Eckhard Bick

A Constraint Grammar Based Spellchecker for Danish with a Special Focus on Dyslexics

Abstract

This Paper presents a new, Constraint Grammar based spell and grammar checker for Danish (OrdRet), with a special focus on dyslectic users. The system uses a multi-stage approach, employing both data-driven error lists, phonetic similarity measures and traditional letter matching at the word and chunk level, and CG rules at the contextual level. An ordinary CG parser (DanGram) is used to choose between alternative correction suggestions, and in addition, error types are CG-mapped on existing, but contextually wrong words. An evaluation against hand-marked dyslectic texts shows, that OrdRet finds 68% of errors and achieves ranking-weighted F-Scores of around 49 for this genre.

1. Introduction

The progressively more difficult task of spell checking, grammar checking and style checking has been addressed with different techniques by all major text processors as well as independent suppliers. However, not all languages are equally well covered by such resources, and their performance varies widely. Also, spell checkers do not usually cater for a specific target group or user context. For Scandinavian languages, the Constraint Grammar approach (Karlsson & al. (eds.) 1995) has been used by several researchers to move from list-based or morphologically rule-based to context-based spell and grammar checking (Arppe 2000 and Birn 2000 for Swedish; Hagen & al. 2001 for Norwegian), and has led to implemented systems distributed by Lingsoft (either integrated into MS Word or as stand-alone grammar checkers under the tradename of Grammatifix).

For Danish, though already burning brightly in Lingsoft's spell- and grammar-checking modules for MS Word, the CG torch has recently been taken up once more by a consortium consisting of DVO (Dansk

A Man of Measure
Festschrift in Honour of Fred Karlsson, pp. 387–396

Videnscenter for Ordblindhed), Mikro Værkstedet and GrammarSoft, and applied to one of the most challenging tasks of all—correcting dyslexics' texts, where Constraint Grammar was used not only for a tighter integration of grammar-checking already at the spell-checking level, but also to create a more efficient ranking system for multiple correction suggestions. The resulting system (OrdRet) has experimented with a number of novel design parameters which will be described in this paper.

2. Why a word list is not enough

Even a traditional, simple list-based spellcheck works quite well for experienced language users that make few and isolated errors. There are, however, a number of problems with the list approach, which can only be solved by employing linguistic resources:

- A full form list is basically an English brain child in the first place. For languages like Danish or German, productive compounding prevents lists from ever being complete (e.g. *efterlønstilhænger*, *kostkonsulent*), and make deep morphological analysis necessary.[1] In fact, Danish children sometimes misspell compounds as separate words just to satisfy their spell checker where it won't accept the compounds.

- Words accepted by list-lookup may still be wrong, *in context,* due to homophone errors, inflexion errors, compound splitting, agreement or word order. This is where spell-checking, in a way, means grammar-checking—syntax being not the object, but the vehicle of correction.

Especially dyslexics or other "bad spellers" may have difficulties in choosing the correct word from a list of correction suggestions. For this target group, a reliable ranking of suggestions is essential:

- For similarity ranking, sound may be as important as spelling, making necessary a phonetic dictionary—and a transcription

[1] Most CG systems, including the ones mentioned above targeting spell-checking, use morphological analyzers that handle inflexion and compounding in a rule-based way.

algorithm as such, because misspelled words can't be looked up in a dictionary

- Some words are simply more likely than others (*lagde* > *læge* > *lage*), and good corpus statistics may help avoiding very rare words outranking very common ones.

- Even words with a high similarity may be meaningless in context (*hun har købt en lille hæsd* [*hæst*|*hest*]) for syntactic or semantic reasons

3. System design

OrdRet is a full-fledged Windows-integrated program, with a special GUI that includes text-to-speech software, a pedagogical homophone database with 9,000 example sentences, an inflexion paradigm window etc. However, in this paper we will be concerned only with the computational linguistics involved, assuming token-separated input and error-tagged output. This linguistic core consists of four levels, (a) word based spell checking and similarity matching, (b) morphological analysis of words, compounding and correction suggestions, (c) syntax based disambiguation of all possible readings, and (d) context-based mapping of error types and correction suggestions.

