Improving feedback on L2 misspellings – an FST approach

Lene Antonsen
University of Tromsø, Norway
`lene.antonsen@uit.no`

Abstract

This paper suggests enriching the finite-state transducers (FST) analyser with erroneous forms marked with error tags, as a way of improving feedback on L2 misspellings. This approach can be useful both in isolated word error correction and in detecting real word errors in context-dependent word correction. But most important, it makes it possible to give metalinguistic feedback on the nature of the errors.

1 Introduction

When learning a language with a rich system of inflectional morphology like North Saami, the learner has to focus on form if the goal is to achieve near-native fluency in L2. The learner's awareness of the relevant morphological processes in the language plays a crucial role.

A computer can parse a language on the basis of its standard linguistic forms. We are looking for a way to enable the computer to parse a language even when the forms produced by learners deviate from the target language forms. In other words, we want to find a way for the computer to interpret learners' intentions as represented in their interlanguage forms. This would make it possible for the computer both to recognise forms even when they are misspelled (overlooking the errors) and to heighten the learners' awareness of morphological processes by correctly interpreting their mistakes.

This paper is structured as follows: Section 2 discusses L2 misspellings and looks at different kinds of feedback and how L2 misspellings are recognised by a generic spell checker. Section 3 describes the enriching of the morphological analyser with systematic misspellings and section 4 describes how the analyser functions in an ICALL program with free input. Finally, in sections 5 and 6 I present a conclusion and some assumptions of how an enriched FST can be utilised in automatic writing assistant tools for language learners.

2 Background

A misspelling is a written form that deviates from the conventions in the written language. The misspelling can result in a non-word, an unintended word form of the same lemma, or a new lemma. A human teacher can usually interpret the student's intention behind the misspellings, but the misspelling makes it more difficult for a computer to give the correct syntactic analysis, and in that way it complicates the human-computer interaction.

2.1 North Saami

The Saami languages are morphologically complex suffixing languages with much suprasegmental morphology. Nouns and verbs have about 100 inflected forms, half of the forms for verbs are finite forms. North Saami is the largest Saami language, with only approximately 17 000 speakers, but both schools and universities offer courses for students who want to learn the language.

The orthography conventions differ substantially from the native language of most of the students. North Saami extends the Latin alphabet with seven letters by means of diacritical marks (e.g. *š, č*), where Norwegian and Swedish use letter combina-

tions (*skj, tsj*). All diphthongs are different from Norwegian and Swedish diphthongs, and some of the graphemes represent other phonemes than in Norwegian and Swedish. Compared to Finnish the differences are smaller, but also the Finnish alphabet does not contain consonant letters with diacritical marks. Especially the Norwegian and Swedish language learners often fail to write the correct form, because of differences between their L1 and Saami, both with regard to morphology and orthographical conventions.

2.2 L2 misspellings

In the learner's production there will be both accidental mistyping and incorrect word forms due to misconceptions of the target language. Corder (1967) makes a distinction between **errors of performance**, which characteristically are unsystematic, and **errors of competence**, which are systematic. From the latter it is possible to reconstruct the learner's knowledge of the language.

The errors of competence can be divided into two groups: morphologically irrelevant, but still systematic ones, like writing *a* instead of *á* in the stem, and morphologically relevant ones, like omitting suprasegmental processes in certain kinds of inflections, e.g. skipping monophtongization when going from the nominative form *viessu* 'house' to the illative form *vissui* 'to the house' (which gives the erroneous form *viessui*), or choosing a wrong inflection for the context. According to the system proposed by James (1998), the former group consists of **substance errors** that violate certain convention for representing phonemes by means of graphemes, and the latter one consists of **text errors**. Also these are systematic errors.

2.3 Feedback

The student usually needs feedback to correct his own errors. The feedback can be a comment saying that something is wrong in the sentence, the erroneous word or phrase can be highlighted, or the student can be provided with the target word or a list of possible target words. Another kind of feedback is a metalinguistic comment saying what is wrong and why, possibly hyperlinked to more information about the phenomenon. 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 word or phrase, but also understand the reason for his misconception.

If the misspelling is an error of performance, it is sufficient to make the student aware of it. But if it is an error of competence, the student needs a correction, and if it is a metalinguistic comment, it is crucial to give a feedback according to the student intended writing and at his own level of competence. This is the challenge when coming real word errors. The student will be confused when getting feedback on syntax instead of the misspelling, e.g. feedback on using an infinite form instead of a finite form, when the student believes he has written a finite form.

