

DanProof: Pedagogical Spell and Grammar Checking for Danish

Eckhard Bick

University of Southern Denmark

eckhard.bick@mail.dk

Abstract

This paper presents a Constraint Grammar-based pedagogical proofing tool for Danish. The system recognizes not only spelling errors, but also grammatical errors in otherwise correctly spelled words, and categorizes errors for WORD-integrated pedagogical comments. Possible spelling corrections are prioritized from context, and grammatical corrections generated by a morphological module. The system uses both phonetic similarity measures and traditional Levenshtein-distances, and has a special focus on compounding/splitting errors common in modern Danish. As a classical spell-checker DanProof achieves F-Scores over 95, and F=88 if compounding correction is included. With the maximal set of error types, 2/3 of all errors are found in school essays, and precision is 91.7%.

1 Introduction

Spell- and grammar-checking is not a new task, and is integrated in many standard text editors for the major languages. However, smaller languages are not so well covered, and the technology is very much inspired by what works for English where simple list checking will identify non-words, and correction suggestions can be found with the editing distance measure using the same list. However, the task is more difficult for morphologically rich languages, where word formation is too productive to allow lists with good coverage. A special problem for Danish is compounding, and standard, English-style spell checkers tempt users to (wrongly) split compounds into their parts just to satisfy their spell-checker. This phenomenon can now lead to a general tendency towards compounding errors in especially informal writing in Danish. Two other problems also deserve special attention: First, many errors are grammatical in nature rather than misspellings, and will lead to

words that do exist in the spelling lexicon, an example being the confusion of finite and non-finite verb endings in Danish (*købe - køber*), which is considered a stigmatizing marker of low-level education. Detecting this error is only possible with context and true sentence analysis. Second, depending on the user group, it is not enough to come up with a loose list of similar words as correction suggestions - only good spellers will immediately see what the correct form is. Bad spellers need a well-prioritized list, or - if possible - just one suggestion, which is also desirable for tasks in automatic tool pipes, such as pre- and postprocessing of machine translation (Stymne & Ahrenberg 2010) or as an OCR module. To achieve such prioritization, simple editing distance is not enough. Rather, other factors, like phonetic similarity, compound-part similarity, frequency and not least context analysis, must be considered.

While initiatives like *hunspell* and the use of finite state transducers (Pirinen & Lindén 2014; Antonsen 2014), have addressed the variability of morphologically rich languages, the use of full-scale grammatical and sentence analysis is rare. For the Scandinavian languages, the Constraint Grammar (CG) approach (Karlsson et al. 1995) has been used for this task (Arppe 2000; Birn 2000; Carlberger et al. 2004 for Swedish; Hagen et al. 2001 for Norwegian), and working systems are distributed by the Finnish company Lingsoft Oy (www.lingsoft.fi). For Danish, a CG-based spell- and grammar-checker for developed with a special focus on dyslexics (Bick 2006), and it is this system, that is the point of departure for our current work. In the following we will show how our own approach makes use of morphological and syntactic analysis for both the task of detecting errors and the task of weighting correction suggestions.

2 System description

DanProof can be used as (a) a command-line tool for corpus work, research or automatic spell-checking of e.g. texts for machine translation, or (b) an end user application with Word-integration and pedagogical comments. The linguistic core consists of four modules, (1) word based spell checking and similarity matching, (2) morphological analysis of words, compounding and correction suggestions, (3) syntax-based disambiguation of all possible readings, and (4) context-based mapping of error types and correction suggestions. In the current version, levels (3) and (4) are actually run several times, first safe error mapping followed by loose morphological disambiguation, then full error mapping followed by strict morphosyntactic disambiguation, and finally a last round of error mapping exploiting syntactic function tags and (implicit) dependencies. Gender or number agreement errors between determiners, adjectives and nouns in an np are a good example for why this is useful: If no error mapping is performed before disambiguation, the latter may have removed an agreement-conflicting noun reading in favor of a verb reading already once the rule is run. On the other hand, disambiguated context may be necessary to decide which word, out of a string of conflicting words, should be tagged as wrong. Finally, long distance agreement, as between subject and subject complement, can only be safely resolved once syntactic relations are established.

