

Danish-to-English Shake'n'Bake Machine Translation with a Constraint Grammar analysis

Søren Harder

July 12, 2002

Abstract

This paper is a short description of my PhD-project written as a presentation for the course “Almen og datamatisk leksikografi - med korpuslingvistiske metoder” in Rosendal, Norway, September 2002.

1 Purpose of the PhD

The purpose of the PhD is to produce a machine translation system from Danish to English, based on the existing Danish Constraint Grammar, which has been developed by the VISL-group at University of Southern Denmark. In developing the system more weight will be put on the linguistic coverage of the system than on the quality of the output. This is to say the translation system should be able to handle most or all of the input that the underlying Constraint Grammar-tagger can handle, but that the quality of the English translation, grammaticality of the output as well as fidelity to the content of the input, may be low. Moreover the system should be modular and 'linguistically sound', so that the work done on the Danish-to-English system can be used as a base for other language-pairs, and so that the quality of the output may be improved with further work.

The architecture of the system is a classical transfer-based one. This means it basically consists of three elements:

- The Analysis component giving a linguistic analysis of the source; in my system a Danish text.
- The Transfer component mapping source lexicon and structure on the equivalent target structure (i.e. English).
- The Generation component producing the surface sentence from the output of the transfer.

2 Analysis

2.1 The VISL-project: <http://visl.sdu.dk>

2.1.1 Overview of VISL

The analysis component is the Constraint Grammar implemented at the VISL-project at University of Southern Denmark in Odense. The acronym VISL stands for Visual Interactive Syntax Learning. The VISL-system was developed to teach grammar and grammatical terminology to students at all levels from primary/secondary schools to university. It is an internet-based tool that allows students to 1) see correct analyses, 2) construct analyses themselves using a mouse-based interface and 3) play a range of games training word class categorisation and sentence function analysis. The student may work either on a tree-bank (a corpus of pre-analysed sentences taken from grammar textbooks or the teaching material of the teachers using the system) or on sentences previously unknown to the system, input by the student and parsed automatically by the server. The VISL-project contains 21 languages all in all, 7 of which have system for automatic parsing, the rest having only pre-analysed tree-banks.¹

2.1.2 Automatic analysis in the VISL

The ability to handle input without any constraints is rather uncommon in language technology; few systems can handle longer stretches of e.g. newspaper or literary text. VISL can, by using the technology of Constraint Grammar. Constraint Grammar, which can be seen as extension of two-level morphology (Koskenniemi 1983, Antworth 1990), differs from classical grammar-based language processing on the one hand, by not defining a formal language that has to 'fit' natural language, and from statistic and machine-learning based methods on the other, by being 'a linguists method' allowing for the application of human insights into language in the construction of the system.

Constraint Grammar works by disambiguation. A preprocessor constructs all possible readings of each word in the sentence. 'Father' may be a noun or a verb. As a verb it may be present tense, imperative, infinitive or subjunctive, as a noun it may function as subject or object or any other of a row of functions. The output from the morphological component may look like this, giving wordform (in angular brackets and quoted), lemma (quoted) and a list of semantic, morphological, syntactic and valency tags²:

¹Danish, English, Portuguese, Spanish, French and Esperanto have automatic parsers. German has an automatic parser, but only for morphology. Swedish and Finnish will have automatic parsers by August 2002. Norwegian (Nynorsk and Bokmål) will have in the not so distant future. The following languages have tree-banks (of varying size): Arabic, Bosnian, Dutch, Ancient Greek, Modern Greek, Italian, Japanese, Latin, Latvian, Russian.

²This is the output you get from <http://visl.hum.sdu.dk/visl/en/parsing/automatic/parse.php>. Syntactic tags like 'subject', 'object', 'dative object' etc. are added (and disambiguated) later. I am using the English system for this exposition; the Danish system is equivalent

"<father>"

"father" <Title> N NOM SG

"father" <SVO> <SV> <Rare> V PRES -SG3 VFIN @+FMAINV

"father" <SVO> <SV> <Rare> V INF

"father" <SVO> <SV> <Rare> V IMP VFIN @+FMAINV

"father" <SVO> <SV> <Rare> V SUBJUNCTIVE VFIN @+FMAINV

This 5-ways ambiguity is resolved by applying a set of Constraint Grammar-rules defining finite-state automata removing or selecting (i.e. removing all other) readings according to rules like “if it follows an article it can’t be a verb”, “if there are no other finite verbs, this is one”. The CG-rule system is gradually extended by trial-and-error: new CG-rules are added and old rules are refined until all ambiguity is removed. The resulting system is very robust and gives an output with a very high correctness level.

The Constraint Grammar doesn’t work with constituents larger than words, due to its being tag-based. Tags, also functional syntactic tags, are *lexical* and *unary*. ‘Lexical’ means that it is only the head of a constituent that is marked for function; in “I will buy the blind horse from Germany” it is ‘horse’, not ‘the’, ‘blind’ or ‘from Germany’ that is object. ‘Unary’ means that you mark it as subject, but do not specify what it is a subject of. The syntactic tags in the VISL-system *do* mark whether the governor of a dependent is to be found to the left (e.g. @<SUBJ) or the right (e.g. @SUBJ>), but not the identity of the governor. With this being said: the formalism still allows the representation of the necessary information for tree building, even if it does so indirectly. This can be seen from the applet on the VISL-site that constructs trees from the CG-tagged sentences.

