to an estimated corpus size of approximately 3 million tokens.

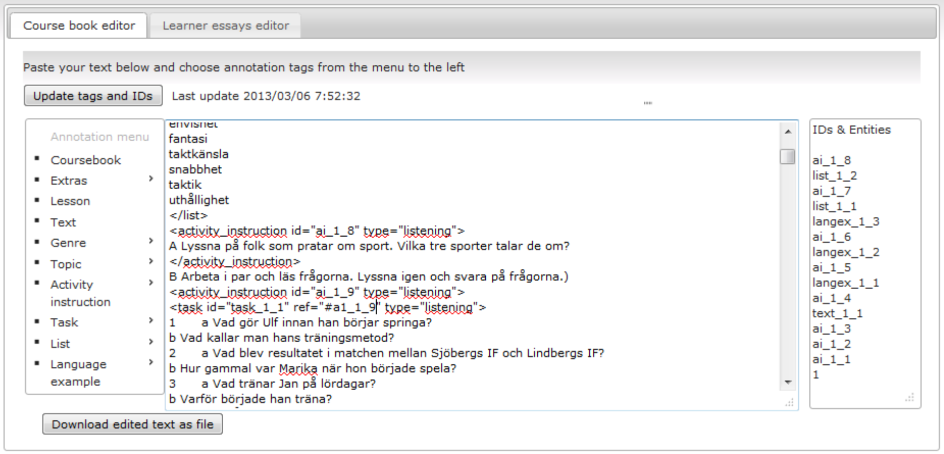
Our pilot level has become B1, with 3 different course books, each containing mixed contents (e.g. half the book B1 level and half the book B2 level; or a part of the book A1/A2, the rest B1), totalling 565 pages.

3.2 Corpus annotation

Annotation of course book texts consists of the following two steps:

1. annotation for CEFR-relevant variables and
2. annotation for linguistic parameters.

We have annotated texts for CEFR-variables using an editor that we developed ourselves. We used Lärka as the basis for the editor. Figure 5 presents the course book editor view: the menu on the left inserts different tags into the text field; the field on the right keeps track of the ids used throughout the file.



Paste your text below and choose annotation tags from the menu to the left

Last update 2013/03/06 7:52:32

```
envishet
fantasi
taktkänsla
snabbhet
taktik
uthållighet
</list>
<activity_instruction id="ai_1_8" type="listening">
A Lyssna på folk som pratar om sport. Vilka tre sporter talar de om?
</activity_instruction>
B Arbeta i par och läs frågorna. Lyssna igen och svara på frågorna.)
<activity_instruction id="ai_1_9" type="listening">
<task id="task_1_1" ref="#a1_1_9" type="listening">
1      a Vad gör Ulf innan han börjar springa?
b Vad kallar man hans träningsmetod?
2      a Vad blev resultatet i matchen mellan Sjöbergs IF och Lindbergs IF?
b Hur gammal var Marika när hon började spela?
3      a Vad tränar Jan på lördagar?
b Varför började han träna?
```

ai_1_8
list_1_2
ai_1_7
list_1_1
langex_1_3
ai_1_6
langex_1_2
ai_1_5
langex_1_1
ai_1_4
text_1_1
ai_1_3
ai_1_2
ai_1_1
1