2.1 Classical spell-checking and similarity matching

After tokenization, this is the first module of our pipe and represents a classical spell-checker. The error finder appends weighted lists of correction suggestions to tokens that either figure in a manually compiled error substitution list (5,800 entries), or that cannot be verified in the fullform lexicon (1,100,000 word forms). The substitution list allows both single- and multi-word forms, as well as variable word parts, and provides ready-made, similarity/likelihood-weighted corrections. To find correction matches from the fullform database, a special matching algorithm was developed, using partial-match databases rather than the full list (which would mean a prohibitive time consumption). The process is then repeated with a phonetically transcribed version of the database. Common permutations, gemination and mute letters are

taken into account, and in a novel approach, consonant and vowel "phoneme skeletons" are matched (e.g. 'straden' – *stdn/áè*). Next, the Comparator computes grapheme (w=written), phoneme (s=spoken) and frequency (f) weights for each correction candidate, using, among other criteria, word-length normalized Levenshtein distances. The different weights are combined into a single similarity value (with 40% below maximum as a cut-off for the correction list), but a marking is retained individually for the highest graphical, phonetic and frequency match value.

2.2 Tagger/parser-based word ranking

It is a core feature of our methodology that the ordinary rule body of a CG parser is used to choose the contextually most acceptable word from a list of correction suggestions. Thus, the best correction candidates are submitted to morphological analysis on par with the original word form, and the result used as input for the tagging stage¹ of the DanGram parser² (Bick 2001), whose about 6,000 rules, with their implicit contextual and semantic knowledge, will hopefully sort out the added ambiguity and single out the correct suggestion³. Too much ambiguity, however, can overwhelm the system, and with multiple errors in the same sentence, contexts become as ambiguous as the to-be-disambiguated word itself and may prevent the CG rules from working properly. Therefore, only the top-ranking correction suggestions are used and the most heuristic (= least safe) rules are excluded at this stage. For DanProof, we also added disambiguation rules specifically targeting spell-checker-suggested forms, and to be run *before* DanGram proper.

Unlike the original version of the spell-checker (called OrdRet, www.ordret.com), we are targeting not dyslexics' text, but ordinary text, or even pre-spellchecked text, with a lower error ratio, and expect edit distances between error and correction to be lower than for dyslexics.

¹ This stage disambiguates part of speech and morphology, but uses syntax only implicitly, avoiding the stricter disambiguation forced by the subsequent function-assigning syntax module.

² A public version of the tagger is accessible for teaching and research through SDU's VISL project [visl.sdu.dk/visl/da/parsing/automatic/]

³ In the correction menu shown to the user, this will then be the number-one suggestion. The other readings will be "resurrected" and appended in the order of their original spellchecker ratings.

Therefore, we were able to use stricter similarity thresholds, resulting in shorter suggestion lists, less ambiguity for the tagger, and more cases with the correct suggestion as first alternative. Fig. 1 illustrates the interplay between the core spell-checker module, DanGram's morphological analysis and disambiguation and the error mapping CG module. Simplified output examples for the individual modules are shown in rectangular text boxes⁴.

