Exploiting long distance collocational relations in predictive typing

Johannes Matiasek
Austrian Research Institute for
Artificial Intelligence
Schottengasse 3
A-1010 Vienna, Austria
john@oefai.at

Marco Baroni
Scuola Superiore di Lingue Moderne
per Interpreti e Traduttori
Università di Bologna
Corso della Repubblica 136
I-47100 Forlì, Italia
baroni@sslmit.unibo.it

Abstract

In this paper, we report about some preliminary experiments in which we tried to improve the performance of a state-of-the-art Predictive Typing system for the German language by adding a collocation-based prediction component. This component tries to exploit the fact that texts have a topic and are semantically coherent. Thus, the appearance in a text of a certain word can be a cue that other, semantically related words are likely to appear soon. The collocation-based module exploits this kind of topical/semantic relatedness by relying on statistics about the co-occurrence of words within a large window of text in the training corpus. Our current experimental results indicate that using the collocation-based prediction module has a small but consistent positive effect on the performance of the system.

1 Introduction

Written communication is a vital factor in human society. Impairments which lead to a reduction of typing speed, therefore, severely influence quality of life and cut off a person from equal participation in the information society.

Since languages display a certain degree of redundancy and predictability, low-speed typists can be supported by Predictive Typing (PT) systems. Such systems attempt to predict subsequent portions of text by analyzing the text already entered by the writer. Character-by-character text entry is replaced by making a single selection as soon as the desired word or word sequence is offered by the system in the selection menu.

The current most popular PT technology (see, for example, (Carlberger, 1998), (Copestake, 1997)) relies on a statistical approach based on the probability of n-grams, i.e., the continuations proposed by the system are strings that often follow the string the user just typed (string frequencies are extracted from a training corpus).

As part of the European Community funded FASTY project, we are currently developing a PT system that augments standard word-based n-gram prediction with part-of-speech-based prediction (an idea already implemented with success by (Carlberger, 1998)), grammar-based prediction, compound processing, inflectional analysis and a user lexicon ((Matiasek et al., 2002), (Baroni, 2002), (Baroni et al., 2002a), (Baroni et al., 2002c)). The FASTY system is being implemented for the Dutch, French, German and Swedish languages.

In this paper, we report about some preliminary experiments in which we attempt to further improve the performance of the German version of FASTY by adding what we label a *collocation-based* prediction module.

The idea behind collocation-based prediction is the following. The standard n-gram model predicts words on the sole basis of their immediate context (n preceding words). However, since texts typically have a topic and are semantically coherent, there are also long-distance relationships between the words in a text that could be exploited to improve prediction.

The appearance in a text of a certain (content) word can be a cue that other, semantically related words are likely to appear soon. For example, if a current newspaper article contains the word *Iraq*, it is also quite likely that it will contain words such as *Bush, Saddam, UN, war*. The collocation-based module exploits this kind of topical/semantic relatedness by relying on statistics about the co-occurrence of words within a large window of text in the training corpus.

As far as we know, this is the first attempt to incorporate a collocation-based module in a PT system (beyond the "recency promotion" mechanism proposed by (Carlberger, 1998), which we discuss in 3 below). However, the same idea was exploited with success by (Rosenfeld, 1996), who integrated a similar component in a statistical language model aimed at automatic speech recognition applications. The work of Rosenfeld constitutes a major source of inspiration for the model we are presenting here.

The results of the experiments we report below indicate that using the collocation-based prediction module has a small but consistent positive effect on the performance of the system.

The remainder of this paper is organized as follows: In 2, we shortly describe PT systems and how they are evaluated. In 3, we present our collocation-based prediction model. In 4, we report about the experiments we ran with the collocation-based module and FASTY. Finally, in the conclusion (section 5) we discuss some of the strategies we plan to follow in order to improve the performance of the current model.

2 Predictive typing for augmentative and alternative communication

While they have other applications as well, PT systems are an important component of augmentative and alternative communication (AAC) devices, i.e., software and possibly hardware typing aids for disabled users (see for example (Carlberger, 1998), (Copestake, 1997), (McCoy and Demasco, 1995)). Besides functioning as typing aids, such devices can be connected to speech synthesizers in order to allow oral communication to people who cannot speak.