3.1 Word based spell checking and similarity matching

The Comparator program handling this level appends weighted lists of correction suggestions to tokens it cannot match in a fullform list (ca. 1,050,000 word forms). First, in-data is checked against a manually compiled error and pattern list (5,100 entries), then against a statistical error data base (13,300 entries). The former was compiled by the author, the latter by Dansk Videnscenter for Ordblindhed, based on free and dictated texts from school age and adult dyslexics (ca. 110,000 words). Both lists provide ready made, weighted corrections. Weight in the data driven list are expressed as probability ratios depending on the frequency of one or other correction being the right one for a given error in context. Multi-word matches are allowed and possible word fusion is also checked against the fullform list.

Time and space complexity issues prevent a deep check on the whole fullform list, but for still unresolved words (the majority), the Comparator then selects correction candidates from specially prepared databases, of which one is graphical, and the other phonetic. Common permutations, gemination and mute letters are taken into account, and as a novel technique, so-called consonant and vowel skeletons are matched (e.g. '*straden*'—*stdn/áè*). Next, the Comparator computes grapheme, phoneme and frequency 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 point for the correction list), but a marking is retained for the best graphical, phonetic and frequency matches individually (e.g. s=spoken, w=written, f=frequency).

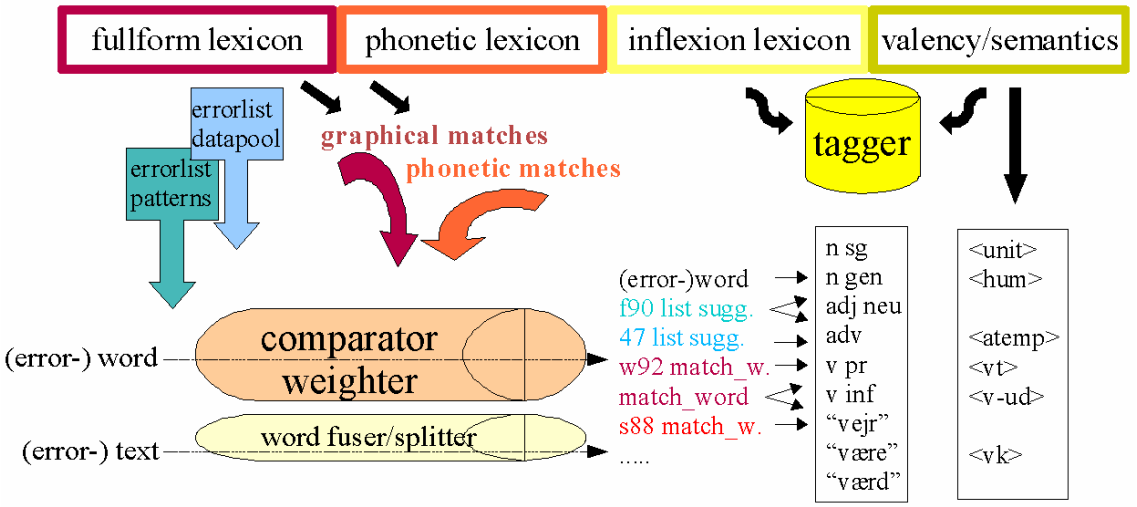

Figure 1. The anatomy of OrdRet 1

3.2 Using a tagger/parser for word ranking

A central idea when launching the OrdRet project was to use a pre-existing well-performing CG-parser for Danish (DanGram, Bick 2001) to select contextually good and discard contextually bad correction suggestions from a list of possible matches. DanGram achieves F-scores of over 99% for PoS/morphology and 95–96% for syntax, but ordinarily assumes *correct* context. However, since our dyslectic data indicates error rates of 25% (!), only the more stable PoS stage was used, where syntax is implicit (as disambiguating rule context), but not explicited for its own sake. Even so,

correction lists had to be truncated at 4–5 words for the tagger run, to limit contextual ambiguity.[2] As a by product, DanGram's mophological analyzer stage delivered its own reading for the erroneous word as such,[3] which was allowed to compete with the correction suggestions, often providing a good composite analysis or semantically classifiable proper noun not (yet) found in OrdRet's fullform list.