Figure 5. Course book editor developed for this project

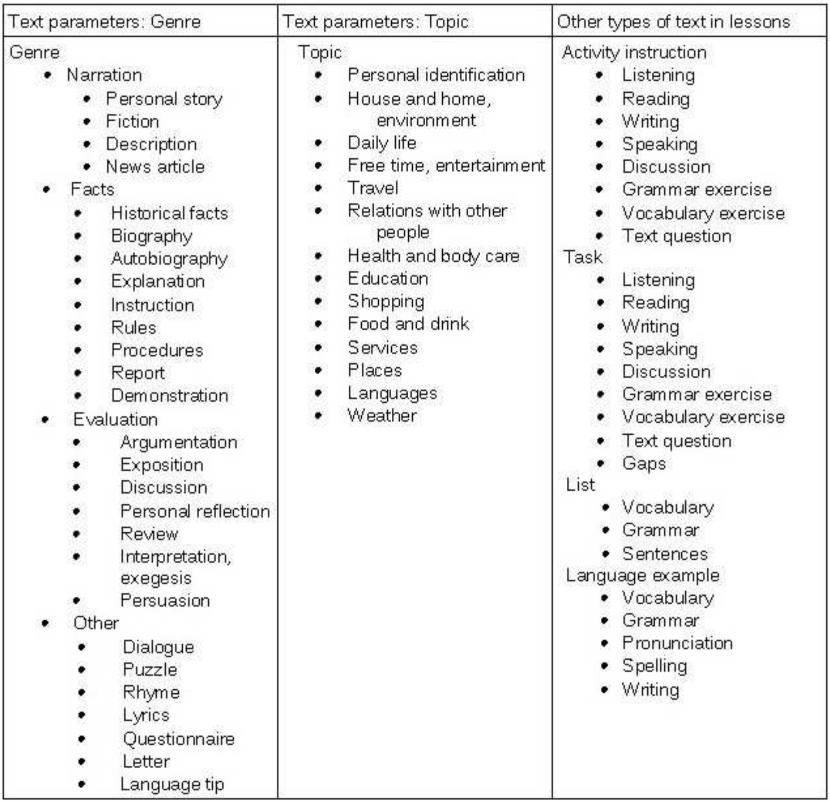
The taxonomy of text variables gives the key to different empirical and NLP-based studies. In our corpus, the text mass is divided into *Extras* (foreword, contents, acknowledgements, etc.) and *Lessons* (i.e. chapters). *Lessons*, further, contain different types of language and are subdivided into *Texts*, *Activity instructions, Tasks, Lists* and *Language examples*. A more fine-grained division of lesson-related text variables is shown in Figure 6.

Text genres is a modified version of genre families described in Martin & Rose (2008). The scheme over genre families has been extended by some macrofunctions according to the CEFR, e.g. *exposition, exegesis* (Council of Europe, 2001:126); as well as by the genre family marked as "*other*" which contains text types that we could not place in any of the main three families (narration, facts, evaluation). Among the a-typical (compared to Matin&Rose's genre families) text types are *puzzles, rhymes, lyrics, questionnaires, letters*, etc. The genre taxonomy is not final since we expect to encounter other deviating categories during the annotation work.

Topics have been derived from the CEFR document (Council of Europe, 2001:52). As with genres, we expect the list of topics to grow during the annotation period to cover the diversity of the topics in the course books.

Activity instructions usually precede the actual *Tasks* (e.g. exercises or text questions) and contain imperative sentences in the majority of cases. *Lists* provide active vocabulary for training or phrases/sentences to use during some tasks; whereas *Language examples* introduce new grammar or vocabulary patterns, that the learner should focus on and often contain explanations.

The division of the language used in *Lessons* into *Texts* and other categories is made to cater for different types of research that can be performed once the corpus is available. We plan, for example, to study the type of questions on different text genres to generalize about how questions differ in number and contents depending upon the genre and topic of the text, which will influence the question generation engine for that particular text genre.



Text parameters: Genre	Text parameters: Topic	Other types of text in lessons
Genre • Narration • Personal story • Fiction • Description • News article • Facts • Historical facts • Biography • Autobiography • Explanation • Instruction • Rules • Procedures • Report • Demonstration • Evaluation • Argumentation • Exposition • Discussion • Personal reflection • Review • Interpretation, exegesis • Persuasion • Other • Dialogue • Puzzle • Rhyme • Lyrics • Questionnaire • Letter • Language tip	Topic • Personal identification • House and home, environment • Daily life • Free time, entertainment • Travel • Relations with other people • Health and body care • Education • Shopping • Food and drink • Services • Places • Languages • Weather	Activity instruction • Listening • Reading • Writing • Speaking • Discussion • Grammar exercise • Vocabulary exercise • Text question Task • Listening • Reading • Writing • Speaking • Discussion • Grammar exercise • Vocabulary exercise • Text question • Gaps List • Vocabulary • Grammar • Sentences Language example • Vocabulary • Grammar • Pronunciation • Spelling • Writing

FIGURE 6. SUBMENUS OF THE MAIN ANNOTATION MENU FOR TEXT VARIABLES.

Once the *course book editor* is stable, it will be available for use for any other L2 language course book annotation, language independent. Since it is web-based, it can be accessed from anywhere without prior installation.