The conclusion to this section is that the analysis component of my machine translation system allows for – or even invites – the broad coverage I aim for and that it returns analyses that probably are sufficiently informative for machine translation, even though they are of a kind much different from what earlier machine translation systems have worked with. If the analyses are not informative enough, then further components may be constructed to supply the needed information, either in the CG-formalism itself or in some other computational formalism (Java or Prolog). The VISL-system *do* furnish semantic tagging, but no real semantic disambiguation. I have myself been working on a case-grammar tagger in the CG-formalism³, but this work has been suspended until the work on the rest of the MT system has begun.

3 Transfer module

The purpose of the transfer module in a machine translation system is to ‘translate’ the source language (Danish) specific analysis that the analysis system provides to a representation that a target language (English) specific generator can use to generate a proper target language sentence. The transfer module of

³Harder (2001), which also gives a more thorough and pedagogical introduction to Constraint Grammar than the space here allows

my system will be written in Prolog and will use code from Trujillo (1999). Of the three transfer methods he mentions, semantic transfer is out of the question, since the CG analysis do not provide a structured sentence-semantic analysis.⁴ Syntactic transfer, i.e. a recursive transformation of source language analysis trees to target language analysis trees, could be used since (as I mentioned above) the CG-analysis *can* be translated to a tree-like structure. I am not certain though, whether the rules are of such a nature that they can be used as a basis for constructing a transfer module. The reason this may be a problem is that there may be configurations that are not disallowed by the tree-building rules, but are still impossible due to features of the CG-analysis.

Thus partly out of necessity, but also partly out of curiosity and interest (and partly due to reasons of analogy I will return to later), I chose the third of the transfer systems that Trujillo (1999) describes, lexical transfer, or as it is also often called in the literature: Shake’n’bake.

3.1 Lexicalist Transfer “Shake’n’Bake”

The governing principle in lexicalist machine translation is that translation relations is not between syntactic or semantic analyses as unified wholes, but between (sets of) words. Constraint Grammar and lexicalist translation are strongly analogous in two ways: the use of constraints and lexicalism.

Constraints The lexicalist approach to MT has developed out of unification style grammars such as Head-Driven Phrase Structure Grammar. The operating principle of unification grammars is that features are inherited upwards in the tree from the words in the sentence to the larger phrases in a structured way and again downwards from the phrases to the words. These features can be seen as constraints on the analysis and semantic interpretation of the sentence, which has given this approach the name ‘Constraint-based linguistics’, which is easily confused with ‘Constraint Grammar’.

Both theories works with sets of rules constraining the possible analyses, even though they do it in different ways. And even if this would not be natural interpretation, the tags in CG can be seen as constraints on interpretation on the lexical level, just as the feature-value pairs in CBL are.

Lexicalism The representations used as analyses in e.g. Head-Driven Phrase Structure Grammar is, seen from a machine translation point of view, extremely redundant. You construct a syntactic tree as well as a semantics (the semantics of the top-node). Moreover all the relevant features and dependency relations have also been ‘percolated’⁵ down into each lexical item, so that each item contains the feature and dependency information relevant to it.

⁴The same argument excludes the non-transfer (a.k.a. interlingua) methods Trujillo (1999) mentions.

⁵This is a popular metaphor in CBL: the coffee percolator where water (features/constraints) is forced up from a tank (a lexical sign in a sentence) by heat and percolates down through ground coffee (the syntactic tree) in to the coffee pot (the other lexical signs in the sentence)

In Shake'n'Bake translation this set of unordered lexical representations is called a bag, and transfer happens by translating one word, or if necessary small groups of words, retaining (or slightly modifying) the dependencies of the source language in the target language.

Both CG and Shake'n'Bake thus works with the word as the unit, and the unary, but order-dependent, dependency representation of the Constraint Grammar should quite easily be translatable to order-independent dependency representation.

4 Generation

I have not yet given great thought to the generator component, producing the English surface sentence from the English bag. I have to reasons to hope that I can keep this module small and easy to build. Firstly, the bag contains the sufficient information, so the grammar may be, to a great amount, lexically driven. Secondly, I do not demand fully grammatical English expressions.

References

- Antworth, E. L.: 1990, *PC-KIMMO: A Two-level Processor for Morphological Analysis*, Summer Institute of Linguistics.
- Harder, S.: 2001, Case grammar-tagging in the VISL constraint grammar, <http://visl.sdu.dk/visl/da/info/casedoc/case.ps>.
- Koskenniemi, K.: 1983, Two-level morphology: A general computational model of word-form recognition and production, *Technical Report 11*, Dept. of General Linguistics, University of Helsinki.
- Trujillo, A.: 1999, *Translation Engines: Techniques for Machine Translation*, Applied Computing, Springer Verlag, London.