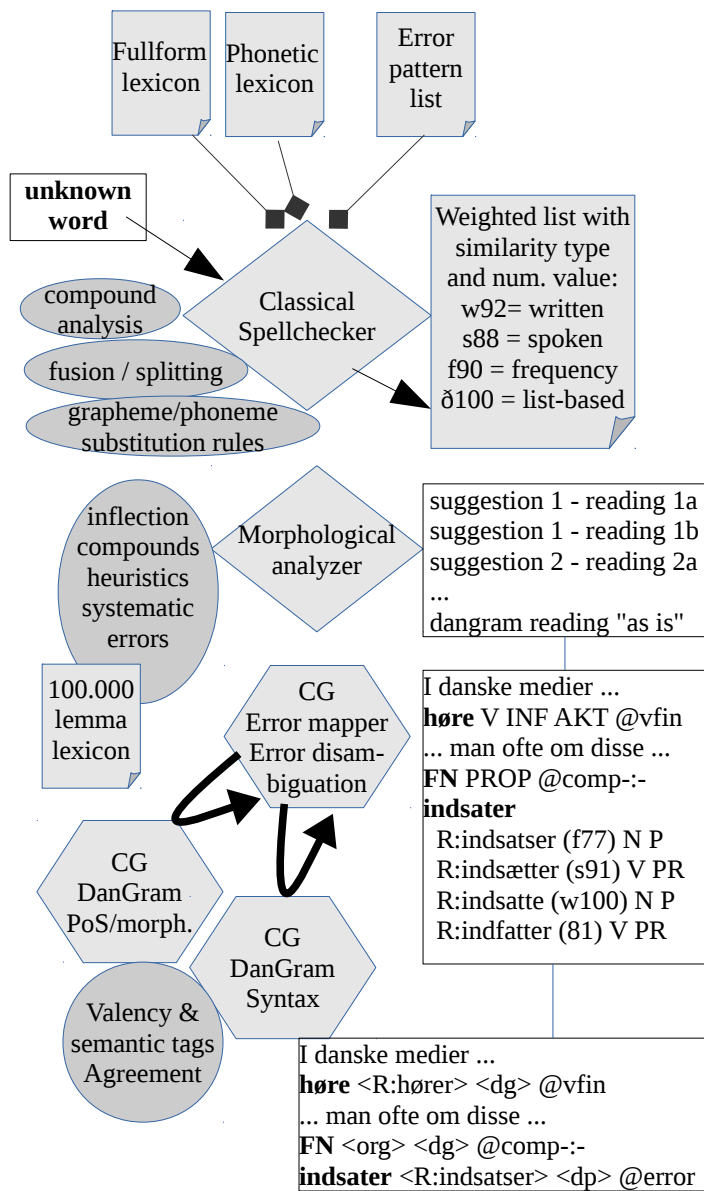


Fig. 1: System architecture

⁴ The literal translation of the Danish example sentence is "In Danish media **hears** one often about these **UN initiatives**." R:... -expressions contain (ambiguous) correction suggestions. V=verb, INF=infinite, AKT=active, PROP=name, N=noun, P=plural, @vfin=finite verb, @comp=compound error

2.3 Morphological recognition

An important difference between our target data and dyslectics texts is lexical variation and word complexity. Thus, we found a much higher percentage of long words and compounds, and there was a higher risk of an "unknown" word in fact being correct rather than an error. Therefore, we extended the compound analysis module of DanGram as well as its heuristic, endings-based morphological word guesser. We also added a confidence tag for "good compounds", based on length and frequency of the compound parts. In the current version, these alternative analyses compete with possible error corrections and their tags are used to make CG rules more cautious, avoiding false positive classification of compounds or rare technical terms as errors.

Finally, we also wished to accommodate systematic errors made by immigrants or foreign language learners in Denmark, in particular endings errors due to category confusions⁵ (e.g. noun gender, regular past tense inflection) or special orthographic rules, such as e-elision for inflected -el/er/en-words ('ministere' -> 'ministre', plural of 'minister'). We therefore modified DanGram's analysis module to recognize and mark this kind of error. Together with the phonological and grapheme confusion tables used by the word similarity module, these cases cover many of the non-semantic L2 learner error types described by Hammarberg and Grigonytė (2014) for Swedish⁶, though obviously not code switching or compounding loans. In order to effectively address the latter, L1-specific rule modules or substitution lists would have to be added.