Annotation for linguistic variables includes annotation for parts of speech (pos), morpho-syntactic information (msd), syntactic relations (ref, dephead, deprel), lemmas, and linking to morphology lexicon (lex, saldo). This is an automated procedure that is used in Korp import pipeline (Borin et al. 2012), Korp being an infrastructure for storing and browsing a large collection of Swedish texts. Example of how a text can look after this annotation is given in Figure 7. In the near future we plan to build infrastructure in Korp for working with CEFR-related variables.

```
<w pos="DT" msd="DT.UTR.SIN.IND" lemma="|en|" lex="|en..al.1|" saldo="|den..1|en..2|"
prefix="|" suffix="|" ref="1" dephead="2" deprel="DT">En</w>
<w pos="NN" msd="NN.UTR.SIN.IND.NOM" lemma="|" lex="|" saldo="|" prefix="|
exempel..nn.1|" suffix="|text..nn.1|" ref="2" dephead="3"
deprel="SS">exempeltext</w>
```

FIGURE 7. EXAMPLE OF A TEXT ANNOTATED FOR LINGUISTIC VARIABLES

4. Concluding remarks

The problem of sparse data is well known in the area of computational linguistics, especially within machine learning, information extraction and other subfields that require reliable reference and training data, a “gold standard”, i.e. data that perfectly matches the purpose so that the instruments can be trained and fine-tuned on it. A collection of course book texts annotated for CEFR variables presented in this paper provides a unique training dataset for a variety of natural language processing tasks relevant for (but not limited to) ICALL, including topic modelling, genre identification, question generation and automatic classification of texts and sentences by their readability.

Access to such data in pedagogical empirical studies facilitates generalizations and approximations about language use in L2 context. With this project, we lay the ground for further pedagogically relevant studies of CEFR related texts in Swedish. The most important for us, however, is the fact that the access to this corpus is the only way to address the research agenda prompted by the development of the ICALL platform for Swedish.

The corpus based on course book texts cannot be made publicly available due to copyright restrictions. However, once the instruments for level classification and eventually topic categorization are reliable, it will be possible to classify arbitrary texts, e.g. texts available through Språkbanken's corpus infrastructure Korp (Borin et al., 2012) into CEFR levels and thematic domains. Since materials from Korp are digitally available, they will facilitate further studies of CEFR specific linguistic aspects per proficiency level. Text classification into levels and topics is eventually planned to be included into the standard annotation process for Korp for any new text collections.

Acknowledgements

The project on collecting CEFR-related texts has been financed partly by the Swedish Department at the University of Gothenburg (UGOT) and partly by Språkbanken, UGOT. Further, we extend our thanks to the publishing house *Liber* for providing electronic materials for our research.

References

Aldabe, I., Lacalle, M.L.D., Maritxalar, M., Martinez, E., Uria, L. (2006). ArikIturri: An Automatic Question Generator Based on Corpora and NLP Techniques. In *Intelligent Tutoring Systems* (2006), 584-594

Amaral, L. & Meurers, D. (2011). On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL* 23(1): 4–24.

Amaral, L., Meurers, D. & Ziai, R. (2011). Analyzing learner language: towards a flexible natural language processing architecture for intelligent language tutors. *Computer Assisted Language Learning* 24(1): 1–16.

Beinborn, L., Zesch, T., & Gurevych, I. (2012). Towards fine-grained readability measures for self-directed language learning. In *Electronic Conference Proceedings* (Vol. 80, pp. 11-19).

Borin, L., Forsberg, M., & Roxendal, J. (2012). Korp – the corpus infrastructure of Språkbanken. *Proceedings of LREC 2012*. Istanbul: ELRA, p.474–478.

Byrnes H. (2007). Perspectives. *The Modern Language Journal*, 91, iv, p.641–645.

Carlsten, C. (2012). Proficiency Level – a Fuzzy Variable in Computer Learner Corpora. *Applied Linguistics,* Volume 33(2), p.161-183

Collins-Thompson, K. & Callan, J. (2005). Predicting reading difficulty with statistical language models. *Journal of the American Society for Information Science and Technology*, 56(13). pp. 1448-1462.

Collins-Thompson, K. and Callan, J. (2007). Automatic and Human Scoring of Word Definition Responses. *Proceedings of NAACL HLT 2007,* 476-483. Rochester, NY.