Since CG is a reductionist method, DanGram will make its choice by letting only one reading survive. In practice OrdRet then re-appends all other suggestions as number 2.3... etc. according to their original weights and user preferences as to list length. The use of DanGram also provides a solution to the high risk of false positive corrections from those cases where the error data-base contains otherwise correct forms used instead of other correct forms. Here, both error marking and correction list are removed if the original token ranks highest after the DanGram run.

3.3 Context-based mapping of grammatical errors

Apart from the DanGram tagger-parser, OrdRet also uses a dedicated error-driven Constraint Grammar (ca. 800 rules) to resolve correction ambiguity, and—most important—to map grammatical errors on otherwise correctly spelled words. While DanGram basically removes (focuses) information, the error-CG *adds* information. For instance, the common Danish '-e/-er' verb-error (infinitive vs. present tense) can often be resolved by checking local and global left context (infinitive marker, auxiliaries, subject candidates). Likewise, adjective gender or number errors can be checked by long, left syntactic relations (subject predicatives) or short, *right,* syntactic relations (agreement with NP head nouns). Suggestions are mapped as @-tags in the style of CG syntactic tags (@inf, @vfin, @neu, @pl), allowing later disambiguation in the case of multiple mappings. In the commercial version of OrdRet, these error types are invisible to the user, and a morphological generator is used to create traditional correction suggestions instead (i.e. full forms). A number of rules map corrections on

[2] With a Danish morphological/PoS ambiguity of about 2 readings pr. word, this makes for a cohort of 8-10 readings to be considered for each error token. Also for reasons of 'ambiguity flooding', only certain error-prone homophones were allowed to compete with otherwise correct words at this stage—not OrdRet's complete database of about 9,000 homophones.

[3] OrdRet also uses DanGram's analyzer to give a user recommendations whether to append an unknown word to its lexicon of "user's own words".

individual words (@:suggestion) in a contextual way, where general, list based suggestions were deemed too risky and ambiguity-prone.[4]

