Peter Juel Henrichsen

MAKE EACH MORPH COUNT

A New Approach to Computational Lexicography for Text Processing

Abstract In this paper we present a newly developed formal framework (as well as its practical implementation) for automatic, lexically driven analysis of Danish text tokens. The framework (called "CLINK") employs a minimal token definition (the "morph") and a compact lexical representation (the "CLINK template"). All morphs (i.e., text elements with individual semantic contribution) are lexicalized using the same template, word forms, affixes, glue elements, punctuation marks, multi-word expressions, etc. Thus, the definition of "lexeme" is reinterpreted in functional-computational terms. The grammar rules of CLINK are purely abstract, viz. those of the Lambek calculus (categorial grammar). This paper gives an overview of the CLINK framework (motivations and application). References to performance metrics will be given (suggesting CLINK to be on a par with the Danish state-of-the-art in PoS-tagging while providing much richer annotation structure). However, we consider the formal framework in itself to be the main contribution of this short paper.

Keywords computational lexicography; language technology; text analysis; categorial grammar; CLINK

1. Introduction

Most dictionaries for human users have entries for words only, making them difficult to utilize in language technology, especially for applications aimed at text (such as grammar and spell checkers, machine translation and dialog systems). In this short paper, we propose a new approach to computational lexicography based on a generalized definition of lexeme, more specifically that any text element with an individual semantic contribution be lexicalized. We shall refer to such elements as lexical morphs (inspired by Haspelmath 2020). The class of morphs thus comprises lexical lemmas on a par with sub-verbal and non-verbal tokens (affixes, glue elements, punctuation marks, icons, and so forth). As a practical demonstration we present the text analyzer CLINK, reading a text and returning each token annotated with a link to a lexical entry (or possibly multiple links). As a lexical base we use the newly published Central Word Register for Danish ("COR", cf. Henrichsen, 2023; Dideriksen et al., 2023; Widmann 2024), complemented with dictionaries for non-verbal lexemes and more. As we shall argue, CLINK'ed text provides enhanced input for language technology.

2. The Central Word Register

Retskrivningsordbogen ("RO", Jervelund et al., 2012) is the dictionary defining the Danish orthographic norm (Act 1997), available also as a machine readable database

This paper is part of the publication: Despot, K. Š., Ostroški Anić, A., & Brač, I. (Eds.). (2024). *Lexicography and Semantics. Proceedings of the XXI EURALEX International Congress.* Institute for the Croatian Language.

called COR$_1$ for language technology. Each lexeme in COR$_1$ carries a unique id specifying its class (PoS or otherwise), as exemplified in Table 1. The database is available at https://ordregister.dk.

Table 1: COR$_1$ samples. Indices with 5 digits: lemma, 3 digits: PoS

Lexeme	**Lemma**	**COR$_1$-id**	**Inflexion**	**Gloss**
hus	HUS	COR.43962.120	neut.sg.indef	*house*
huset	HUS	COR.43962.121	neut.sg.def	*the-house*
huse	HUS	COR.43962.122	neut.pl.indef	*houses*
husene	HUS	COR.43962.123	neut.pl.def	*the-houses*
mus	MUS	COR.74798.110	com.sg.indef	*mouse*
mus	MUS	COR.74798.112	com.pl.indef	*mice*
cyber	CYBER	COR.02858.890	prefix	*cyber-*

Notice in Table 1 that “mus” is ambiguous in Danish between a singular and a plural reading; but the corresponding COR ids are not. In general, COR linking (i.e., annotating text tokens with COR ids) effectively disambiguates the homographs.

3. Morphological Analysis as Formal Deduction

CLINK’s formal framework is based on categorial grammar (CG). Historically, CG has mainly been used for syntactical analysis of the sentence and its constituents. CG logic is however quite general and just as fit for morphology, or indeed any complex analysis with a syntax, a semantics, and a compositional relation between the two (Wood, 1993, is a good CG primer for linguists; Morill, 2010, is more complete).[1]

3.1 The Lexical Template

Each morph must be available to CLINK in the form of a lexical template. Formally, the template is a 5-tuple (**text-in**, **text-out**, **cat**, **sem**, **phon**). Observe in Table 2 that, for most morphs imported from COR$_1$, the category (**cat**) is identical to the PoS index, and the semantic proxy (**sem**) to the lemma index; exceptions are function morphs (such as the prefix “cyber” and the grapheme “.”) not occurring as free forms. Categories containing a slash require an argument: $y\backslash x$ thus needs an argument y to its left (x/y to its right) in order to become x. For example, two adjacent categories 601 601\602 will reduce to 602.