2.4 Context-based error mapping

The next stage of the system is a dedicated error-driven Constraint Grammar (ca. 1450 rules) that maps grammatical errors on otherwise correctly spelled words. While DanGram is basically reductionist and removes (focuses) ambiguity, the error-CG *adds* information. For instance, the common Danish '-e/-er' verb-error (infinitive vs.

⁵ Unlike English, Danish has 2 grammatical genders and two regular past tense endings, which do not follow strict patterns, and have to be learned together with the word.

⁶ This study uses the ASU learner corpus. No corresponding data exist for Danish, but since the two languages are closely related, the inventory of error types can be assumed to be the same or at least very similar.

present tense, cf. example (b)) can often be resolved by checking local and global left context (infinitive marker, auxiliaries, subject candidates). Likewise, gender and number errors can be checked by noun phrase context (examples a,d). Suggestions are mapped⁷ as @-tags in the style of CG syntactic tags, e.g. @pl (plural), @vfin (finite verb) or @utr (common gender). In the examples below, rule conditions are paraphrased in parentheses. DanProof's last stage generates corrected wordforms <R:....> from these inflectional tags, and in Word's graphical user interface, the tags are "translated" into error types and expanded with explanations and examples (see footnote⁸ for translations).

- (a) Det er også disse **menneske** (@pl <R:mennesker>) der mener ... (*noun phrase agreement: plural determiner*)
- (b) 25 procent af alle voksne danskere **leve** (@vfin <R:lever>) i en **kerne** (@comp-) familie. (*subject candidate to the left, absence of infinitive-triggering contexts such as auxiliaries*)⁹
- (c) Hun besøgte **barndoms** (@comp-) veninden. (*indefinite singular noun in the genitive, immediately preceding definite noun*)
- (d) Det var en **stort** (@utr <R:stor>) oplevelse. (*noun phrase agreement*)
- (e) Bægeret var fuld (@sc-neu <R:fuldt>). (*long-distance agreement between subject and subject complement*)
- (f) Det har **vært** (@error <R:været>). (*'været' V wins over 'vært' N after auxiliary*.)
- (g) Hun ønsker ikke **og** (@:at) hjælpe. (*infinitive to the right, infinitive-triggering verb to the left*)

Of course, not all errors are based on wrong inflection. Thus, the rules also mark casing, sentence separation, apostrophe and hyphenation

⁷ Possible multiple mappings will be sorted out by subsequent contextual disambiguation rules.

⁸ (a) It is also these **people** that think ..., (b) 25 percent of all adult Danes **live** in a **nucleus** family, (c) She visited [the/her] **childhood** friend, (d) It was a **great** experience, (e) [The] cup was full, (f) It has **been** ..., (g) She does not want **to** help

⁹ In the real rule, there are 5 different negative contexts, for safety, as well as various other conditions.

errors, as well as word insertion and deletion, and fusion/splitting errors (cf. @comp- in example (b-c), all of which are not normally treated - or not treated well - by commercial spell-checkers. Finally, individual word substitution rules are added in a contextual way, where general, list based suggestions would have been too risky. While OrdRet only used tags for this (e.g. @:at in example (g)), we are also using APPEND rules for the same purpose in DanProof. APPEND rules are a relatively new feature in CG, implemented in the CG-3 compiler (Bick & Didriksen 2015), and add complete new reading lines *after* morphological analysis. Thus, we can include new tags, such as PoS and inflection, for the correction word and allow the disambiguation rules to compare the suggested form to the original one with regard to context compatibility.

One problem with inflectional error mapping is DanGram's disambiguation, which may well discard correct forms for the sake of erroneous ones if the context also contains erroneous forms. Thus, it may not be possible to re-map a finite verb as infinitive, because the same context that would allow the error-CG to do this, may have led DanGram to discard the verb-reading altogether if the word form as such (or any of its correction suggestions) was, say, a noun or adjective. Therefore, the safest error-mapping rules are run twice – both before and after DanGram. As "before"-rules they may apply while the necessary context is still in place, avoiding disambiguation interference. Run again as "after"-rules, the same rules may capture other necessary contexts that have been made safe by DanGram in the meantime, allowing the rules find and mark further errors.