2.4 L2 and spell checkers

Most spell checkers are generic and made for L1 users, but also language learners use them. The feedback from the spell checker is usually a suggestion for a more appropriate target word, or more often, a list of candidates for the target word. Most spell checkers detect errors and suggest corrections without using context, and therefore only detect non-word errors.

For detecting real word errors it is necessary to use the context. A real word error can lead to a syntactic or a morphosyntactic error, the challenge for the spell checker is to point out which word in the sentence is the incorrect one. There is a work in progress on building a grammar checker for North Saami that also considers real word errors, with L1 users as the target group (Wiechetek, 2012).

Another challenge, independent of whether it is a non-word error or a real word error, is to give the correct suggestion for how to correct the word (Kukich, 1992). The algorithm for suggesting correct candidates in spell checkers for native writers, is based on using as few editing steps as possible, going from the misspelled word to the target word.

A few spell checkers for non-native writers have been developed, most of them are specifically targeting certain error classes. There are spell checkers that incorporate lists of common misspellings in the target language, retrieve suggestions based upon the phonological representation of the misspelling, address morphological triggered misspellings and oth-

ers provide references to alternative spellings, e.g. on the internet. Nevertheless, spell checkers for non-native writers are rare. (Rimrott and Heift, 2008a). There is no such spell checker for North Saami.

Spell checkers are constructed in order to identify errors and give the most relevant suggestion for the correction, but in a language learning context, it can be even better to be able to give metalinguistic feedback to the student, e.g. 'Remember diphthong simplification when adding the suffix -i' for the misspelling *viessui*, which is used as example in section 2.2. Alternatively, one can ignore the misspelling in favour of concentrating upon the syntax of the learner's input.

In order to test the effect of a spell checker, I annotated errors in a corpus consisting of L2 sentences (4633 words, 800 sentences, 739 misspellings). Rimrott and Heift (2008a) present a similar testing for German, but unlike them, I considered also real word errors.

The North Saami spell checker[1] is based on dictionary lookup and dynamic compounding, and is designed for native speakers. The word forms are produced with finite-state transducers, which are explained in section 3.

The error model is based upon edit distance, which is the number of operations applied to the characters of a string: deletion, insertion, substitution, and transposition. In the literature, the edit distance has usually been found to cover more than 80 % of the misspellings at distance one. (Levenstein, 1965; Damerau, 1964). In the algorithm of the North Saami spell checker there are additionally phonetic rules. Errors with the same error distance are ranked based upon phonetic likelihood.

Testing the L2 corpus on a test bench[2] for the spell checker gave a precision of 0.92 and a recall of 0.74. The real word errors constitute 26.0 % of the errors and are therefore not detected by the spell checker. I also looked at the generation of the correct suggestions. In table 1 it appears that for 19.9 % of the misspellings, the program could not generate a correct candidate at all. The average edit distance for these misspellings were 2.74.

Testing shows that for L2 writers, the order in

true positives	correct cand. among top 3	no correct cand. among top 3	no correct cand.
99.9 %	67.7 %	12.3 %	19.9 %
aver. edit distance	1.39	1.59	2.74

Table 1: The spell checker's candidates for the true positives. For 32.3 % of the words that were correctly indentified as misspelling, there was no correct candidate among the top three candidates. N=563

which the words appear in the suggestion list, seems to be an influencing factor for selecting one word over another (Rimrott and Heift, 2008b). This implies that an L2 student is probably not able to choose between a large number of candidates. Table 1 shows that only in 67.7 % of the cases the correct suggestion is among the top 3. This result is poorer than the accuracy level above 90 %, which is usually reported on L1 misspellings, when the first three guesses are considered (Kukich, 1992). For the North Saami spell checker the level is 85 % for L1.

This test demonstrates that the spell checker is not sufficient for L2 writers because a relatively big part of their misspellings are real word errors that are not identified, and for the non-word errors the generation and ranking of candidates was not good enough for 32.3 % of the cases. The main reason is that the average edit distance for the L2 misspellings was as high as 1.54. A similar annotated corpus of L1-sentences gave an average edit distance of 1.26. The second reason is probably that the phonetic rules, which rank candidates, do not suit L2 writers, who often are not sure about the word's pronunciation.

