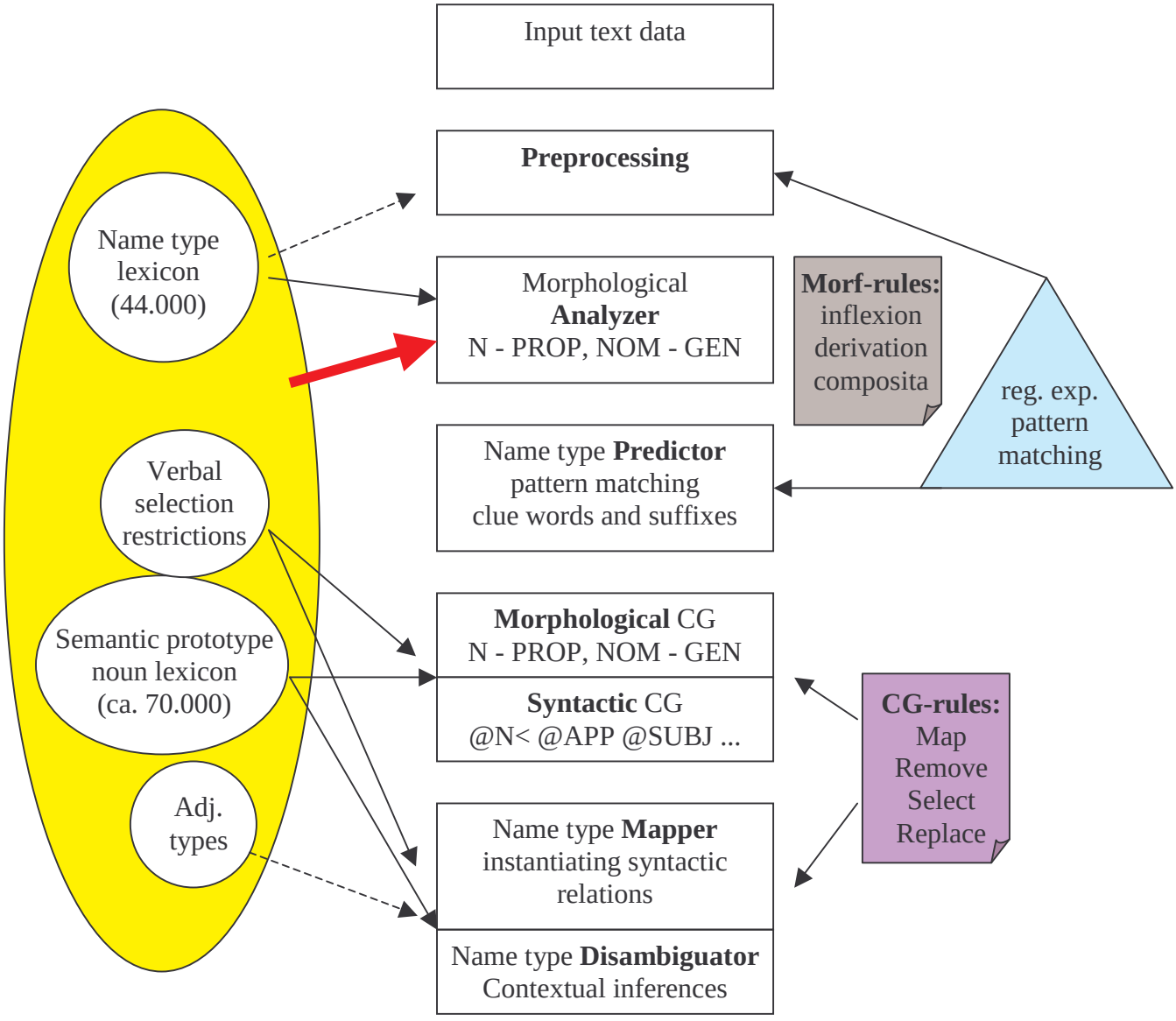
# Multi-level NER in a CG framework

Eckhard Bick

Institute of Language and Communication, Southern Denmark University

lineb@hum.au.dk, http://visl.sdu.dk

This is a short description of the DanGram parsing system from a name tagging perspective, with examples and text data specifics taken from the Korpus90/2000 annotation initiative. Since DanGram in its basic structure is based on progressive multi-level parsing, it was a natural step to delegate the difficult task of NER to different specialised levels, too, thus treating named entities successively as first strings, words, types, and then as contextual units at the morphological, syntactic and semantic levels, consecutively. While lower levels mainly use patter matching tools, the higher levels make increasing use of context based Constraint Grammar rules on the one hand, and lexical information, both morphological and semantic, on the other hand:



## 1. NE string recognition at raw text level, pattern based

dancorp.avis, dancorp.pre, dan.pre

The very first stage of name recognition in the Korpus90/2000 data is concerned with un-separated and highlighted source names, author names, and headings, a problem also found in copy-pasted e-mail or web text:

> **København N.** Førerbevis til ældre
> **LUXEMBORG** I over 700 år har ...
> **RADIO & TV** Københavns Sommerunderholdning
> Bedste da. resultat er opnået af Kurt Nielsen, der kom i finalen i 1953 og 1955. **KuNi INFORMATION**. Det er ikke sandt ...
> Kontokort **Af BENNY SELANDER** Stadig flere ...
> **HELGE ADAM MØLLER** Medlem af folketinget

The next task concerns the distinction between names and sentence initial capitalisation, where sentences are not separated by punctuation, a problem frequent in scanned or electronically acquired text with loss of line spacing and other typographical "separation information".