PT systems provide the user with a *prediction window*, i.e., a menu that, at any time, lists the most likely next word candidates, given the input that the user has typed until the current point. If the word that the user intends to type next is in the prediction window, the user can select it from there. Otherwise, the user will keep entering letters, until the target word appears in the prediction window (or, of course, until she finishes typing the word).

PT systems typically base their predictions on various forms of n-gram statistics extracted from one or more training corpora.

The (percentage) *keystroke savings rate* (*ksr*) is a standard measure used in AAC research to evaluate word prediction systems (see, for example, (Carlberger, 1998) and (Copestake, 1997)). The *ksr* can be thought of as the number of keystrokes, in percent, that a "perfect" user could save by employing the relevant word predictor to type a certain text, over the total number of keystrokes that are needed to type the same text without using the word predictor.

Usually, the *ksr* is defined by

$$ksr = (1 - \frac{k_i + k_s}{k_n}) * 100 \qquad (1)$$

where: k_i is the number of input characters actually typed, k_s is the number of keystrokes needed to select among the predictions presented by the model and k_n is the number of keystrokes that would be needed if the whole text was typed without any prediction aid. Here, we make the reasonable assumption that the user will need one keystroke to select among the predictions in the prediction window, i.e., that k_s equals 1.

The *ksr* is influenced not only by the quality of the prediction model but also by certain parameters of the prediction process, most importantly by the number of predictions to select from the user is presented with, i.e., by the size of the prediction window. In the simulations we report about below, we assumed a prediction window of 5 words. This is a number of predictions that can be scanned quickly by most users (cf. (Hunnicutt and Carlberger, 2001)), and it is the default setting in FASTY.[1]

[1] However, our tests have shown that an increase of 1.5-2 percent points in *ksr* can be expected if the number of predic-

Using *ksr* as an evaluation measure has the drawback that an exact computation of the *ksr* is possible only by running a simulation of the prediction process. However, it is the measure that reflects best the benefits a disabled typist has when using a word prediction system.

Preliminary tests show that the German version of the FASTY system can reach, when the training and testing corpora are similar, a *ksr* as high as 66%. As far as we know, this is the highest *ksr* achieved by a system implemented in a language other than English.

3 Collocation-based prediction

The collocation-based prediction model exploits the fact that, because of topical/semantic factors, the appearance of a word in a text can be a cue that other, related words are likely to appear soon.

We use the term *collocation* (see, for example, (Manning and Schütze, 1999), chapter 5) in a rather loose sense, to refer to any pair of words $w1$ and $w2$, where the occurrence of $w1$ in a text makes the appearance of $w2$ in the same text more likely. In particular, adapting the terminology of (Rosenfeld, 1996), we refer to $w1$ as the *trigger*, and $w2$ as the *target*.

The Swedish PT system described in (Carlberger, 1998) incorporates a simple and effective form of collocation-based prediction, that Carlberger labels *recency promotion*. Recency promotion exploits the fact that words (more interestingly, content words) are likely to occur more than once in the same text. Thus, all else being equal, a candidate word that already occurred in the current text should be preferred over a candidate word that is new in the current text. Carlberger reports that recency promotion brings a significant improvement in terms of keystroke savings.

In certain cases, the *user lexicon* as implemented in the FASTY system can provide a very simple form of recency promotion. The aim of the user lexicon is to provide resources targeted at the user's specific vocabulary and style in order to bias the general language model towards the user's needs. The user lexicon can be extracted from texts the user has written previously or that reflect her needs, or it can be empty. In any case, during the operation of the FASTY system, all content words the user enters are recorded (along with the corresponding bigrams) and immediately added to the user lexicon. At the end of the session the user may decide to save the augmented user lexicon or to abandon the changes.

The immediate augmentation of the user lexicon has the (desirable) effect that words previously unknown to the system (e.g., proper names) may be predicted immediately after they have been typed the first time, and (more to the point of the current discussion) the probability of the words that the user typed up to this point in the current session is boosted up. However, when starting with an empty lexicon, these words may disturb the prediction process, since each of them receives a very large portion of the probability mass reserved to the user lexicon. Currently, there is no adjustment or decay implemented, since we assume a non-empty user lexicon, in which case the distorting effects are minimal. Still, preliminary tests indicate that even the use of an initially empty user lexicon can bring up the *ksr* by about 3 percent points.