Finally, there is a second, syntactic run (5,000 rules) of DanGram and a third round of error-mapping exploiting the syntactic tags, as does the subject complement rule in example (e) - as opposed to the "easier" noun phrase agreement error (d).

2.5 Pedagogical comments on error types

A major difference between OrdRet and DanProof, besides the target group adaptations, is the fact that the latter makes use of its error classification for pedagogical purposes. Each error that is not just a simple spelling error comes with a (short) definition and a (longer) explanation, as well as examples and links to

external material such as on-line exercises and text book excerpts. All in all, about 35 error types are covered.

Error type	@inf
Definition	infinitiv (navnemåde)
Explanation	Du har sandsynligvis tilføjet et overflødig -r til en infinitiv, der dermed bliver til et finit verbum. En vigtig regel er at et verbum (udsagnsord) er en ubøjet infinitiv (uden -r), hvis der til venstre står 'at' eller vil/ville, kan/kunne, skal/skulle, bør/burde. Omvendt ...
Examples	De begynder <u>at</u> danser [danse] 'Han <u>forstår</u> engelsk' - 'Han <u>kan</u> forstå engelsk'
Links	En mulig øvelse er <i>R-problemer - verber</i> , samt VISL's grammatikspil <i>Balloon Ride</i> .

Table 1: Pedagogical comment fields (see footnote¹⁰ for translations)

An added advantage from making error types transparent to the user, rather than just marking words as "wrong", is that the user can actively switch certain error types on or off. For a good speller with a good grasp of grammar, for instance, a high proportion of grammatical error markings will be false positive, while a lone false positive may be a fair price for a bad speller to pay for ridding himself of a dozen errors on the same page. Having an on/off setting for grammatical errors on a whole, or individual ones, remedies this problem. Similarly, some users employ uppercasing for emphasis, or prefer English-inspired apostrophes for names, and if this is a conscious decision, marking it only antagonizes the user.

A known problem with Danish orthography is that erstwhile errors often become allowed forms, and may even become the only allowed form, if sufficiently many people make the error. On the other hand, many individuals stick to the originally learned spelling over a life time. Therefore, DanProof adds markers (<frequent>, @green) for "wrong but widely used" forms,

¹⁰ Explanation: You have probably add a superfluous -r to an infinitive, thereby turning it into a finite verb. An important rule is that a verb is a non-inflected infinitive (without -r), if the words 'to' or 'will/would', 'can/could', 'shall/should' can be found to the left. Conversely, ..., Examples: The begin to **dances** [dance]; He understands English - He can **understand** English; Links: A possible exercise is *R-problems - verbs*, and VISL's grammar game *Balloon Ride*

making possible an on/off-switch for "strict" spelling errors only.

2.6 The graphical user interface

DanProof has a graphical user interface integrated into Microsoft Word, with side bar fields for error-marked paragraphs and dynamic comment fields. In the main text window, optional colored underline marking can be activated, mimicking Word's own "correct spelling while writing" mode.

3 Evaluation

To evaluate the performance of DanProof, we looked for texts that would have some errors but not as many as dyslectics' texts, and not as few as published texts. High school exam texts seemed to be a good compromise and we decided to use Danish high school exam essays by Greenlandic speakers (Bæk et al. 2009). The essays (6632 words) were analyzed with DanProof and error markings inspected and corrected manually. In a second round of inspection false negatives were added, i.e. errors the system hadn't found. The texts did contain both ordinary spelling errors¹¹ and grammatical errors, but also many confusion spelling errors, i.e. errors where a word is replaced by another (wrong) word, but with the correct spelling (e.g. 'det' -> 'de'). We therefore computed performance at four different levels:

- All error markings
- Spell: Only spelling errors, excluding grammatical errors, but including compounding errors (fusion/splitting), hyphen and case
- Lex: Same as Spell, but not counting false positives if the word is not listed in *Retskrivningsordbogen* (e.g. 'fucked', 'adj') and not counting false negatives if the word does exist in *Retskrivningsordbogen* (e.g. 'da' [dag], 'single' in compounding errors)
- Classic: Same as Lex, but words are counted as error-marked, if DanProof marked them as unknown, yet feasible compounds

¹¹ This is not always the case nowadays because students use Word's list-based spell checker while writing, so students will change an un-accepted word until it matches an existing word - leaving only confusion errors, compounding errors and grammatical errors.

	Recall	Precision	F-score
All	65.1	91.7	76.1
Spell	86.8	90.8	88.6
Lex	93.7	96.7	95.2
Classic	100.0	98.3	99.1

Table 2: Error detection performance, school essays

As can be seen from the table, DanProof is very reliable if used as a traditional spell-checker (Classic and Lex), even when the more difficult task of compounding correction is added for otherwise correctly spelled words (Spell). With the full range of error types, precision is still acceptable (even a little higher than for "Spell"), but recall is lower - DanProof misses out on about 1/3 of all errors of the addressed type.

Qualitative error analysis of false negatives showed that particularly difficult error types, recall-wise, are @insertion (i.e. missing words) and deletion (@nil). Confusion without grammatical motivation (@:...) was rarely spotted, but this is probably data-specific for the Greenland setting. Thus, 1/3 of the cases were confusion of the subject pronouns 'det' and 'de' which are hard to distinguish contextually, plus cases outside of DanProof's current scope, e.g. idioms and choice of preposition.

	Recall	Precision	F-score
@error (47)	83.0	95.1	88.6
@upper (28)	100.0	96.6	98.3
@comp- (25)	76.0	100.0	86.4
@comp-:- (22)	90.9	95.2	93.0
@nil (14)	28.6	100.0	44.5
@insert (12)	8.3	100.0	15.3
@vfin (9)	66.7	85.7	75.0
@: (35)	5.7	50.0	10.23
e.g. @:de (10)			
@pl (8)	62.5	83.3	71.4
@utr (7)	100.0	87.5	93.3
@def (4)	75.0	60.0	66.7
@new (3)	100.0	60.0	75.0
@neu (6)	16.7	100.0	28.6
@idf (4)	25.0	50.0	33.3
@lower (4)	75.0	100.0	85.7
@inf (4)	100.0	100.0	100.0

Table 3: Error type-specific performance

A direct comparison with OrdRet is difficult because of the different target domains, and because the OrdRet evaluation by Bick (2006) evaluated correction suggestion priority lists,

rather than simple matches, and weighted correction suggestions with their inverse rank in the list. If a weighted score is approximated by assigning a weight of zero to all cases where the correct form was not matched, DanProof does get better scores for its essay texts than OrdRet had for its dyslectics texts¹², although OrdRet has a "performance reserve" because of the presence of correct suggestions at lower list ranks.

	R	P	F-score
All-weighted (DanProof)	61.6	86.7	72.0
All-weighted (OrdRet)	43.0	58.0	49.4

Table 4: Comparison OrdRet - DanProof

As a real-life control, we used MicrosoftWord 2007 on the same essays, and found considerable differences, both in scope and performance. First of all, Word does not find compounding errors and can't recognize names, the former creating false negatives, the latter false positives. It does even worse than DanProof on deletion and insertion, and it marks relatively few grammatical errors, albeit almost without false positives. In a direct comparison, this leads to very low - and unfair - scores¹³ for the "all"-evaluation due to low recall. For "spell" and "lex", however, Word still finds considerably fewer errors than DanProof. Precision is better without counting names, but is still hampered by the missing compound analysis (e.g. *kønstradition* [gender tradition], *boginteresse* [book interest], *livsrygsæk* [life backpack], *middagsræs* [noon rush]).

