Trost et al. (2005) and Trnka et al. (2005), we assume that one additional keystroke is required for the selection of a word from the list and that a space is automatically inserted afterwards. Note also that words, which have already occurred in the list, will not reappear after the next character has been inserted.

The perplexity measure, which is frequently used to assess statistical language models, proved to be less accurate in this context. We still present perplexities as well in order to provide comparative results.

2 Language modeling and semantics

2.1 Statistical Language Models

For about 10 to 15 years statistical language modeling has had a remarkable success in various NLP domains, for instance in speech recognition, machine translation, Part-of-Speech tagging, but also in word prediction systems. N-gram based language models (LM) estimate the probability of occurrence for a word, given a string of *n*-1 preceding words. However, computers have only recently become powerful enough to estimate probabilities on a reasonable amount of training data. Moreover, the larger *n* gets, the more important the problem of combinatorial explosion for the probability estimation becomes. A reasonable trade-off between performance and number of estimated events seems therefore to be an *n* of 3 to 5, including sophisticated techniques in order to estimate the probability of unseen events (smoothing methods).

Whereas n-gram-like language models are already performing rather well in many applications, their capacities are also very limited in that they cannot exploit any deeper linguistic structure. Long-distance syntactic relationships are neglected as well as semantic or thematic constraints.

In the past 15 years many attempts have been made to enrich language models with more complex syntactic and semantic models, with varying success (cf. (Rosenfeld, 1996), (Goodman, 2002) or in a word prediction task: (Fazly and Hirst, 2003), (Schadle, 2004), (Li and Hirst, 2005)). We want to explore here an approach based on *Latent Semantic Analysis* (Deerwester et al, 1990).

2.2 Latent Semantic Analysis

Several works have suggested the use of *Latent Semantic Analysis* (LSA) in order to integrate semantic similarity to a language model (cf. Bellegarda, 1997; Coccaro and Jurafsky, 1998). LSA models semantic similarity based on co-occurrence distributions of words, and it has shown to be helpful in a variety of NLP tasks, but also in the domain of cognitive modeling (Landauer et al, 1997).

LSA is able to relate coherent contexts to specific content words, and it is good at predicting the occurrence of a content word in the presence of other thematically related terms. However, since it does not take word order into account ("bag-of-words" model) it is very poor at predicting their actual position within the sentence, and it is completely useless for the prediction of function words. Therefore, some attempts have been made to integrate the information coming from an LSA-based model with standard language models of the n-gram type.

In the LSA model (Deerwester et al, 1990) a word w_i is represented as a high-dimensional vector, derived by *Singular Value Decomposition* (SVD) from a term × document (or a term × term) co-occurrence matrix of a training corpus. In this framework, a context or history h (= w_1, ... , w_m) can be represented by the sum of the (already normalized) vectors corresponding to the words it contains (Landauer et al. 1997):

$$\vec{h} = \sum_{i=1}^{m} \vec{w}_i \qquad (2)$$

This vector reflects the meaning of the preceding (already typed) section, and it has the same dimensionality as the term vectors. It can thus be compared to the term vectors by well-known similarity measures (scalar product, cosine).

2.3 Transforming LSA similarities into probabilities

We make the assumption that an utterance or a text to be entered is usually semantically cohesive. We then expect all word vectors to be close to the current context vector, whose corresponding words belong to the semantic field of the context. This forms the basis for a simple probabilistic model of LSA: After calculating the cosine similarity for each word vector $\vec{w}_i$ with the vector $\vec{h}$ of the current context, we could use the normalized similarities as probability values. This probability distribution however is usually rather flat (i.e. the dynamic