3 Enriching the FST with systematic misspellings

3.1 Finite-state transducers

Instead of listing all word forms of a language, one may list all the stems and affixes, and combine them to word forms by means of finite-state automata, see figure 1 for an example.

A finite-state transducer is a finite-state automaton that maps between two strings of characters: the

---

[1] http://divvun.no/
[2] Moshagen (2008) describes the test bench.

Figure 1: This finite-state automaton produces the word forms *lávka, lávkan* ('bag.N') and *girji, girjin* ('book.N').

word form itself and the grammatical word, like in figure 2: *girjin* (lower level) and *girji+N+Ess* (upper level).
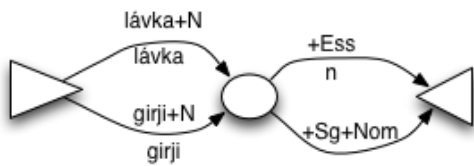

Figure 2: This finite-state transducer produces the same word forms as in figure 1, but it also maps between the word form and the grammatical word.

3.2 Modelling misspellings with FST

The FST models the language in question by producing the correct word forms. But the FST can also model systematic misspellings with specific error tags in the upper level. In that way the analyser identifies the word as an erroneous form of a certain grammatical word. The modelling of misspellings be utilised in several ways:

1. The ranking of suggestion candidates in isolated word correction can be improved by giving priority to systematical L2 errors, some of them with an edit distance bigger than 1.

2. The morphological analysis combined with error tag makes it easier to detect real word errors in context-dependent word detecting.

3. The specific error tag also makes it possible to give metalinguistic comments about the morphological nature of the misspellings, both for non-word and real word errors.

According to the system of errors in section 2.2, two kinds of systematic errors can be added to the FST: substance errors (errors in encoding/decoding), and text errors (usage errors), like omitting suprasegmental processes.

The FST that the North Saami spell checker is based upon, consists of a lexical transducer `lexc` and a phonological transducer `twolc` for the suprasegmental processes (Koskenniemi, 1983). It is compiled with the Xerox compilers (Beesley and Karttunen, 2003), and is available as open source[3] under the terms of the GNU General Public License. I have added systematic misspellings to both the lexical and the phonological transducers. Additionally, certain kinds of misspellings are taken care of by concatenating the final transducer with another transducer containing these misspellings.

3.3 Adding paths to the lexical transducer

Suffixes are added and some vowel and consonant changes are made in the lexical transducer. The ordinary illative suffix *-ii* for nominals with trisyllabic stem, is added in `lexc`. For the same stems I made an extra path with the suffix for nominals with bisyllabic stem, *-i*, marked with the error tag `IllErr` (= incorrect illative suffix) in the upper (here: right) level. In example 1 are the analyses for the misspelling *hivssegi* and the target form *hivssegii*.

Ex. 1
```
"<hivssegi>" "hivsset" N Sg Ill IllErr
"<hivssegii>" "hivsset" N Sg Ill
                        'to the toilet.N'
```

Some suprasegmental processes that are taken care of in the phonological transducer, are triggered by a dummy symbol in the lexical transducer. The erroneous path is made without this dummy, e.g. inflections with strong grade for the consonant centre when there should have been weak grade, as in figure 3. The error tag in upper level is `CGErr` (= lacking consonant gradation), see example 2. The target form is *áhku*.

Ex. 2

```
"<áhkku>" "áhkku" N Sg Nom 'grandmother.N'
"<áhkku>" "áhkku" N Sg Acc CGErr
"<áhku>" "áhkku" N Sg Acc
```
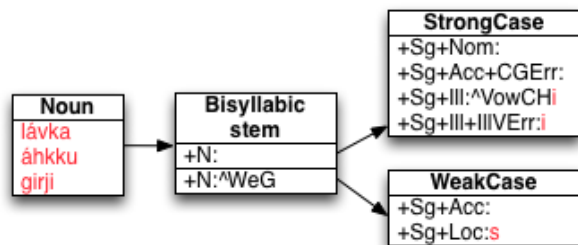

Figure 3: The lexical transducer, `lexc`, is adding both suffixes (*-s, -i*) and dummies for the phonological transducer to the stem. The dummies here are `WeG` for consonant centre in weak grade and `VowCH` for vowel change. The erroneous paths without the dummies are marked with error tags: +CGErr and +IllVErr.