> Glimrende rolle **Det** er en ...
> Forholdet til Walesa "**Mit** forhold til Walesa er, som det var for ti år siden," siger ...

In texts where upper case is used for highlighting, highlighted words have to be distinguished form all-uppercase names, not least organisation names (NATO, UNPROFOR, USA) brand names (VHS), and chemicals (DNA):

> **MENS** børnene venter, **JOURNALIST** Michael Larsen

For this task, the preprocessing programs use both lists of non-name short words and heuristics (5 or more uppercase letters indicate non-names, unless used first or last, in "author position". After triage, highlight candidate words are then lowercased for later morphological analysis.

**Dancorp.pre** is the main text correction module. It repairs misread letters like ä, ü, ö, å, á, í etc. which often are altered in a specific corpus section in specific, sometimes context dependent ways, creating fals spaces and punctuation breaks in the process. In Danish text, most of these letters occur predominantly in names. Finally, spacing and hyphenation is corrected (US A, EF-landende, Central-og Østeuropa, 40 års-dagen)

**Dan.pre** handles NE string recognition proper. Basically, this module uses regular expressions to identify polylexicals, creating "words" by substituting in-word blanks with equal signs. It does this in a general task context of linebreak-separating words and other graphical units (punctuation, numbers, brackets) for later morphological processing. Here, one guiding NER principle is, of course, to fuse consecutive uppercase-initial strings into name units:

**person names:** Nyrup=Rasmussen
**institutions:** Odder Lille Friskole
**organisations:** ABN-AMRO=Asia=Equities
**events:** Australian Open

For personal names, the pattern recognizer checks for a list of international in-name prepositions, articles and co-ordinators: *Maria dos Santos, Paul la Cour, Peter the Great, Margarete den Anden, Ras al Kafji, Osama bin Laden*. Most of these particles occur, of course, also name initally: *da*

*Vinci, van Gogh, de la Vega, ten Haaf, von Weizsäcker*, where lower case makes recognition more difficult.

Similarly, certain international chaining particles are accepted in other name types:

- **organisation names:** Dansk Selskab for Akupunktur, University of Michigan, Golf-Centeret for Strategiske Studier, Organisationen af Olieeksporterende Lande
- **place names:** Place de la Concorde
- **brands:** Muscat de Beaumes de Venise
- **media:** le Nouvel Observateur
- **events:** Slaget på Reden

Once, part of a name string is recognised, it is allowed to grow to the right. Thus, 'Den/Det' in mid-sentence is a safe name-initialiser, even with an intervening lower case word: *Det ny Lademann Aps.*

In the name chaining process, non-alphabetic characters make for complications. Thus, normally, punctuation marks would be word separators. However, in Christian name initials, web-urls, e-mail addresses, abbreviations or book titles, and sometimes, creative brand names, punctuation marks can be part of the name string[1]:

- initials: P.=Rostrup=Bøjesen, Carl=Th.=Pilipsen
- web-addresses: http://www.corp.hum.sdu.dk
- e-mails: lineb@hum.au.dk
- personal name additions: Mr.=Bush, frk.=Nielsen, Bush=jr./sr., hr.=Jensen, ...
- professional titles: Dr.=A.=Clarke, cand.=polit., mag.=art.
- company names: Aps., D/S Isbjørn
- vehicles: H.M.S. Polaris
- geographicals: Nr.=Nissum, Kbh.=K, St.=Bernhard. Mt.=Everest

Both Arabic and Roman numerals can be part of name strings, and in the case of version numbers and car names, the full character spektrum can be mixed in one name:

- yearly events: Landsstævne='98
- card names: Ru=9, Sp=E
- license plates: TF=34=322
- town adresses: 8260=Viby=J
- house addresses: 14a,=st.=tv.
- kings: Christian IV
- dated or versionized products: Windows 98
- vehicle names: Honda Civic 1,4 GL sedan, Citroën ZX Aura 1.6i, Peugeot 206, 1.6 LX van, 1,9 TDI, DC10 fly, købe 50 V44 vindmøller
- news channels: TV2, DR 1, Channel 4
- bible quotes: Mt 28,1-10

[1] Word-internal punctuation is not, of course, specific to names - the dan.pre module also recognizes e.g. complex numbers strings, prices (kr. 45,-), and abbreviations. In the case of the latter, a necessary distinction between sentence final and sentence internal abbreviation is made, and if necessary, a full stop is added after the abbreviation dot.

A special problem is whether &-signs and slashes should be handled as word internal characters or as separate coordinator units ("words"):

K/S Storkemarken
Munster & Co., Møller & Baruah (fused company names?)
Hartree & V. Booth: Safety in Biological Laboratories (separate authors?)
NATO/FN
1560 kJ/420 kcal

Book and film titles are sometimes, but not always, enclosed in quotes. If so, punctuation marks may occur within the NE, and there is a potential confusion with ordinary quoted speech. If not, it may be difficult to handle words with lower case words within the NE:

Med den melankolske 'Light Years' anslås ...
Filmen "Stay Cool" blev trukket ...
Han havde læst Den uendelige historie.

In the case of single quotes, the distinction from genitive-apostrophes and ellision apostrophes has to be made.