	Recall	Precision	F-score
All	20.8	54.6	30.1
All-nonprop	20.8	71.6	33.1
Spell	75.0	51.1	60.8
Spell-nonprop	75.0	70.3	72.6
Lex	81.8	54.9	65.7
Lex-nonprop	81.8	77.6	79.6

Table 5: Word2007 performance

Once DanProof recognizes a word as wrong, the assigned error type is usually reliable (95.7% for "all", 96.6% with "spell" settings). For the

¹² A more direct comparison by running both systems on the same data was not possible because the original OrdRet setup could not be reconstructed.

¹³ On the other hand, Word marked some simple spacing and punctuation errors that were not in the scope of our DanProof test.

correct error type markings, the suggested new word form was correctly chosen in 95.8% of cases, independently of "all" or "spell" settings. Word had a correct suggestion in 84.4%, and this was offered as the first choice in 68.9%, indicating that DanProof's context-based prioritization does make a difference.

Since the density of errors to be found is very much dependent on genre and text authors, an alternative measure of "experienced performance" is the number of false positives or false negatives *per page*¹⁴. Thus, for our essays, DanProof had 0.7 false positives per page with the 'all'-settings, and 0.4 false positives per page with 'spell' settings. For false negatives, the numbers were 4 and 0.4, respectively.

DanProof uses the tag @new, if it deems a word correct, but has done so using productive compound analysis. Conversely, @check! is used for words that are not "safely wrong" because no correction alternative was found, but that are more likely to be wrong than @new, because no productive analysis was found either. In a 178,000 word newspaper corpus chunk from Korpus2000 (...), @new was used 347 times, and was wrong on only 2 occasions (99.4% accuracy). Confronted with the same word list, Word2007 had false positives in 54.2%, evidently due to not having a compound analysis module. @check! was used 120 times and proved to be a very mixed category, with 23.3% spelling errors, 17.5% foreign words and 8.4% names (mostly lowercase brands, pharmaceuticals etc.), i.e. less about half were ordinary Danish words. Word2007 accepted 1/3 of the latter as correct, indicating DanProof would profit from a larger lexicon to supplement its compound analysis. Still, in a hybrid setup, given that the @new category is safe and 3 times bigger than the @check category, and that Word rejected half of the former, Word would probably benefit more from DanProof input than vice versa. In any case, the two systems' strengths seem to be in different areas, which would make hybridization, maybe with an arbiter system, a good idea.

¹⁴ Lingsoft, for instance, claims less than 1% false positives per page for their products [http://www.lingsoft.fi/en/506, 19 Apr 2015]

4 Conclusion and outlook

We have described how a Constraint Grammar environment can be used to enhance a classical spell-checker module in a number of ways:

- weighting of correction suggestions for non-words and dubious words
- reduce the number of false positives through compound analysis and name recognition
- mapping and classification of grammatical errors
- syntactic validation of split compound recognition

For its target domain, the system achieved better recall and precision than its predecessor system (OrdRet) and outperformed MicrosoftWord's standard spell-checker, not least with regard to false positive non-word marking, split compounds and grammatical error-typing. For correctly typed errors, the right correction alternative was chosen in over 95% of cases. However, performance for grammatical, conditioned errors is not on par with the system's accuracy for classical spell-checking, and should be improved.

Transparent error-typing and confidence grading (@error, @new and @check!) allowed us to add pedagogical comments, but at the time of writing graphical integration into MicrosoftWord was not finished, and should be followed up by classroom testing and teacher feed-back, possibly integrated with existing didactical tools.