Whether a more general collocation-based prediction system, such as the one we are proposing here, will bring further improvements beyond those provided by recency promotion is not a trivial issue: (Rosenfeld, 1996) has showed that, in the English corpus he analyzed, the best predictor for the occurrence of a word was the word itself in 68% of the cases, and in 90% of the cases the word itself was among the 6 best predictors .[2]

In the next two subsections, we describe how we extract collocations (in the sense given above) from the training corpus (3.1), and the way in which we implement collocation-based prediction for the PT task (3.2).

tions is raised to 7, and a gain of 3 percent points or above if 9 predictions are used.

[2]These results were obtained by considering wordforms, i.e., treating different members of the same inflectional paradigm as different words. Given that Rosenfeld also reports that words sharing the same stem are good predictors of each other, we expect that the proportion of good "self-triggers" would be even higher if the counts had been based on lemmas. On the probability that the same word will occur more than once in the same text, see also (Church, 2000).

3.1 Collocation extraction

In order to score the "textual association strength" between words, we calculate the (pointwise) *mutual information* for each pair of content words[3] in our training corpus.

Mutual information, first introduced to computational linguistics by (Church and Hanks, 1989), is one of many measures that seems to be roughly correlated to the degree of semantic relatedness between words.[4] The mutual information between two words $w1$ and $w2$ is given by:

$$I(w1, w2) = \log \frac{Pr(w1, w2)}{Pr(w1)Pr(w2)} \qquad (2)$$

Intuitively, the larger the deviation between the empirical frequency of co-occurrence of two words and the expected frequency of co-occurrence if they were independent, the more likely it is that the occurrence of one of the two words is *not* independent from the occurrence of the other.[5]

In particular, (Brown et al., 1990) observed that when mutual information is computed in a non-directional fashion and by counting co-occurrences of words within a relatively large window, but excluding "close" co-occurrences (which would tend to capture multi-word names, phraseological units and lexicalized phrases), the measure identifies semantically related pairs.

Since this is the type of relationship we are after (dependencies between adjacent words are already captured by the standard word-based bigram model), we computed mutual information by considering, for each pair, only co-occurrences within a maximal window of 50 words and outside a minimal window of 2 words.[6]

Notice that, since directionality is not taken into account (i.e., both $w1...w2$ and $w2...w1$ are counted as occurrences of the same bigram), for each $w1$ $w2$ pair (where the first word is the trigger, the second the target) with a certain mutual information, there is also an equivalent $w2$ $w1$ pair with the same mutual information.

The top 20 collocates of the word *Hund* "dog" that were found in this way were:

	MI		MI
eingeschläfert	15.297	Hündin	13.853
Herrchen	15.297	Schwanz	13.831
Terrier	14.841	Hüte	13.771
Hundes	14.764	Haustiere	13.744
Vierbeiner	14.548	Quarantäne	13.618
Rottweiler	14.493	Haustier	13.583
Frauchen	14.417	Hund	13.513
biß	14.249	Bohlen	13.481
gebissen	14.007	Katze	13.371
Leine	13.863	Besitzerin	13.074

As this example shows, in general the collocations extracted using this method are quite plausible.

3.2 Generation of collocation-based completions during prediction

The collocation-based component has access to the list of trigger-target pairs that was constructed from the training corpus as we just described. At startup time, this list is read by the predictor and a hash table associates each trigger with a list of its targets and the corresponding scores. These scores are stored as integers and are derived from the original mutual information measure by multiplication with an adjustable factor[7].

During the prediction process, the collocation-based model keeps track of the last n words entered (the window size n is an adjustable parameter – see discussion in 4 below) and it maintains a trie of current targets for prediction. Every time the window changes (i.e., the user starts to type the next word) the trie is newly computed. For each trigger word in the sliding context window, the associated targets and their scores are looked up in the trigger-target hash table. Each target word together with its score is entered into the collocation-based prediction trie. Thereby the score is interpreted as a count that can be used during prediction for computing the probability of the suggested completions.