- **genitive:** 'The Artist's Album', Bush' korte besked. 'Big Momma's House'
- **company names:** Kellogg's
- **ellision:** Pic du Midi d'Osseau, Côte d'Azur, Montfort l'Amaury
- **fixed naming systems:** O'Connor, O'Neill

**2. NE string recognition at raw text level, lexicon supported**

dan.pre

Not all NE chain fusion can be done with pattern matching alone, and sometimes there will be chunking ambiguities that can only be resolved by consulting a lexicon. Dan.pre therefore has at its disposal lexical word lists defining certain name contexts, and the name part of the parser's core lexicon, including polylexical information and semantic name types.

For instance, in the above mentioned pattern based name chaining, some lower case particles should only be accepted in certain left or right contexts, like 'for' in conjunction with 'selskab', 'organisation' etc. Here the recognizer needs lexical knowledge at the simplest level, in the form of an "allowed word list", optionally with an inflectional addition like (en|et|ern?e?)?, i.e. something in the vein of the famous "stop lists" of statistically based taggers. Another example is the conjunction 'og', which can be said to be pattern-handled in *Petersen og Co.*, but needs real lexical lists to recognize *Told og Skat, Sol og Strand, Se og Hør*. Integration of numerical expressions into names can be helped by word lists, too. Thus, a list of car company names helps NOT integrating numbers into other names:

Peter købte en Peugeot=206.
* Den gang købte Peter=206 pakkegaver.

Numbers can be kept off all-uppercase organisation and brand candidates, if a list of unit/measuring words with <num+> valency is given, like *procent, pct., mill., mio., kr.* etc.:

... tjente **Toyota 6,8** procent
tallet for Europa er 60 procent og for **USA 75** procent

Extensive word list are also used where sentence initial capitalisation has to be distinguished from names. These list contain typical sentence initial small words *(fordi, derfor, siden, ifølge, har),* which are partly used to correct faulty sentence separation, partly to split name chain candidates where the second word in a sentence is a (upper case) name *(Fordi=Peter=Jensen ikke havde sendt ...).* One of these lists contains nouns, prefixes and suffixes with <+name> valency, i.e. professions, family members, group words and the like: *adjunkt, ...chef, historiker, institut, kollega, ...erske, ...trice, virksomhed, ...ør, Hoved..., Vice....* In the examples below, # marks the name chain splitting point:

lektor i børsret ved Københavns Universitet # Per Schaumburg Müller
Landsstyreformand # Jonathan Motzfeldt
Styresystemet # Windows
Stillehavsøen # Okinawa
Vicelagmand # Oli Nilsson

In the case of fairly safe morphological endings (genitive 'erne' and verbal 'ede') pattern matching was originally used with an inverse list of *forbidden* split words *(Horsens, Jens, Vincens, Enschede* etc.). Later, all splitting rules were subjected to a lexical look-up of the entity to be split, preventing splitting for lexicalized items. In the case of genitive candidates, split halves are lexicon checked individually, and splitting is allowed if the first part is a known person, organisation, institution og political unit (i.e. humanoids). Furthermore, the second half can be checked for name type, too, allowing only for certain defined combinations, or refusing, for instance, genitive geographicals like *Jensens Plads, Rådmans Boulevard*:

Sonofons <org> # GSM 900-net
Richard Strauss' <hum> # Zarathustra
New Yorks <civ> # Manhattan
Beach Boys' <org> # Brian Wilson <hum>

Without a genitive marker, things are more difficult, but nevertheless, with a lexicon based chunk splitter, name - name sequences of the following types can be detected:

Kommende ambassadør i Kairo # Christian Oldenburg
Bagefter hentede Peter # Maria
Så ansatte IBM # Kevin Mondale
Derfor forlod Jensen (Peter Jensen) # (FC) København

As the variability of the last example indicates, the chunker has to do more than one lexicon lookup each time, covering, for instance, in a 4-part name chain the 1+3, 2+2 and 3+1 combinations.

## 3. NE word recognition, morphological analyzer with lexical data and compositional rules

dantag, dan.post

The morphological analyzer, *dantag,* uses lexicon name data in two ways:

(a) full lexicon entries (especially major place names and Christian names, but also a number of common surnames and company names, ca. 45.000 entries), semantically subclassified into 20 classes, falling into 6 major categories (jf. Fefor 2002)
(b) partial lexicon entries, used by the lexical analyser prior to disambiguation, in order to heuristically assess unknown multi-word names with a known first, second or third part (e.g.

known Christian names with unknown surnames, or known company names with geographical extensions: *Toshiba Denmark*).