While word-based grammatical errors such as agreement errors and the so-called -r errors are well-covered, further syntactical error types should be added, such as word order errors and comma-checking. The latter is a sensitive, almost political, issue in Denmark, and should definitely be part of a Danish proofing suit, but is being addressed by a parallel R&D project, and therefore not evaluated here.

References

- Antonsen, Lene. 2014. Evaluation of a North-Saami FST-Based Spellchecker Program. Presentation at SLTC 2014 [http://divvun.no/workshops/NorWEST2014/presentations/Antonsen.pdf]

- Bick, Eckhard. 2001. En Constraint Grammar Parser for Dansk. In Widell, Peter & Kunøe, Mette (eds.), *8. Møde om Udforskningen af Dansk Sprog, 12.-13. oktober 2000*, p. 40-50. Århus: Århus University.
- Bick, Eckhard. 2006. A Constraint Grammar Based Spellchecker for Danish with a Special Focus on Dyslexics". In: Suominen, Mickael et.al. (ed.) *A Man of Measure: Festschrift in Honour of Fred Karlsson on his 60th Birthday*. Special Supplement to SKY Journal of Linguistics, Vol. 19. pp. 387-396. Turku: The Linguistic Association of Finland
- Bick, Eckhard & Didriksen, Tino. 2015. CG-3 - Beyond Classical Constraint Grammar. In: Beáta Megyesi: Proceedings of NoDaLiDa 2015, Vilnius. pp. 31-39. Linköping: LiU Electronic Press
- Birn, Jussi. 2000. Detecting grammar errors with Lingsoft's Swedish grammar checker. In Nordgård, Torbjørn (ed.) *NODALIDA '99 Proceedings from the 12th Nordiske datalingvistikkdager*, p. 28-40. Trondheim: Department of Linguistics, University of Trondheim.
- Bæk, Jan & Elmo, Agnete & Olesen, Claus & Hartmann, Peter. 2009. Evaluering af skriftlig eksamen for Dansk i Grønland [http://www.uvm.dk/Uddannelser-og-dagtilbud/Gymnasiale-uddannelser/Information-til-censorer-paa-de-gymnasiale-uddannelser/~media/UVM/Filer/Udd/Gym/PDF11/Proever_og_eksamen/Censorvejledninger_dansk_maj_2011/110504_14.ashx] and [[http://www.iserasuaat.gl/fileadmin/user_upload/Te st_files/Raad_og_vink_Groenland_2009.doc](http://www.iserasuaat.gl/fileadmin/user_upload/Te_st_files/Raad_og_vink_Groenland_2009.doc)]
- Carlberger, Johan & Domeij, Rickard & Kann, Viggo & Knutsson, Ola. 2004. The development and performance of a grammar checker for Swedish: A language-engineering perspective. *Natural Language Engineering*, 1 (1).
- Hagen, Kristin & Lane, Pia & Trosterud, Trond. 2001. En grammatikkontrol for bokmål. In Vannebo, Kjell Ivar & Helge Sandøy (eds.) *Språkknyt 3-2001*, p. 6-9, 47. Oslo: Norsk Språkråd
- Hammarberg, Björn & Grigonytė, Gintarė. 2014. Non-Native Writers' Errors - a Challenge to a Spell-Checker. Presentation at SLTC 2014. [http://divvun.no/workshops/NorWEST2014/abstracts/Hammarberg_Grigonyte.pdf]
- Karlsson, Fred & Voutilainen, Arto & Heikkilä, Jukka & Anttila, Arto. 1995. *Constraint Grammar: A language-independent system for parsing unrestricted text*, pp. 1-88. Berlin: Mouton de Gruyter.
- Pirinen, Tommi A. & Lindén, Krister. 2014. State-of-the-Art in Weighted Finite-State Spell-Checking. In: Proceedings of CICLing 2014.
- Stymne, Sara & Ahrenberg, Lars. 2010. Using a Grammar Checker for Evaluation and Postprocessing of Statistical Machine Translation. In: Proceedings of LREC 2010.