[3] We determine which words are content words by running the whole corpus through the XEROX German morphological analyzer (Karttunen et al., 1997).

[4] We also experimented with the log-likelihood ratio measure proposed by (Dunning, 1993). However, the results were consistently worse than with mutual information.

[5] (Rosenfeld, 1996) uses the closely related *average* mutual information measure to identify trigger-target pairs.

[6] See (Baroni et al., 2002b) and (Baroni et al., 2002c) for other studies in which we used this technique to identify semantically related words.

[7] We experimented also with nonlinear scores, but these performed worse.

The textual distance of the trigger to the current word is also taken into account. We use a *decay function* that reduces the score of a target proportional to the distance of its trigger from the current word. Thus, the prediction score S_p of a word at the current input position $n+1$ (given a trigger window size of n) is computed as the sum of the decayed trigger scores S_t of this word as a target of all triggers w_i in the context window:

$$S_p(w_{n+1}) = \sum_{i=1}^{n} \frac{S_t(w_i, w_{n+1})}{n-i+1} \qquad (3)$$

The probability assigned to a possible completion by the collocation-based model is given by the sum of the scores associated with the pairs in which the candidate completion is a target, divided by the sum of the scores of all currently active targets (i.e., the targets associated with the last n words).

In the experiments reported below, we integrated the collocation-based model in a version of the FASTY system consisting of word-based bigram and unigram models and a tag-based trigram model.[8]

The probability estimate for the (completion of the) next word is derived by interpolating the unigram probability of the word (given the already entered, but possibly empty, prefix), the bigram probability of the word (given the previous word and the prefix) and a term derived from the tag model. This term is computed as in (Carlberger, 1998) by summing up the probability of each tag to occur with the next word (according to the trigram-based tag model) multiplied by the probability that this tag occurs with the predicted word (for details, see (Carlberger, 1998)).

The collocation-based model is integrated into the system by interpolating the word-based unigram statistics with the trigger-based target word probabilities.

We decide to combine the collocation-based model with the unigram model since, in earlier experiments, we observed that reducing the weight assigned to the bigram model from its empirically determined optimum has a strong negative impact on *ksr*. Moreover, conceptually, the semantic information exploited by the collocation-based model seems to belong with the lexical factors reflected in unigram statistics, rather than with the mostly syntactic factors that govern the other models.

In the current version of FASTY, the optimal interpolation weights are empirically determined via hand-tuning. Clearly, this is an aspect of the system where further work is needed.

4 Experiments with collocation-based prediction

Extraction of trigger-target pairs was performed on the APA (Austrian Press Agency) newswire articles from January to October 1999 (containing 27,251,629 words). We collected the 9,235,073 long-distance bigrams made of content words that occurred at least 5 times in this corpus. Only pairs of words occurring in the same article and within the window specified in 3.1 above were considered. After calculating the mutual information scores and pruning away pairs with a mutual information score of less than 10, as well as pairs containing punctuation, all-caps abbreviations, and proper names (as these might be only relevant to the training corpus), we arrived at a manageable set of 306,367 pairs built from 33,693 trigger words.

The other word-based models and the tag-based model were trained on the same corpus.

For a first set of experiments, we created a test corpus distinct from the training corpus by randomly selecting 11 articles from the Frankfurter Rundschau Corpus (newspaper articles from June 29 to July 12 of 1992), containing 10,109 words (70,984 characters).

Simulation runs were performed on each article separately, since the articles are topically different and it was undesirable (and also not conforming to the practical use of FASTY) to carry over the triggers from one article to the other. The total *ksr* scores were computed from the sums of the different character counts according to equation (1). Note that the figures for the number of typed characters include the keystrokes needed for selecting

[8] The word-based unigram model is augmented by an (independently needed) morphological lexicon, that serves as a backup resource if the statistical model runs out of predictions. As mentioned above, the full FASTY system also includes compound prediction, grammar-based prediction and a user lexicon.

among the predictions.