3.4 Generating misspellings with the phonological transducer

The phonological transducer changes letters under specific conditions. In figure 4 the consonant centre is changed from *hkk* to *hk*, or *vk* to *vkk*, if it is followed by one or more vowels and `WeG`, which is a dummy[4].

```
hkk -> hk, vk -> vkk, ... || _ Vow* WeG ;
```

Figure 4: The phonological transducer, `twolc`.

Some misspellings are generated by first adding a path with error tags to both upper and lower level in `lexc`, and then removing the error tag from the lower level under special conditions in `twolc`. The analyses with error tag in both levels are then removed from the output of the FST, by means of regex-rules.

The erroneous path can be a rule that changes letters generally from a letter with a diacritic mark to a letter without, e.g. changing *á* into *a*. The path with the error tag `AErr` remains in the upper

---
[4]For details, see Trosterud and Uibo 2005.

level only if the change happens. In example 3, the misspelling *barru* and the target form *bárru* are analysed:

Ex. 3

```
"<barru>" "bárru" N Sg Nom AErr
"<bárru>" "bárru" N Sg Nom     'wave.N'
```

Other rules change letters under special conditions, such as diphthong simplification, and the erroneous path with error tag `DiphErr` (= omitted monophthongization) will remain only if the diphthong simplification does not happen. The misspelling *viessui* and the target form *vissui* are analysed in example 4:

Ex. 4

```
"<viessui>" "viessu" N Sg Ill DiphErr
"<vissui>" "viessu" N Sg Ill
                        'to the house.N'
```

3.5 Adding paths by concatenating transducers

There is also a special transducer for lowercase initial letters in place names, which is concatenated to the main transducer after the first compilation process. All forms have the tag `LowercaseErr` in the upper level, and in example 5 is the analysis of the misspelling *lundas* and of the correct *Lundas* ('in Lund'):

Ex. 5

```
"<lundas>" "Lund" N Prop Plc Sg Loc
      LowercaseErr
"<Lundas>" "Lund" N Prop Plc Sg Loc
```

3.6 More readings before disambiguation

Table 2 lists the systematic misspellings I added to the FST. Two of them are substance errors, `Lowercase` and `AErr`. The latter one is *a* instead of *á*, the most frequent letter with diacritic mark in the North Saami alphabet.

The other misspellings in table 2 are text errors, products of incorrect inflection. All erroneous forms are marked with an error tag that characterises their nature, like `AiErr` (= a inflection error): *a* is written where it should be vowel change from *i* to *á* caused

of an inflection. Most of the systematic misspellings were added to nouns.[5]

error tag	erroneous form	target form
Lowercase (place names)	londonis	Londonis 'London.SgLoc'
AErr (general rule)	manna	mánná 'child.SgNom'
AiErr (verbs)	boahtan	boahtán 'come.V.PrfPrc'
CGErr (nouns)	skuvlas	skuvllas 'school.SgLoc'
DiphErr (nouns)	viessui	vissui 'house.SgIll'
IllVErr (nouns)	skuvlai	skuvlii 'school.SgIll'
IllErr (nouns)	hivssegi	hivssegii 'toilet.SgIll'

Table 2: Systematic misspellings added to the FST.

By enriching the morphological analyser with erroneous forms, the number of possible readings increases. In figure 5, the morphological analysis of the sentence is done with the regular FST. There are two misspellings, which are unknown to the analyser.

```
"<Ahkku>"
 "Ahkku" ?
"<manná>"
 "mannat" V IV Ind Prs Sg3
"<lundii>"
 "lundii" ?
"<odne>"
 "odne" Adv
```

Figure 5: 'Grandmother goes to Lund today.' analysed with the regular FST.

The same input is then analysed with the error-FST in figure 6, and the misspellings are recognised as an erroneous form of *áhkku* 'grandmother' (*a* instead of *á*), and an erroneous form of the place name *Lundii* (in illative case 'to Lund'), with initial lowercase letter. Also the correctly spelled word *manná* 'goes' gets several possible erroneous readings.

Disambiguation of the multiple readings will be explained in section 4.1.

