Generating natural word orders in a semi–free word order language: Treebank-based linearization preferences for German

Gerard Kempen[1] and Karin Harbusch[2]

[1] Dept. of Psychology, Leiden Univ. and MPI for Psycholinguistics, Nijmegen
kempen@fsw.leidenuniv.nl
[2] Computer Science Dept., Univ. of Koblenz–Landau
harbusch@informatik.uni-koblenz.de

Abstract. We outline an algorithm capable of generating varied but natural sounding sequences of argument NPs in subordinate clauses of German, a semi-free word order language. In order to attain the right level of output flexibility, the algorithm considers (1) the relevant lexical properties of the head verb (not only transitivity type but also reflexivity, thematic relations expressed by the NPs, etc.), and (2) the animacy and definiteness values of the arguments, and their length. The relevant statistical data were extracted from the NEGRA–II treebank and from hand-coded features for animacy and definiteness. The algorithm maps the relevant properties onto "primary" versus "secondary" placement options in the generator. The algorithm is restricted in that it does not take into account linear order determinants related to the sentence's information structure and its discourse context (e.g. contrastiveness). These factors may modulate the above preferences or license "tertiary" linear orders beyond the primary and secondary options considered here.

1 Introduction

Computational sentence generators should be able to order constituents in agreement with linearization preferences and habits of native speakers/writers. This knowledge can be attained by exploiting text corpora (cf. [1]). In the following we concentrate on extracting appropriate word order rules for German, a (semi-)free word order language.

Target languages with strict word order rules do not present much of a problem here although the grammaticality contrast between examples such as *Pat picked a book up* and *?Pat picked a very large mint–green hardcover book up* [2, p. 7] shows that, even in English, knowledge of linear order preferences comes in handy. In the case of (semi–)free word order languages, the problem of how to select natural sounding permutations of constituents from among those licensed by the grammar is much more widespread. Sentence generators striving for natural and varied output, e.g., in question–answering systems or computer–supported language training environments, should neither select the same permutation at all times, nor produce the various grammatical permutations at random.