This lexicon module introduces both cross word class ambiguity, e.g. *Otte, Hans* in sentence initial position, and name class internal ambiguity, e.g. *Lund* (place/person), *Audi* (company/vehicle) etc. The latter is not necessarily lexicon-registered, if it is systematic, like in the use of town names for sports teams. The category <media> expresses a systematic ambiguity title/organisation *(Jyllandsposten, TV2).*

Upper case words with no entry in the name lexicon will first be checked as lower case against the ordinary lexicon, using full inflexional and derivational analysis. Second, a compositional reading is tried, cutting the word in parts and matching the first against the name lexicon and the rest against the ordinary and suffix lexicon. The resulting analysis draws its word class tag from the word class of the final part of the composition:

**ANC**-kontor N, **G8**-mødet
**Taleban**styrel N, **Martini**cocktails N, **Mars**året N
**AB**'ernes [AB'er] N, **AGF**'ere N
**EU**-godkendt ADJ, **Heisenberg**'ske ADJ

Similarly, names may be treated as nouns in inflected disguises:

**EMS**en (N DEF), **EMS**'en (N DEF)

However, singular definite forms will be treated as names if the lexicon says so, a sensible option for "frozen usage", as in *Sovjetunionen* og *Folketinget.*

While hyphens between uppercase and lowercase words indicate name-derived nouns, hyphens between uppercase words often indicate human names, - though not always:

Jarl Friis-Mikkelsen, Jean-Pierre Wallez, Blomster-Jensen
hovedvej Pec-Prizren <top>, Lolland-Falster <top>
Al-Qaida <org>, CO-Industri <org>, Jyllands-Posten <media>

As part of its heuristic iterations, the morphological analyzer will retry not-identified words substituting hyphens for equal signs and vice versa, i.e. 'Al Qaida' (Al=Qaida) will be found also if entered only as Al-Qaida, and the other way around.

Finally, if all else fails, or if another word class reading is "unsafe" (compounds, sentence initial), the *dan.post* programme will assign a heuristic name reading to upper case words, though in most cases without a semantic subtype guess.

**4. NE semantic type prediction, semi-lexical compositional heuristics**

cg2adapt.dansk

This is where the semantic type predictor goes to work. The program respects safe, i.e. fully lexicon based subtype readings from the morphological analyser and tries to confirm lexicon based half-guesses (e.g. known name type as first part of a name chain). In all other cases, the type predictor tries to instantiate morphological (derivational) and otherwise pattern based type clues for the different categories:

**<tit>** e.g. quotes, in-name function words (articles, pronouns etc.), "semantic things" *(-loven, -brev, -song, -report, Circulære=, Redegørelse=, Dictionary= ...)*

**<media>** e.g. *-avis, -bladet, -tidende, Ugeskrift=, Kanal=, Channal=, Nyt= ...*
**<occ>** e.g. *Expedition, -freden, -krig, -krise, =Rundt, Projekt=, Konference=, Slaget=*
**<V>** e.g. *Boeing/Mercedes/Toyota=, =Combi, =Sedan, HMS=, USS=, M/S= ...*
**<brand>** e.g. *Macintosh/Phillips/Sanyo=[0-9]*, wine types:*=Appelation, =Cru, =Sec, Edition, Yamaha/Siemens=*, quality markers:*=Extra, =de=Luxe, =Ultra ...*
**<hum>** e.g. suffixes:*-sen, -sson, -sky, -owa*, infixes: *ibn, van, ter, y, zu, di*, abbreviated and part-of-name titles: *frk., hr., Madame, Mlle, Morbror, jr., sr., Mc=, Al=, =Khan*
**<A><B>** e.g. [A.Z][a-z]+(=[a-z]+([ae]ns|ea|is|um|us))+
**<civ>** e.g. *=SSR/Republik, =Town/Ville*, suffixes: *-ager, -borough, -bølle, -dorf, -hausen, -løse, -ville, -polis* (a number of these will receive both <civ> and <hum> tags for later disambiguation
**<top>** e.g. *=Bahnhof, =Bakker, =Kirke, =Manor, =Sund, =Prospekt, Islas=, Ciudad=, Gammel=, Lake=, Rio=, Sønder/Vester/Øster/Nørre=*, suffixes: *-fors, -kanten, -kvarteret, -marken*, addresses: *-stien, -strasse, -torv, -gade, -vej* (the latter are also used by dantag)
**<org>** e.g. in-word capitals: *[a-z][A-Z] (MediaSoft)*, "suffixes": *Amba, GmbH, A/S, AG, Bros., & Co ...*, type indicators: *=Holding, =Organisation, =Society, =Network, Bank=of=, Banco=d[eiao], K/S, I/S, Klub=, Fonden=*, morphological indicators: *-con, -com, -ex, -rama, -tech, -soft*
**<inst>** e.g. *=Ambassade, =Airport?, =Børnehave, =Institut, =Universitet, =Bibliotek, =Hotel, Chez=*, morphologicals: *-eriet, -værk, -handel*
**<mat>** e.g. *-[cpt]am, -[cz]id, -lax, -vent, Retard=*, uppercase + number *(NO2, H2O)*
**<common>** e.g. *=Collection, =Samling, Ugens=*, cards: *Spa?=, Ru=*

One problem are interferences between the differen sections of the type predictor. For instance, place names can be part of club names, and human names can be part of prize titles. This is why ordering the sections is important, or even running parts of a certain type predictor disjunctly or iteratively. Also, some of the pattern-lists have NOT-conditions quoting partial or overlapping patterns that would indicate other semantic name classes.

Finally, the type predictor assigns <non-hum> tags, preventing over-usage of this most common category in the cg-based part of the system. The <non-hum> category judges from e.g. non-alphabetic characters, in-word capitals, coordinators *(og, eller)*, certain English function words *(of)*, non-human suffixes *(-tion)* etc.