---

[5]The makefile and source files can be downloaded: https://victorio.uit.no/langtech/branches/errortag/gt/

```
"<Ahkku>"
 "áhkku"  CGErr Sg Acc AErr
 "áhkku"  CGErr Sg Gen AErr
 "áhkku" N Sg Nom AErr        <- correct
"<manná>"
 "mannat" V IV Ind Prs Sg3    <- correct
 "mánná" Hum N Sg Nom AErr
 "mánná" Hum N CGErr Sg Acc AErr
 "mánná" Hum N CGErr Sg Gen AErr
"<lundii>"
 "Lund" N Prop LowercaseErr Plc Sg Ill
"<odne>"
 "odne" Adv
```

Figure 6: 'Grandmother goes to Lund today.' analysed with error-FST. The error tags are explained in table 2. The correct analyses are marked.

4 Evaluation

The erroneous forms with error tags in the analysis make it possible to recognise the target form. The evaluation will show to which extent the added erroneous forms cover the L2 misspellings, and how the multiple readings influence upon the disambiguation. I will also discuss the consequences for the size of the FST.

4.1 Test bench

I use the syntactic analyser from an existing ICALL-program[6] as a test bench for the error-FST. The ICALL-program accepts free-input, and has L2 learners as its target group.

In the ICALL-program there are three different question-answer drills with free input. For two of them, the questions are generated, for one of them the student can answer freely. The other one presents 2-4 lemmas, which should form part of the answer. The third drill is a tailored dialogue, but the student can answer freely to the questions. All three QA-drills use the same analyser. The tutorial feedback concerning grammatical errors is given in a separate window, and the user is allowed to correct the answer until it is accepted.

The morphological analyser is the one described in section 3. The morphological disambiguator is implemented in the Constraint Grammar (CG) framework (Karlsson et. al, 1995). The rules are

---

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

compiled with vislcg3[7], and they are manually written, context dependent rules used for selecting and discarding analysis.

The CG-rule set consists of two parts. The first part is a rule set that disambiguates the user's input only to a certain extent. The rule set is relaxed compared to the ordinary disambiguator, in order to be able to detect relevant readings despite a certain degree of grammatical and orthographic errors in the input. The second part of the rule set contains rules for giving feedback to grammatical errors. Question and answer are merged, and given to the analyser as one text string, with only a tag as delimiter between question and answer, so that one can refer to the question and the answer separately in the CG-rules[8].

These ICALL programs are designed with students at introductory level as the target group. Till now feedback to misspellings in the program is handled by pointing to the unrecognised word form, asking the student to check the spelling. Only a couple of systematic real word errors give more specific feedback to the student on the nature of the misspelling. The misspellings constitute the biggest problem for the human-computer interaction (Antonsen et.al., 2009b). Pointing out the misspelled word is not enough help to the student. The system is not like a human reader able to read the answer in a robust way, and detect what the student intended to write.

By using an error-FST as morphological analyser in the ICALL-program, it should be to some extent possible to recognise the student's intended word, and also to make more CG-rules, which trigger metalinguistic feedback as help for the student, e.g. 'X misses diphthong simplification'.

4.2 Test results

I have been testing a part of the programs' student log, consisting of 2705 question-answer pairs. The pairs were parsed with the regular and the error-FST, respectively, and then parsed with the CG-rules.

The erroneous forms in the error-FST cause the number of analyses to increase from 74 517 to 83 582 (+ 12.1 %), from 2.26 to 2.54 per word form

[7]http://beta.visl.sdu.dk/constraint_grammar.html

[8]For details, see Antonsen et.al. 2009a.

before disambiguation. But the disambiguation is quite efficient, as shown in table 3. The erroneous path `CGErr` is the most productive one. It gives only real word errors, and therefore many correct word forms get a possible error analysis in addition to the correct analysis. But the extra readings do not mess up the disambiguation, which removes 93.7 % of the extra readings.

The `IllErr`-path, incorrect illative suffix added to nominals with trisyllabic stem, gives only non-word errors, and the word forms get only this analysis. The other erroneous paths can both give real word and non-word errors.

errortag	before disamb.	after disamb.
CGErr (nouns)	1786	113
AErr (general rule)	1395	524
Lowercase (place names)	534	65
AiErr (verbs)	214	95
IllVErr (nouns)	74	27
IllErr (nouns)	28	28
DiphErr (nouns)	22	16

Table 3: Parsing 2705 QA-pairs with error-FST. The number of analyses with error tag before and after disambiguation.