[1] To the best of our knowledge, Lambek grammar has not been in use for commercial NLP since the early 1990es, and even then only in smallish and strictly rule based systems.

Table 2: CLINK templates. The pair **text-in**/**text-out** constitutes a difference list (as in Prolog or Haskell). C=any characters; P=punctuation marks; A=alphabetic characters; ^{0}=possibly empty. X,Y=category. a,b=semantic term; s01,s02,... = semantic proxy for morphs not in COR$_1$. /.../ = phonetic forms (computer-readable symbols, Wells, 1997); crd,ord,pfunc = phonetic functions (notice that ‘silent’ morphs, even ‘’, may affect their surroundings).

lexeme	text-in	text-out	cat	sem	phon
‘mus’	[mus|*C*^{0}]	*C*^{0}	110	43651	/mu:?s/
‘mus’	[mus|*P*^{0}]	*P*^{0}	112	43651	/mu:?s/
‘hus’	[hus|*C*^{0}]	*C*^{0}	120	43962	/hu:?s/
‘cyber’	[cyber|*A*]	*A*	*X*/*X*	λa.a(02858)	/sAJbC/
‘e’	[e|*A*]	*A*	*X*\(*Y*/*Y*)	λab.b(a)	pfunc(GLUEe)
‘’	*A*	*A*	*X*\(*Y*/*Y*)	λab.b(a)	pfunc(GLUE0)
‘6’	[6|*P*^{0}]	*P*^{0}	601	s06	/sEgs/
‘.’	[.|*P*^{0}]	*P*^{0}	601\602	λa.ORD(a)	crd(a)>ord(a)
‘.’	[.|*P*^{0}]	*P*^{0}	*X*\(*X**41)	λa.[a,s41]	pfunc(fulstp)

3.2 Compound Analysis

Compound nouns like “cybermus”$_{SG}$ (*cyber mouse*), “husmus”$_{SG}$ (*house mouse*) and “musehus” (*house for mice*) are segmented into morphs as illustrated in Figure 1 (rows “SEG”).

SEG	[cyber]	[mus]		[cybermus]
SEQ	*X*/*X*	110	==>	110
SEM	λa.a(02858)	43651		43651(02858)

SEG	[hus]	[]	[mus]		[husmus]
SEQ	120	*Q*\(*Y*/*Y*)	110	==>	110
SEM	43962	λab.b(a)	43651		43651(43962)

SEG	[mus]	[e]	[hus]		[musehus]
SEQ	110	*Z*\(*W*/*W*)	120	==>	120
SEM	43651	λab.b(a)	43962		43962(43651)

Fig. 1: Compound analysis: segmentation, sequent, semantic form

Each segmentation is mapped to a sequent (rows “SEQ”) in the form ANTE==>CON, where ANTE (antecedent) specifies the list of categories, and CON (consequent) is the hypothesis to be proven. In case CON can be deduced from ANTE in formally valid steps,[2] the sequent has found its proof – and the

[2] The Lambek calculus has these seven proof rules:
(/**L**) *L0 A/B M1 N0* ==> *C* IF *M1* ==> *B* AND *L0 A N0* ==> *C*;
(\L) *L0 M1 B\A N0* ==> *C* IF *M1* ==> *B* AND *L0 A N0* ==> *C*;
(***L**) *L0 A*B M0* ==> *C* IF *L0 A B M0* ==> *C*;
(/**R**) *L1* ==> *A/B* IF *L1 B* ==> *A*; (**R**) *L1* ==> *B\A* IF *B L1* ==> *A*;
(***R**) *L1 M1* ==> *A*B* IF *L1* ==> *A* AND *M1* ==> *B*; (**axiom**) *A* ==> *A*;
A, *B*, *C* are categories, and *Ln*, *Mn*, *Nn* are lists of *n*+ categories.
Wood (1993) explains how to use the proof rules: For each rule *Seq* IF *Prem*, install a sequent for *Seq*, unify all variables, and you get one or more new premises *Prem*. Repeat for each premise until reaching **axiom**. You’re done.

compound word its category. In Figure 1, proofs are found for X=110, Q=120, Y=110, Z=110 and W=120. Observe that each proof is associated with a lambda-formula (rows "SEM") specifying the semantic relations between the morphs (compare "husmus" and "musehus").