Council of Europe. (2001). *The Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press.

Council of Europe. 2009. *Relating language examinations to the Common European Framework of Reference for Languages: learning, teaching, assessment (CEFR). A Manual*, Strasbourg: Language Policy Division.

Dávid, G.A. 2010. Linking the general English suite of Euro Examinations to the CEFR: a case study report. In Martyniuk, W. (Ed.) *Aligning Tests with the CEFR*. Cambridge University Press, p.177-203.

Einarsson, J. (1976). *Talbanken: Talbankens skriftspråkskonkordans/ Talbankens talspråkskonkordans*. Lund University.

Francois, T. & Miltsakaki, E. (2012). Do NLP and Machine Learning Improve Traditional Readability Formulas? In *Proceedings of the First Workshop on Predicting and Improving Text Readability for Target Reader Population*, NAACL

Hawkins, J. A. & Buttery, P. (2009). Using learner language from corpora to profile levels of proficiency: Insights from the English Profile Programme. In Taylor, L. & Weir, C. J. (Eds). *Language Testing Matters: Investigating the Wider Social and Educational Impact of Assessment,* 158-175. Cambridge: Cambridge University Press.

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

Heilman, M., Collins-Thompson, K., Callan, J. and Eskenazi, M. (2007). Combining Lexical and Grammatical Features to Improve Readability Measures for First and Second Language Texts. *Proceedings of NAACL HLT 2007,* 460-467. Rochester, NY.

Heimann Mühlenbock, K. (2013). *I see what you mean: Assessing readability for specific target groups.* PhD Thesis. Data linguistica, University of Gothenburg.

Hultman, T. G. & Westman, M. (1977). *Gymnasistsvenska*. Lund: Liber Läromedel.

Johansson Kokkinakis, S. & Magnusson, U. (2011). Computer based quantitative methods applied to first and second language student writing. *Young urban Swedish. Variation and change in multilingual settings.*University of Gothenburg, 105-124.

Kate, R. J., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R. J. Roukos, S. & Welty, C. (2010). Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 546-554). Association for Computational Linguistics.

Khalifa, H., Ffrench, A. & Salamoura, A. 2010. Maintaining alighnment to the CEFR: the FCE case study. In Martyniuk, W. (Ed.) *Aligning Tests with the CEFR*. Cambridge University Press, p.80-101.

Kilgarriff A., Charalabopoulou F., Gavrilidou M., Bondi Johannessen J., Khalil S., Johansson Kokkinakis S., Lew R., Sharoff S., Vadlapudi R, Volodina E. (accepted, LREJ 2013). Corpus-Based Vocabulary lists for Language Learners for Nine Languages. *Language Resources and Evaluation Journal*, special issue.

Kilgarriff, A., Husak, M., McAdam, K., Rundell, M., & Rychlý, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. In *Proc. Euralex*.

Knoop, S. & Wilske, S. (2013). Automatic Generation of Gap-Filling Vocabulary Exercises for Mobile Learning. 2nd workshop on NLP in Computer-Assisted Language Learning. *Proceedings of the NODALIDA 2013 workshop on NLP for CALL*. Linköping Electronic Conference Proceedings 85.

Källgren, G., Gustafson-Capková, S. and Hartmann, B. (2006). *Manual of the*

Stockholm Umeå Corpus version 2.0. Department of Linguistics, Stockholm University.

Lindberg, I. & Johansson Kokkinakis, S. (2007). *OrdiL - en korpusbaserad kartläggning av ordförrådet i läromedel för grundskolans senare år.* Göteborgs universite

Lindberg, I. & Johansson Kokkinakis, K. (2009). Word Type Grouping in Swedish Secondary School Textbooks - An Inventory of Words from a Second Language Perspective *Multilingualism, Proceedings of the 23rd Scandinavian Conference of Linguistics*. 337-339

Little D. (2007). The Common European Framework of Reference for Languages: Perspectives on the Making of Supranational Language Education Policy. *The Modern Language Journal* 91, p.645–655.

Little D. (2011). The Common European Framework of Reference for Languages: A research agenda. *Language Teaching,* Vol 44.3, p.381-393. Cambridge University Press 2011.

Martin, J.R. & Rose, D. (2008). *Genre Relations.* Equinox Publishing Ltd.

Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V. & Ott, N. (2010. Enhancing Authentic Web Pages for Language Learners. *Proceedings of the 5th Workshop on Innovative Use of NLP for Building Educational Applications, NAACL-HLT 2010, Los Angeles.*

Milton, J. (2009). *Measuring Second Language Vocabulary Acquisition*. Toronto: Multilingual Matters.

Nagata, N. 2009. Robo-Sensei‘s NLP-based error detection and feed-back generation. *CALICO Journal*, 26(3), 562–579.

Nivre, J., Nilsson, J. and Hall, J. (2006). Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation. In P*roceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)* Genoa: ELRA. 1392-1395.

North, B. (2007). The CEFR illustrative descriptor scales. *The Modern Language Journal* 91, p.656–659.

Nyström, C. (2000). *Gymnasisters skrivande. En studie av genre, textstruktur och sammanhang*. Uppsala: Uppsala universitet.

Pijetlovic, D. & Volodina, E. (forthcoming). Developing a Swedish spelling game on an ICALL platform. *Proceedings of EuroCALL 2013*.

Pilán, I., Volodina, E. & Johansson, R. (forthcoming). Automatic selection of suitable sentences for language learning exercises. *Proceedings of EuroCALL 2013*.

Szabó, G. 2010. Relating language examinations to the CEFR: ECL as a case study. In Martyniuk, W. (Ed.) *Aligning Tests with the CEFR*. Cambridge University Press, p.133-144.

Tanaka-Ishii, K., Tezuka, S., & Terada, H. (2010). Sorting texts by readability.*Computational Linguistics*, *36*(2), 203-227.

Teleman, U. (1974). *Manual för grammatisk beskrivning av talad och skriven svenska*. Lund.

Toole, J. & Heift, T. (2002). Task-Generator: A Portable System for Generating Learning Tasks for Intelligent Language Tutoring Systems. *Proceedings of ED-MEDIA 02, World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Charlottesville, VA: AACE: 1972-1978.

Volodina, E. and Borin, L. (2012). Developing a freely available web-based exercise generator for Swedish. *CALL: Using, Learning, Knowing. EuroCALL Conference, Gothenburg, Sweden, 22-25 August 2012, Proceedings.* Eds. Linda Bradley and Sylvie Thouësny. Research-publishing.net, Dublin, Ireland.

Volodina, E., Borin, L., Loftsson, H., Arnbjörnsdóttir, B. & Örn Leifsson, G. (2012a). Waste not, want not: Towards a system architecture for ICALL based on NLP component re-use. Workshop on NLP in Computer-Assisted Language Learning. *Proceedings of the SLTC 2012 workshop on NLP for CALL.* Linköping Electronic Conference Proceedings 80: 47-58.

Volodina, E., Johansson, R. & Johansson Kokkinakis, S. (2012b). Semi-automatic selection of best corpus examples for Swedish: Initial algorithm evaluation. Workshop on NLP in Computer-Assisted Language Learning. *Proceedings of the SLTC 2012 workshop on NLP for CALL.* Linköping Electronic Conference Proceedings 80: 59–70.

Volodina, E. & Johansson Kokkinakis, S. (2012). Introducing Swedish Kelly-list, a new lexical e-resource for Swedish. *Proceedings of LREC 2012*. Istanbul: ELRA.

Westhoff G. (2007). Challengens and Opportunities of the CEFR for Reimagining Foreign Language Pedagogy. *The Modern Language Journal* 91, p.676–679.

Östlund-Stjärnegårdh, E. (2002). *Godkänd i svenska? Bedömning och analys av gymnasieelevers texter.* Uppsala: Uppsala universitet.

Sponsors of NODALIDA 2013 & NEALT

WeSearch

iness

Lingit

max manus

computas

The Center of Estonian Language Resources

DET HUMANISTISKE FAKULTET
KØBENHAVNS UNIVERSITET

GSLT

Lingsoft®
LANGUAGE SOLUTIONS

Mikro Værkstedet

National Library of Norway

textUrgy™

www.tungutaekni.is

Design • Joel Priestley
Photo • Arthur Sand

NEALT Proceedings Series 17 • ISBN 978-91-7519-588-9
Linköping Electronic Conference Proceedings 86
ISSN 1650-3740 (Online) • ISSN 1650-3686 (Print) 2013