## 5. NE word class and case disambiguation, rule and context based

dancg.morf

This module is a full-fledged cg-grammar with sentence-wide context rules. It consists of 3.300 rules and has access to word class, inflexion, verbal and nominal valency potential, semantic noun and adjective class etc. The idea is to disambiguate the morphological readings suggested by the analyzer at an earlier stage. Names are only a minor part of this task. Nevertheless it is much safer to contextually disambiguate, say sentence initial imperatives from heuristic proper nouns, than at the pattern matching stages. Also, some names have to be morphologically disambiguated as to nominative (NOM) and genitive (GEN). Some simple principles used by the grammar are:

- <+name> valency of preceding noun: *filmen **Tornfuglene** PROP <tit>*
- semantic product class <sem> in preceding noun: *Lynda La Plantes berømmede tv-serie "Mistænkt"*
- topologicals rather than topological-derived nouns: *Amagerbrogade PROP <top>*
- establishment NOM rather than human GEN, if no np-head to the right: *vi spiste på **Marion's** i går*

- GEN - GEN and NOM - NOM matching in name coordination: ***Peters*** *NOM og Jensen kom kørende*
- NOM name readings are discarded in favor of GEN names, if there is an indefinite noun or NOM name to the right with only matching prenominals in between: ***Australiens*** *mest kendte sangere.*
- Sentence-initially, names are discarded in favor of verbs and function words, if followed by an np
- non-compound nouns are favoured over heuristic names
- heuristic names are favoured over compound names in a left lower case context
- non-heuristic names are favoured over compound or derived nouns sentence-initially or in left upper case context

Of course, proper nouns can also for their part form valuable context for the disambiguation of other classes, for instance making modifier word classes and articles less likely to the left, while making nouns with <+name> valency more likely. <hum> names, in particular, help choosing certain human class nouns the left (<Hprof>, <Hfam>).

**6. NE chaining, a repair mechanism for faulty NE string recognition at levels (1) and (2)**

cleanmorf.dan

This module in general instantiates word fusing and word separating choices too hard or too ambiguous for the preprocessors to make, and therefore left to the CG level for contextual disambiguation. For instance, a distinction is made one-token and two-token readings of *det her, I andre, alt=efter,* and adverbial quantifier use in *langt=fra, en=smule* or *mere=end* is distinguished from analytical uses.

In the case of name chains, this module fuses *Hans Jensen* og *Otte Nielsen,* but keeps *Hans Porsche* and *Otte PC'er* separate, drawing on the CG's word class and subtype disambiguation of *Jensen PROP <hum>, Nielsen PROP <hum>, Porsche PROP <V>* and *PC'er N <cc-h>*.

Another task is to fuse chains of PROP and certain semantic N types, if the noun is in upper case, and if fusion is not already recognised earlier, e.g.:

PROP + N <build> -> PROP <top>: Betty=Nansen Broen
PROP + N <HH> -> PROP <org>: Betty=Nansen Foreningen
PROP + N <sem> -> PROP <tit>: Betty=Nansen Prisen

An advantage of this method is that the semantic type assignment can draw on the full semantic noun lexicon, not only on the patterns and lists of the name type predictor (4).

Finally, the module repairs erroneous name splitting performed by dan.pre, for instance in the case of heuristic genitive first halves where the individual parts later are semantically typed in such a way as to make a one-token analysis likely, for instance:

PROP <org, media> + PROP <top, civ>: Dansk=Røde=Kors Afrika[2]
PROP <civ> + PROP <org, inst>: Danmarks Monetære Institut

**7. NE function classes, mapped and disambiguated by context based rules**

dancg.syn (ca. 4000 rules)

[2] If the '-s' in 'Kors' had triggered a genitive splitting in dan.pre, which is not the case any longer, due to lexical look-up in that preprocessor module.

Names can, of course, exercise most of the syntactic functions of ordinary np's (subject, object, argument of preposition). However, some functions are either more limited and specific in use or, on the contrary, more elaborate, in the case of proper nouns. Thus, name predicatives appear with a more limited set of verbs (@SC with *være, hedde*, @OC with *kalde, døbe*, not, e.g., *blive, gøre*), and cannot be free predicatives. On the other hand, especially in a news text corpus, there are name specific types of apposition and other nominal postmodifiers with a high frequency for names:

(i) @N< (nominal dependents)
*præsident **Bush**, filmen "**The Matrix**"*
(ii) @APP (identifying appositions)
*Forældrebestyrelsens forman, **Kurt Chistensen**, anklager borgmester ...*
(iii) @N<PRED (predicating appositions)
*John Andersen, distrikschef, **Billund**, 60 år*

All of these syntactic relations, once established, can at a later level be used by the system to derive semantic name types from lexical semantic information residing in the corresponding noun heads.

## 8. NE semantic types, mapped and disambiguated by context based rules

dancg.prop (428 rules)

This level works with the same 20 NE types used by the lexicon (3) and type-predictor (4), but has the decisive edge of being able to draw on syntactic relations and sentence context. Thus, rules can exploit lexical semantic information pertaining to *other* word classes, especially noun prototypes and verbal subcategorization, to a lesser degree adjective types. Like the syntactic grammar, *dancg.prop* consists of both a mapping CG and a disambiguation CG. The former is capable of adding to or even overriding semantic class types from the preceding levels, while the latter removes or selects semantic type tags where the lexion, heuristic type predictor or mapping grammar created ambiguities.