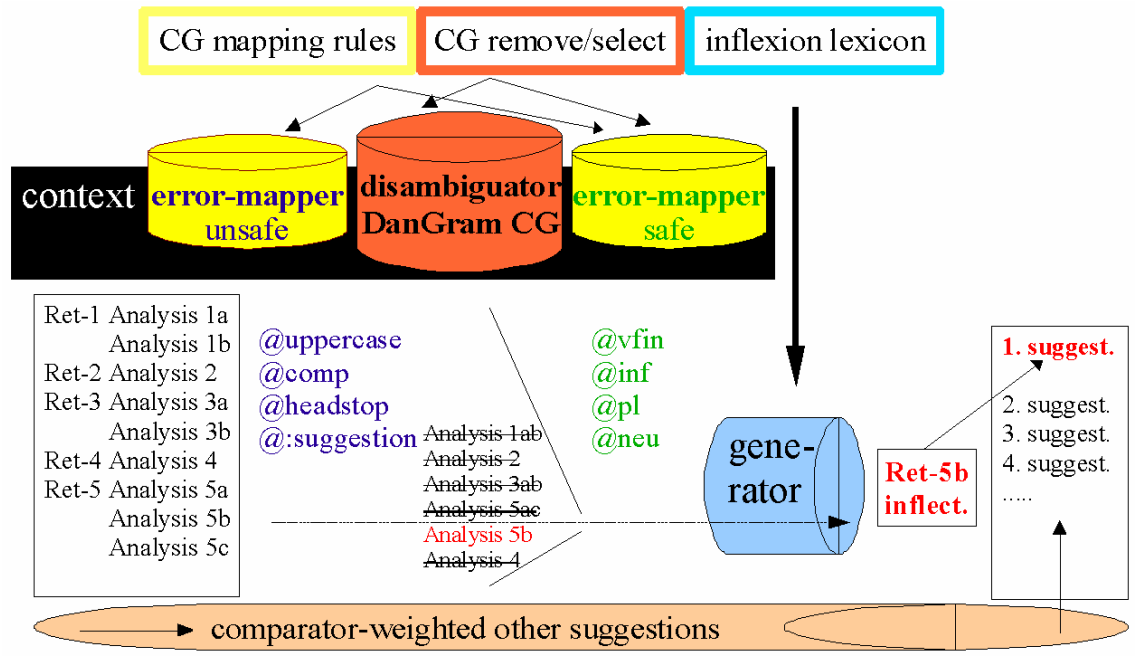

Fig. 2. The anatomy of OrdRet 2

One problem with error mappings is the conflict with 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. As a solution, the error-mapping rules with the lowest heuristicity (i.e. the safest ones) are run twice—both before and after DanGram. Thus, "before"-rules may apply while the necessary context is still in place, avoiding disambiguation interference. On the other hand, the same rules are tried again as "after" rules, together with more heuristic rules, since by that time some safe context conditions may have been instantiated by DanGram, allowing more rules to work.

[4] The error-CG also suggests changes in case, adds punctuation and creates sentence windows for itself and DanGram. The latter task is all the more important for dyslexics' texts, where full stops and sentence initial upper case are often emitted, leaving only syntactic and word order hints for sentence separation.

4. Examples

Hun har en opfattelse af at **kvinde*** (@pl) er bedre til det **merster**** (R:meste). *(no indefinite singular non-mass nouns without prenominals)*

Han kan ikke **hører*** (@inf) dig. *(auxiliary verb context)*

Han ønsker ikke **og**** (@:at) forstyrre. *(infinitive right, verb with infinitive-valency left)*

Min søster er **syge plejerske*** (@comp). *(dictionary lookup)*

Hun besøgte **barndoms*** (@comp-) veninden. *(indefinite singular noun in the genitive, immediately preceding definite noun)*

Glasset var **fuld*** (@sc-neu). *(subject agreement of subject predicative)*

Jeg er **træt*** (@headstop) jeg vil hjem ... *(syntactic indicators for sentence separation)*

Det har **vært**** (R:været) en lang dag. *('været' V wins over 'vært N' after auxiliary)*

*(In actual screen mode, errors would appear with a color coding, which is here indicated by asterisks - * for green and ** for red)*

5. Evaluation

200 texts, amounting to 36,046 tokens (32,512 words), were randomly selected from DVO's hand-corrected database of dyslexics' texts, and used as test data. In the original version of this manually controlled gold standard, one word out of six was marked as wrong, but inspired by a check on OrdRet false positives, about 10% additional errors (i.e. errors not annotated correctly) could be identified in the data.

For the evaluation, OrdRet was run without its statistical error word database, but with its manually compiled pattern database. In order to be

able to evaluate ranking quality for correction suggestions, weighting points were assigned as 1/rank, i.e. 1 point if the correct suggestion was ranked highest, ½ if it was ranked second, 1/3 for third place and so on. Only the top 5 suggestions were taken into account. With these metrics, *simple recall* thus means a hit within the first five, while *weighted recall* represents the rank weighted (lower) figures. For instance, if the correct suggestion is ranked second on average, weighted recall will be 50% lower than simple recall. Though somewhat unorthodox, *weighted precision* and *weighted F-score* were calculated with the same metrics.

	simple recall	simple precision	simple F-Score	weighted recall	weighted precision	weighted F-Score
all levels (red & green)	67.9	91.7	78.0	**43.0**	58.0	**49.4**
safe mode (no green)	54.6	**99.1**	70.4	34.0	**61.8**	43.9
word level (i.e. no CG)	59.6	89.8	71.6	32.32	48.7	38.9
word level (no green)	49.1	93.4	64.4	25.2	47.8	33.0
Word 2002 (all levels)	53.5	97.3	69.1	19.7	35.7	25.4

Table 1. Performance

For comparison with a standard text editing environment, the same texts were also run through the combined spell and grammar checker module of MS Word 2002[5] (default settings). The numbers show that OrdRet is considerably better than a conventional spell/grammar checker at finding errors and, in particular, ranking correction alternatives in