The analysis also recognises combinations of the erroneous forms, like in example 6, where the word *fallejohkas* is recognised as a misspelling of the target form *Fállejogas* despite of an edit distance of 4.

Ex. 6

```
"<fallejohkas>" "Fállejohka" N Prop Plc Sg
      Loc LowercaseErr CGErr AErr
"<Fállejogas>" "Fállejohka" N Prop Sg Loc
```

The disambiguation does not need to be complete, because of the special CG-rules deciding whether the student gets an error feedback or not.

Table 4 lists how many misspellings were found in the corpus, and what kind of analysis they got. By parsing the test corpus with the regular FST combined with the CG rule set, the target form was recognised for only 8.1 % of the misspellings. They were recognised by means of special CG-rules for systematic real word errors. By parsing the test

corpus with the error-FST, the target forms could be recognised regardless of whether they were real word errors or non-word errors. The target form was recognised for 44.0 % of the misspellings.

Errors The target form was	Reg.FST.		Err.FST	
not recognised	871	91.9 %	563	56.0 %
recognized	77	8.1 %	443	44.0 %
Total	948	100 %	1006	100 %

Table 4: Parsing 2705 QA-pairs. Comparing the regular FST with the error-FST. Some sentences have more than one misspelling.

Table 5 contains a comparison of the error messages, which were given with the two different FST's. In addition to feedback on misspellings, the student also gets feedback on syntactic errors, e.g. 'Remember the agreement between subject and verbal', and semantic comments, e.g. 'You must use the given verb.' The latter message is given if the student does not use the given lemmas in the QA-drill that calls for it. All QA-drills require that the students formulate complete sentences, otherwise they get comments on that (here called comment on semantics).

In table 4, the error-FST diagnoses more errors as misspellings than the regular FST, because more of the real word errors are recognised as misspellings instead of syntactic errors, see also table 5. E.g. the frequent misspelling *vuolggan* in example 7 gets a noun analysis with the regular FST. The error-FST gives an additional analysis as a misspelled finite verb with target form *vuolggán*, and the disambiguation can therefore result in a feedback about a misspelling instead of a syntactic error or a messages about a missing finite verb in a sentence:

Ex. 7

```
"<vuolggan>"
 "vuolgga" N Ess              'departure'
 "vuolgit" V IV Ind Prs AiErr Sg1 'I leave'
```

The number of error feedback tags is bigger than the number of actually given feedbacks, since some sentences get more than one error feedback, but the

system presents only one at a time to the student. Sometimes two or more feedback tags are related to the same error. Important is that the precision and recall did not decline when using the error-FST compared to the regular FST.

Feedback	Reg.FST.	Err.FST
Misspellings	751	804
Syntactic errors	1181	1071
Comments on semantics	599	527
Altogether	2531	2402
Number of sentences giving feedback on errors	1560	1561

Table 5: Parsing 2705 QA-pairs. Some sentences have more than one error feedback. Prec=0.96 Rec=0.99 for both FST's.

Among the unrecognised misspellings there are some frequent systematic groups that could be added to the FST, e.g. omitting vowel change in trisyllabic nominal stems and omitting monophthongization and consonant gradation in verbs.

4.3 The size of the FST

All the extra paths make the FST much bigger. The size of the error-FST is almost ten times as big as the regular FST, as shown in table 6, even if most of the error paths added to the error-FST so far are for nouns only. Paths with missing monophthongization and missing consonant gradation are also relevant for inflection of verbs and adjectives. The compilation time increases with 570 %, e.g. on a MacBook Pro (OS 10.6.8) from 3.5 minutes to 23.5 minutes. The time needed for initiating the analysis is more important, but in the ICALL program in which the error-FST was tested, the lookup process is done in a standby server, and start-up delay is thus not relevant. The size of the FST still has impact on the time for analysis, but not so dramatically. However, it is possible to make the error-FST smaller by removing rare dynamic compounding and derivation paths, which are not likely to occur in the language of L2-students.

5 Conclusion

Enriching the FST-analyser with erroneous forms marked with error tags gives promising results. It

	Regular FST	Error FST
size	41.5 Mb	398.8 Mb
	100 %	959 %
states	497 632	4 739 590
arcs	1 062 995	10 297 121

Table 6: The size of the regular FST and the error-FST.

makes the syntactic analyser able to recognise systematic misspellings, both real word errors and non-word errors, even if the edit distance is big.