### 8.1 Cross-nominal prototype transfer

As a matter of principle, the named entity CG module uses semantic *noun* classes as a context, more or less transferring a head noun's prototype class to a proper noun dependent, where the latter syntactically attaches to the former. Consider, for instance, the following example rules covering the prototype <top> (place) and using the semantic noun type set N-TOP for contextual disambiguation:

MAP (<top>) TARGET (PROP @N<) (-1(N NOM) LINK 0 N-TOP) ;

This rule "instantiates" place-hood (<top>) for names that have been marked as postnominal dependents (@N<) by the syntactic grammar, if their head is itself a place-word (N-TOP), as in "i byen Rijnsburg". Note that this rule is simplified, as it usually has to compete with the other two name categories, that imply +LOC, i.e. <civ> (towns and countries) and <inst> (institutions).

MAP (<top>) TARGET (PROP) (1 @N<FUSE LINK 0 N-TOP) ;

This rule covers "Uppenskij katedralen", where the missing hyphen first leads to a 2-part analysis, which, however, fails to assign 'katedralen' a normal syntactic function, leaving @N<FUSE to survive all REMOVE-rules for all other functions. @N<FUSE is than used to (a) linke the two words, (b) to transfer "place-hood" to the name "Uppenskij".

SELECT (<top>) (0 @SUBJ>) (*1 @MV LINK 0 <vk> LINK *1 @<SC LINK 0 N-TOP) ;

This rule matches subject names with subject complements that are safe place nouns, as in "Moskva er en by i Rusland", and could be run in the reverse, as in "Den største by i Rusland er

Moskva". More complicated versions of this rule cover, for instance, "Moskva er en af de største byer i Rusland".

SELECT (<top>) (0 NOM) (*1 (<rel> INDP @SUBJ>) BARRIER NON-KOMMA LINK *1 VFIN LINK 0 @FS-N< LINK -1 ALL LINK *1 @MV LINK 0 <vk> LINK *1 @<SC LINK 0 N-TOP);

This rule, one of the most complex that will be given here, checks for place subject complements (@SC N-TOP) in relative clauses (@FS-N<) with a relative pronoun (<rel> INDP) at most one komma away to the right of the named entity in question. The following rule checks relative transobjective relations in a similar way:

MAP (%top) TARGET (PROP NOM) (*1 ("som") BARRIER NON-KOMMA LINK 0 @OC> LINK *1 @MV LINK 0 ("kalde" AKT) LINK *1 N-TOP BARRIER NON-PRE-N/ADV LINK 0 @<ACC) ; # Strongyle, som de gamle grækere kaldte øen

MAP (%tit) TARGET (PROP NOM) (0 @P< OR @AS<) (-1 ("som") LINK 0 @N< OR @AS-N<) (-2 N-SEM) (NOT -2 N-HUM) ;

This rule draws on nominal type context through ”som” comparison links, and can override other previous name tags: *tv-programmer som "Robinson-Ekspeditionen"* (ellers <occ>, men her <tit>)

### 8.2 Coordination based type inference

Drawing on and matching syntactic tags from the syntactic CG-module, the name type-mapper first establishes a secondary tag for "close coordinators" (&KC-CLOSE), with one rule for each matched syntactic function, e.g.:

ADD (&KC-CLOSE) TARGET (KC) (*1 @SUBJ> BARRIER @NON->N) (-1 @SUBJ>)

Most of these rules are simple, but when coordinating (flat) prepositional dependents (@P<) one has to take into account whether or not the left context pp in question is itself a postnominal of a @P< nominal, in which case there arises ambiguity as to which of the two left @P< is the true conjunct of the the right context @P<.

Once established, close coordinators can be used in disambiguation rules:

REMOVE %non-h (0 %hum-all) (*-1 &KC-CLOSE BARRIER @NON->N LINK -1C %hum OR N-HUM LINK 0 NOM); # e.g. Arafat @SUBJ> og hans Palæstinas=Selvstyre @SUBJ>

SELECT (<top>) (1 &KC-CLOSE) (*2C <top> BARRIER @NON->N) ;

The following coordination based rules exploit that Danish has 2 sets of third person personal pronouns, one for humans, another for non-humans. The rules shown are function specific, and thus part of a whole set of rules for individual syntactic functions, a complexity that could be reduced to a single pair of rules by using the &KC-CLOSE tag:

```
SELECT %hum (0 @SUBJ>) (1 KC) (2 ("han" GEN) OR ("hun" GEN)) (*3 @SUBJ> BARRIER
@NON->N/KOMMA) ; # Hejberg og hans skole

REMOVE %hum (0 @SUBJ>) (1 KC) (2 ("den" GEN) OR ("det" GEN)) (*3 @SUBJ> BARRIER
@NON->N/KOMMA) ; # Anden Verdenskrig og dens mange slag
```

### 8.3 PP-contexts

```
MAP (<top>) TARGET (PROP) (-1 ("for" PRP)) (-2 ("syd") OR ("vest") OR ("nord")
OR ("øst")) ;
```

This rule is an example for a place specific narrow context, looking for "syd for Odense" kind of patterns. Similar rules exist for "indgang til", "vejen til" and for the preposition "nær" in immediate left context.

```
ADD (<top>) TARGET (PROP @P<) (-1 ("i" PRP)) (NOT -1 @PIV) (NOT -2 <+i>) ;
```