3.3 Lexical and Structural Ambiguity

In general, homographs have distinct semantic projections.[3] Danish "lyst" can, for example, be a verbal participle (*shined*), a noun (*desire*) or an adjective (*bright*), all three pronounced differently. Approximately 14.5% of COR_1's word forms are homographs, many of these highly frequent. Among the ten most frequent tokens in typical text,[4] eight are homographs in COR_1 ("i", "en", "til", "af", "på", "at", "det", "for"), token "for" with no less than seven entries.

Danish is, like German, a compounding language. Words like "husmusehus" are readily created, in this case ambiguous between the meaning *house for domestic mice* or *mouse house for in-house use*. The reader may wish to confirm, CLINK-style, the two projections 43962(43651(43962)) and (43962(43651))(43962).

The Danish rules of interpunction are another source of ambiguity. The grapheme "." is lexically ambiguous between a full stop with sentential scope (as in "Klokken er 6.", *It's 6 o'clock.*) and an ordinal suffix with local scope (as in "på 6. sal", *on the 6th floor*). In the latter case, the morphs "6" and "." do combine morphologically; but not in the former. The projections are [s06,s41] and ORD(s06), respectively, the former simply listing the arguments, the latter composing them.

4. Computers Prefer Lambdas

Computers like COR-ids and λ-forms far better than text tokens. This is a challenge for any NLP project. Even state-of-the-art MT systems (machine translation) and chatbots often choke on lexical and structural ambiguities. Consider e.g., the Danish sentence "husmus bor i hus", translating naturally into *house mice live in houses* or even *domestic mice are living indoor.* Google Translate, in contrast, suggests "house mouse lives in house" – neither accurate nor pretty. In the same vein, Danish synthetic voices are notorious for mispronouncing homographs like "for", "sig", "lyst", "så". Last but not least, Danish spell and grammar checkers (MS-Word, Google Docs, LibreOffice) score poorly on compound recognition and punctuation. We believe that preprocessing the input text CLINK-style could solve most of these issues. Of course, this requires a CLINK parser comparable in performance to the state-of-the art for Danish language taggers. However, CLINK being an open standard, we can hope for a wider scope of contributions. More on this below.

[3] Notice that "mus"_{SG} and "mus"_{PL} both project to 43651 as they belong to the same lemma and hence share the same proxy. In fig.1, "mus"_{112} would also produce a proof (for Z=112), however with the same semantic projection as for "mus"_{110}

[4] All texts in DaGW (Danish Gigaword Corpus, see Derczynski et al. 2021).

4.1 Text Annotation

CLINK (standard configuration) annotates each input token for **cat** (proven consequent) and **sem** (λ-form). Consider an example from the classical Danish prose poem Ved Frokosten (fig. 2).

Gud	[gud]	900:0	14223:0
ske	[ske]	900:1	14223:1
Lov	[lov]	900:2	14223:2
for	[for]	880	00093
Sofahjørnets	[sofa][][hjørnets]	125	41222(77215)
Fløjl!	[fløjl][!]	110*43	[40705,s43]

Fig. 2: Excerpt from "Ved Frokosten" by Johs.V. Jensen (1905), "Gud ske Lov for Sofahjørnets Fløjl!" (literal translation: *God be praised for the-sofa-corner's velvet!*). Notice that "gud ske lov" is a lexicalized multiword expression, alternatively spelled "gudskelov"

Efficient Lambek provers are readily available. In addition, CLINK has a range of strategies for disambiguation based on morphological congruence, syntactical coherence, context cues and statistics (frequency tables, tensor models). See Henrichsen (2023) for discussion. CLINK's selection algorithm is highly parameterized, allowing user-defined combinations of rule based analysis and machine learning. Of course, many parts of CLINK are not specific to the current project.

We present some early performance metrics (cf. Bick 2023); however, our intended contribution lies mainly with the generalized lexicological scope (the morph) and its associated representation (the template).[5]

We are currently developing COR.TALE, a COR$_1$ compatible database with phonetic information suitable for TTS. COR.TALE (due in 2025) will allow phonetic output along with the **cat** and **sem** annotations (with very little runtime/footprint penalty thanks to the CG-style compositionality).