Even though the number of analyses per word form increases, it does not destroy the disambiguation in a restricted ICALL program. In fact, by means of the erroneous forms some errors are re-classified from syntactic or semantic errors to misspellings.

The error tags make it possible not only to recognise the target form, but also to give tutorial feedback on the nature of the error to the student. When the analyser identifies the grammatical word despite the misspelling, it is possible to ignore misspellings in favour of giving feedback on syntax.

The size of the error-FST expands exponentially, but it can be trimmed for L2 users.

6 Future work

It will be useful to have a closer look at the nature of L2 misspellings in a larger material, and give more erroneous forms to the FST, combined with restricting of the dynamic derivations and compounding, so the FST will not be too large for implementation in end-user applications.

In a spell checker for isolated non-word errors one may test how useful it is to rank the correction candidates with a combination of edit distance and the erroneous forms from the FST, instead of using phonetic rules as was done for the L1 spell checker.

I will also try out the combination of error-FST and constraint grammar in free-input student tasks that are less restricted than the present ICALL-program. Constraint grammar has been tried out for ruling out correction candidates that are grammatically unacceptable in spell checker programs for English L1 and Danish dyslectics (Agirre et.al., 1998; Bick, 2006).

The combination of erroneous forms with error

tags and constraint grammar parsing makes it possible to give metalinguistic feedback. It is important to look more into the human-computer interaction, e.g. by means of looking at the log to see how the students correct their input after getting metalinguistic feedback and making a survey for the students about how useful they find the feedback.

References

Eneko Agirre, Koldo Gojenola, Kepa Sarasola and Atro Voutilainen. 1998. Towards a Single Proposal in Spelling Correction. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*: 22–28. Association for Computational Linguistics.

Lene Antonsen, Saara Huhmarniemi and Trond Trosterud. 2009a. Constraint Grammar in Dialogue Systems. *Proceedings of the 17th Nordic Conference of Computational Linguistics.* NEALT Proceeding Series. Volum 8: 13–21. Odense, Denmark.

Lene Antonsen, Saara Huhmarniemi and Trond Trosterud. 2009b. Interactive pedagogical programs based on constraint grammar. *Proceedings of the 17th Nordic Conference of Computational Linguistics.* NEALT Proceeding Series Volum 4: 10–17. Odense, Denmark.

Kenneth V. Beesley and Lauri Karttunen. 2003. *Finite State Morphology.* CSLI publications in Computational Linguistics. USA.

Eckhard Bick. 2006. A Constraint Grammar Based Spellchecker for Danish with a Special Focus on Dyslexics. (ed.) Suominen, Mickael et al. *A Man of Measure – Festschrift in Honour of Fred Karlsson*: 387-396. The Linguistic Association of Finland.

S. P. Corder. 1967. The significance of learner's errors. *International Review of Applied Linguistics* 5: 161–169.

Frederick J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM* 7.

Carl James. 1998. *Errors in language learning an use: exploring error analysis.* Longman. USA.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä and Arto Anttila. 1995. *Constraint grammar: a*

language-independent system for parsing unrestricted text. Mouton de Gruyter.

Kimmo Koskenniemi. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production.* Publications of the Department of General Linguistics, University of Helsinki, No. 11.

Karen Kukich. 1992. Techniques for Automatically Correcting Words in Text. *ACM Compiling Surveys* 24(4): 377–439.

Vladimir I. Levenstein. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10.

Sjur Nørstebø Moshagen. 2008. A language technology test bench – automatized testing in the Divvun project. Proceedings of the Workshop on NLP for Reading and Writing – Resources, Algorithms and Tools. *NEALT Proceeding Series* 3: 19–21.

Anne Rimrott and Trude Heift. 2008a. Evaluating automatic detection of misspellings in German. *Language Learning & Technology* 12(3): 73–92.

Anne Rimrott and Trude Heift. 2008b. Learner responses to corrective feedback for spelling errors in CALL. *System* 36(2): 196–213.

Trond Trosterud and Heli Uibo. 2005. Consonant Gradation in Estonian and Sami: Two-Level Solution. (Eds) Antti Arppe et al. *Inquiries into Words, Constraints and Contexts:* 136–150.

Linda Wiechetek. 2012. Constraint Grammar based Correction of Grammatical Errors for North Sámi. *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SaLT-Mil 8 – AfLaT2012):* 35–40.