This rule derives place-hood for an argument of a preposition, if that preposition is 'i' and the pp is NOT an object (@PIV) as in "deltage i" and if the there is no valency demanding nominal context left of the preposition (<+i>), as in "forelsket i". Note, that the rules run in layers, and that a later heuristic rule will not only not map <top> afte <+i> contexts, but actually *remove* "older" <top> readings in that context: `REMOVE (<top>) (0 @P<) (-1 ("i" PRP)) (-2 (<+i>))`, as most nouns with <+i> valency prototypically ask for non-place arguments. This is, of course, an unsafe rule, as there are metaphorical and other exceptions ("forelsket i Venedig"), but if other, earlier and safer, rules have already disambiguated the place/human ambiguity, no harm will be done even in the exception cases.

```
REMOVE %non-top (-1 ("fra" PRP) OR ("til" PRP)) (-2 N-DIST) (-3 NUM) ; # godt 40
km fra Madras

MAP (%org) TARGET (PROP NOM @P<) (-1 ("i" PRP)) (-2 ("afdelingsleder") OR
("ansat") OR ("chef") OR ("direktør") OR ("forvaltningschef") OR ("koordinator")
OR ("personalechef") OR ("souschef")) (NOT 0 <top> OR <civ>) ;
```

### 8.4 Genitive mapping

```
MAP (%org) TARGET (GEN @>N) (*1 (N IDF) BARRIER @NON->N LINK 0 GEN-ORG) (NOT 0
<inst> OR <media> OR <party> OR <civ> OR <top>) ;
# Microsofts generalforsamling/aktiekurs

MAP (%org) TARGET (GEN @>N) (*1 (N IDF) BARRIER @NON->N LINK 0 GEN-ORG/HUM) (NOT
0 <inst> OR <media> OR <party> OR <civ> OR <top> OR <hum>) ;
# Microsofts/ Bill=Gates advokat/hjemmeside

REMOVE %non-h (0 GEN LINK 0 %h) (*1 N BARRIER @NON->N/KOMMA LINK 0 (<p>) OR
(<pp>)) ;
This rule removes non-human readings on genitive names, if they ”own” thoughts
(<p>) or thought products (<pp>). Note that the %non-h set is distinct from
%non-hum in that it is a more limited set, respecting <org>, <party> etc. as
human-type. Thus, the rule would not apply to, say, companies ”owning” plans.
```

### 8.5 Prenominal context: Using adjective classes

Like nouns, adjectives have been semantic type classified in DanGram’s Lexicon. Though not extensively used yet, this information is used by a few rules in the name type cg, the most important drawing on the class of human adjectives. Two cg-sets are defined for this purpos, one type based, more general (ADJ-HUM), one word based and thus safer and more limited (ADJ-HUM&):

```
LIST ADJ-HUM = <Dphys> <Dpsych> <Dsoc> <Drel> ;

LIST ADJ-HUM& = <alder> "adfærdsvanskelig" "adspredt" "affektlabil" "afklaret"
"afmægtig" "afslappet" "afstumpet" "afvisende" "agtbar" "agtpågivende" "agtsom"
"alert" "alfaderlig" "alkærlig" "altopgivende" "altopofrende" "alvorsfuld" ....
```

The following rules exploit the ADJ-HUM classes for mapping <hum> tag onto hitherto untyped (<heur>) names:

```
MAP (%hum) TARGET (<heur> PROP NOM) (-1 AD LINK 0 ADJ-HUM&) (*-2 (ART S DEF)
BARRIER @NON->N) ; # Den langlemmede Kanako=Yonekura

ADD (%hum) TARGET (<heur> PROP NOM) (-1 AD LINK 0 ADJ-HUM) (*-2 (ART S DEF)
BARRIER @NON->N) ; # Den langlemmede Kanako=Yonekura
```

## Name type errors in running text

| **Korpus90** (jan. 2002) | chunk 1: lexicon-PROP | | chunk 2: heuristic PROP | |
|---|---|---|---|---|
| | instances (100.000 words) | percentage of all PROP readings | instances (100.000 words) | percentage of non-lexicon readings |
| wrong major class (6 classes) | 266 | 5.0 % | 151 | 9.2 % |
| wrong subclass, same major class | 31 | 0.6 % | 3 | 0.2 % |
| false positive PROP reading (incl. "overchunking") | 56 | 1.2 % | 27 | 1.6 % |
| false negative (missing) PROP (incl. "underchunking") | 22 | 0.4 % | 13 | 0.8 % |
| cross-class ambiguity (major classes) | 7 | 0.1 % | 0 | 0.0 % |
| all proper nouns | **5330** | | 4793 | |
| of these: not in lexicon | 1833 (34.4%) | | **1641** (34.2%) | |

| **Korpus2000** (jan. 2003) | all PROP (4.6% of words) 3091 PROP elements | | heuristic PROP | |
|---|---|---|---|---|
| | instances (ca. 43.000 words) | percentage of all PROP readings | instances (ca. 43.000 words) | percentage of non-lexicon readings |
| wrong major class (6 classes) | 99 | 5.0 % | 67 | 7.3% |
| wrong subclass, same major class | 16 | 0.8 % | 8 | 0.9 % |
| false positive PROP reading (incl. "overchunking") | 11+16=27 | 1.4 % | 6+14=20 | 2.2 % |
| false negative (missing) PROP (incl. "underchunking") | 6+10=16 | 0.8 % | 6+10=16 | 1.7 % |
| cross-class ambiguity (major classes) | 9 | 0.5 % | 1 | 0.1 % |
| all proper nouns | **1.996** | | 1.996 | |
| of these: not in lexicon | 922 | | **922** | |

In this evaluation, a 43.000 word running text chunk from the Danish Korpus2000 was automatically analysed and name type tagged. Errors were counted across categories and within categories. While overall error rates (5% cross-category and 0.8% category internal) roughly resemble what was found for the Korpus90 data in 2001, there are som important differences, too.