[5] This module was developed by Lingsoft. However, no extensive publication could be found in the public domain on the internal architecture and performance of the Danish proofing system used in MS Word, and it must be noted, that our evaluation figures for MS Word on dyslexics' errors are only intended as a kind of base line for this particular genre. No safe comparative conclusions can be made for other text types, and the numbers do not necessarily reflect the potential of Lingsoft's tools in isolation, since both MS Word's two-step design of in loco orthographical checking and in context grammar checking, as well as possible API restrictions may have imposed restrictions on full optimisation.

dyslexics' texts (weighted recall 43.0 as opposed to Word's base line of 19.7). The price, a lower-than-optimal *unweighted* precision, is compensated for by making a distinction between safe (red) and unsafe (green) errors. In unweighted terms, the gain in recall and loss in precision is in the "green" area, while "red" errors have an unweighted precision and recall close to the base line (99.1 and 54.6, respectively). In weighted terms, all figures, both red and green for both recall and precision, are above the base line (between 60% and 115% in relative terms). Though even the context-less, word-level part of the system is better at ranking than the base line (weighted F-score of 33 as opposed to 25.4), it is here that the CG-levels have their main impact (49.4), at least for the type of texts we examined.

6. Perspectives

The system's strong point, using local and global context for correction weighting and grammar checking, is also its weak point in terms of precision, and the underlying error-mapping CG should be improved in a data-driven way. User feed-back may determine how best to balance recall and precision. User co-operation will also be essential for any attempt to tackle the sparse data problem in OrdRets error database, which so far only covers a moderate part of the lexicon and for most entries lacks the statistical clout to compute safe weighting values.[6]

So far, punctuation has only been handled in connection with sentence separation and abbreviations, but comma-checking CG rules could be implemented as a second stage, exploiting already-corrected, safer context for their mapping conditions. The comma task has certain urgency for Danish, since the language after experiencing a number of contradictory reform initiatives finally seems to settle for a grammatically inspired comma, which language users will have to relearn.

Thanks

[6] At present, a correction-list suggestion drawn from the database for a given (error) word has to be checked manually for completeness, not least for short words with many close similar, because the most similar word may not even have occurred in the data set. To a certain degree, the problem is now remedied by using homophone entries for similarity list completion.

I would like to thank my partners in the OrdRet project for their help and comments in the evaluation process, in particular Birgit Dilling Jandorf and Julie Kock Clausen from Dansk Videnscenter for Ordblindhed who have made their gold standard test data available and performed a manual comparison with MS Word. I would also like to thank GrammarSoft programmer Tino Didriksen, who has programmed the evaluation program and made the necessary changes in OrdRet's dll and the xml data files.

References

Arppe, Antti (2000) Developing a grammar checker for Swedish. In Nordgård, Torbjørn (ed.) *NODALIDA '99 Proceedings from the 12th Nordiske datalingvistikkdager*, p. 13–27. Trondheim: Department of Linguistics, University of Trondheim.

Bick, Eckhard (2001) En Constraint Grammar Parser for Dansk. In Peter Widell & Mette Kunøe (eds.) *8. Møde om Udforskningen af Dansk Sprog, 12.–13. oktober 2000*, p. 40–50. Århus: Århus University.

Birn, Jussi (2000) Detecting grammar errors with Lingsoft's Swedish grammar checker. In Torbjørn Nordgård (ed.) *NODALIDA '99 Proceedings from the 12th Nordiske datalingvistikkdager*, p. 28–40. Trondheim: Department of Linguistics, University of Trondheim.

Hagen, Kristin, Pia Lane & Trond Trosterud (2001) En grammatikkontrol for bokmål. In Kjell Ivar Vannebo & Helge Sandøy (eds.) *Språkknyt 3-2001*, p. 6–9, 47. Oslo: Norsk Språkråd.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä & Arto Anttila (eds.) (1995) *Constraint Grammar: A language-independent system for parsing unrestricted text*. Natural Language Processing 4. Berlin & New York, NY: Mouton de Gruyter.

Contact information:

Eckhard Bick
University of Southern Denmark
Rugbjergvej 98,
DK-8260 Viby J
eckhard(dot)bick(at)mail(dot)dk
http://beta.visl.sdu.dk