5. Concluding Remarks

COR$_1$ and CLINK's primary target group is the NLP developers. As many of these are neither trained linguists nor L1 speakers of Danish, it was imperative for the CLINK project to employ a lexicological approach with as little language specific grammar as possible. CG's minimalistic rule base thus came in handy. As a corollary, the CG base also relieves the morphologist of the dilemma between a nomenclature derived entirely from the target language and one imported from a culturally dominant foreign language. Maybe the CLINK template, for these reasons, could find its use in foreign languages too. We are currently preparing a workshop for the smallest languages in

[5] Measuring CLINK's performance reliably is not trivial as we currently have no manually verified test corpora with CLINK annotation for **cat**, **sem** and **phon**. We are currently working on such a gold corpus, aiming at 10 mio. manually verified CLINK-annotated tokens extracted from DaGW (Derczynski, 2021). Preliminary tests suggest that CLINK's performance is comparable to the Danish state-of-the-art (Asmussen, 2014; Bick, 2023), however the output of course being richer (including **sem** and **phon**).

the Nordic region (e.g., Greenlandic, Faroese, Samic languages) exploring CLINK as a basis for NLP; contact the author for more info.

In the Danish NLP sector (natural language processing), text preprocessors of all sorts (tokenizers, PoS-taggers, lemmatizers, classifiers) are being developed independently for TTS (text-to-speech), MT, word processors, chatbots and so on (Kirschmeier, 2019). We suggest a joint venture and invite all computational linguists to contribute to ordregister.dk with fresh COR-dictionaries (e.g., for specialized areas of expertise), parser modules, COR_1 related improvements and discussion. Above all, CLINK is intended as a framework for cross-organizational co-operations (such as Nimb, 2022; Bick, 2023) and friendly competition.

References

Asmussen, J. (2014). *Design of the ePOS tagger.* Technical report, DK-CLARIN. korpus.dsl.dk/clarin/corpus-doc/pos-design.pdf

Bick, E. (2023). Linking Danish Parser Output to a Central Word Repository. In: *Proceedings of KONVENS-19.* ACL (pp. 93–101).

Derczynski, L. et al. (2021). The Danish gigaword corpus. In: S. Dobnik et al. (Eds.), *NEALT Proceedings Series No. 45* (pp. 413–421).

Dideriksen, C. et al. (2022). Det Centrale Ordregister. *Nyt fra Sprognævnet 2022* (online only, https://dsn.dk/nyt-fra-sprognaevnet). ISSN 2446-3124.

Haspelmath, M. (2020). The morph as a minimal linguistic form. *Morphology*, *30*, 117–134. https://doi.org/10.1007/s11525-020-09355-5

Henrichsen, P. J. (2024). Tekstordet som grammatisk domæne. In J. Heegård (Ed.), *Ny forskning i grammatik* 31 (pp. 53–70).

Jensen, J. V. (1906). Digte 1906. Gyldendalske Boghandel.

Henrichsen, P. J. (2023). Det Centrale Ordregister. In: L. Holmer et al. (Eds.), *Nordiska studier i lexikografi 16* (pp. 113–126).

Jervelund, A. Å. et al. (2012). *Retskrivningsordbogen.* 4th edition. Danish Language Council Press (legal authority: cf. *Retsinformation.* ACT 332 14/05/1997, https://retsinformation.dk).

Kirschmeier, S. et al. (2019). *Dansk sprogteknologi i verdensklasse.* Report commisioned by Ministry of Culture. ISBN 978-87-89410-77-7.

Morill, G. V. (2010). *Categorial Grammar: Logical Syntax, Semantics, and Processing.* Oxford Linguistics.

Nimb, S. et al. (2022). COR.SEM – den semantiske del af Det Centrale OrdRegister (COR). *Lexico Nordic,a 29*, 73–95.

Wells, J. C. (1997). SAMPA computer readable phonetic alphabet. In: D. Gibbon et al. (Eds.), *Handbook of Standards and Resources for Spoken Language Systems.* Mouton de Gruyter. Part IV, section B.

Widmann, T. (2024). The Central Word Register of the Danish Language. In M. Medveď, et al. (Eds.), *Proceedings of eLex 2023* (pp. 91–103).

Wood, Mary McGee (1993). *Categorial Grammars.* Routledge.

Contact information

Peter Juel Henrichsen
Danish Language Council
pjh@dsn.dk