First, false positive/negative readings and chunking errors are higher for Korpus2000, suggesting differences in "corpus purity", for instance with regard to quotation marks[3]. Second, almost half of all names had to be heuristically analysed (as opposed to a third in Korpus90), reflecting the 10-year gap between the to sets of data, and the hight productivity of the proper noun lexicon. However, these heuristic tags had a somewhat lower incidence of cross-category errors. The weight of in-category errors shifted from lexical to heuristic, reflecting the more extensive use of heuristic name type mapping rules.

**Cross-class and class-internal name type errors**

| *tagged as:*<br>*should be:* | <**hum**><br><A><br><B> | <**org**><br><party><br><media> | <**top**><br><civ><br><inst> | <**occ**> | <**tit**>[4]<br><genre><br><ling> | <**brand**><br><V><br><common><br><mat><br><astro> | sum cross-cat. | sum all |
|---|---|---|---|---|---|---|---|---|
| <**hum**><A><B> | 4 | 2 | 7 | 1 | 1 | 2 | **13** | 17 |
| <**org**><party><media> | 17 | 1 | 5 | 1 | 1 | 2 | **26** | 27 |
| <**top**><civ><inst> | 19 | 7 | 11 | 0 | 0 | 2 | **28** | 39 |
| <**occ**> | 1 | 0 | 0 | 0 | 0 | 0 | **1** | 1 |
| <**tit**><genre><ling> | 14 | 4 | 1 | 1 | 0 | 1 | **21** | 21 |
| <**brand**><V><astro><br><common><mat> | 4 | 4 | 2 | 0 | 0 | 0 | **10** | 10 |
| sum cross-category | **55** | **17** | **15** | **3** | **3** | **7** | **99** | |
| sum all | 59 | 18 | 26 | 3 | 2 | 7 | | 115 |
| error bias = n(tag-error) / n(cat-error) | **4.2** | 0.7 | 0.5 | 3 | **0.14** | 0.7 | | |
| tag frequency | 925 | 358 | 607 | 22 | 61 | 30 | 2005 | |
| | 46.1 % | 17.9 % | 30.3 % | 1.1 % | 3.0 % | 1.5 % | | |
| error incidence = n(tag-error) / n(tag) | 5.9 % | 4.7 % | **2.5 %** | 13.6 % | 4.9 % | **23.3 %** | | |

The next table breaks down error frequency according to name type category. Obviously, the big categories, like <hum> (almost half), <top> (a third) and <org> (a sixth) account for correspondingly large share of errors. Expressed as relative error incidences, however, <top> stands out as particularly "safe" (2.5 %), while the rare categories of <occ> and <brand> impress as "unsafe" (13% and 23% errors, respectively).

Another interesting detail becomes visible when computing an "error bias", here defined as the ratio between erroneous usage of a given tag-category ("over-usage") and recall-failures for a given category ("under-usage"). For instance, person names <hum> are over-used by the system, usurping proper nouns that should have been read as something else, while titles <tit> are under-used, i.e. often tagged as something else (usually when unquoted).

Category-internal errors occur almost exclusively in the <hum> category, where the rare animal and plant names may be read as human personal names, and in the <top> category, where rare <civitas> names may be underspecified as ordinary place names[5].

[3] In fact, in 36 cases, probably due to scanning or excepting techniques, (partly) uppercase headlines, author lines and the like were fused to subsequent sentences in a way escaping automatic sentence boundary recognition. "Name" candidates resulting from such fusion were left out of this evaluation.

[4] The <tit> category also contains weather names <wea> and disease names <disease>. However, no instances were found of these categories in the text chunk examined.

[5] This error may in some cases even stem from the lexicon, since place names were partly heuristically compiled from "safe" contexts, and only later partly moved into the newly introduced <civitas> category.

The CG-system referred to in this text, and some relevant applications, are described in more detail in the publications listed below. For an on-line demo see http://corp.hum.sdu.dk/names.html, and the general VISL-pages http://visl.sdu.dk. Searchable corpus-samples for Danish can be accessed at http://corp.hum.sdu.dk. More information on the history and concept of DSL's Korpus90/2000 is available at http://www.dsl.dk.

Asmussen, Jørg, "Korpus 2000", in *Korpuslingvistik (NyS30),* Akademisk Forlag/Copenhaguen University, 2001

Bick, Eckhard, *The Parsing System 'Palavras' - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Århus: Aarhus Universitetsforlag, 2000

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

Bick, Eckhard "Morfosyntaktisk opmærkede corpora for Dansk: Korpus90/2000 og Arboretum", in *9. Møde om Udforskningen af Dansk Sprog, Proceedings,* Århus University, 2002

Bick, Eckhard, "Named Entity Recognition for Danish", in: *NORFA year book 2002*, (forthcoming)

Karlsson, Fred & Voutilainen, Atro & Heikkilä, Juka & Anttila, Arto (eds.), *Constraint Grammar, A Language-Independent System for Parsing Unrestricted Text*, Berlin: Mouton de Gruyter, 1995