As a baseline we computed the *ksr* without employing the collocation-based model. The interpolation weights were set to 0.2 for the word unigrams, 0.6 for the word bigrams, and 0.2 for the tag model. These settings have proven useful in previous experiments.

The results of the baseline experiment are shown in Tab. 1.

sample	chars total	chars typed	ksr
#1	3924	2092	46.69
#2	5517	3021	45.24
#3	5215	2514	51.79
#4	4887	2646	45.86
#5	10121	6104	39.69
#6	7099	3953	44.32
#7	6658	3603	45.88
#8	2683	1459	45.62
#9	4827	2607	45.99
#10	4399	2054	53.31
#11	15654	7297	53.39
Total	70984	37350	47.38

Table 1: Baseline *ksr*

In order to find an optimal setting for the parameters of collocation-based prediction, we experimented with different sizes of the trigger context window and with and without distance-based decay (cf. equation 3). It turned out that a change in window size within the 20-to-50 word range does not have a significant effect on performance. Reducing the trigger window to 10 words, however, leads to a noticeable degradation in performance.

Using the decay function leads to a slight improvement in performance (especially in combination with larger windows).

Although the differences are rather small (except for the simulations with very narrow windows), we consistently achieved the best results using a trigger window of 50 words together with the decay function. The results reported here are based on these settings.

We used the same weights as in the baseline experiments for the word-based bigram model and for the tag-based trigram model. The total weight assigned to the word-based unigram model and to the collocation-based model was kept constant as 0.2, while the relative weights of these two models were systematically varied.

The x-axis in the figures below is scaled to show the percentage of weight assigned to collocation-based prediction over the total weight allocated to the word-based unigram and collocation-based models together (for shortness, this is called "unigram weight" in the figures).

In Fig. 1 the changes in *ksr* depending on the weight of collocation-based prediction are shown for the different test documents. It turns out that collocation-based prediction leads to an improvement in *ksr* for each document. However, these improvements tend to be rather small, and they are not always obtained with the same weight settings.

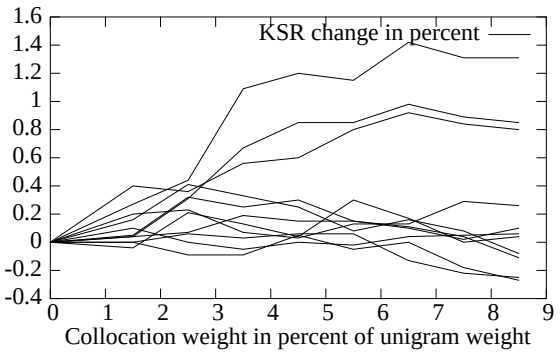

Figure 1: *Ksr* gain for different texts

The overall improvement is shown in Fig. 2. The best overall *ksr* gains are achieved when the weight of collocation-based prediction contributes about 5-7% of the weight allocated to the word-based unigram and collocation-based models.

This is also the range in which the sample texts benefitting most from collocation-based prediction showed the highest *ksr* gains.

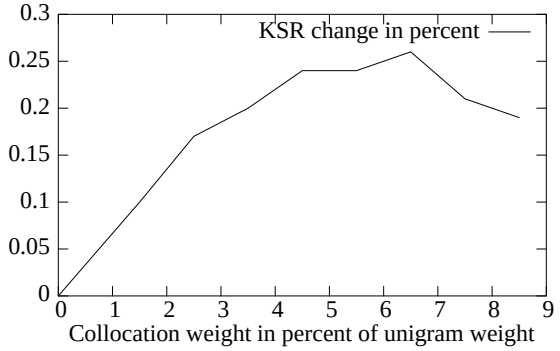

Figure 2: Overall *ksr* gain

We then conducted a second set of experiments using the same training corpus but a much larger

test set. This test set was constructed by extracting 999 articles from the Donaukurier Corpus, for a total of 300,449 words (approximately 2.23MB of characters).

Simulation runs were performed (on each article separately) without and with the collocation-based model. For the simulation without collocations, we used the same settings as above. For the simulation with collocations, we used the weights that gave the best results in the previous tests (unigrams: 0.187, collocations: 0.013, bigrams: 0.6, tag model: 0.2).

Again, using the collocation-based model led to a (small) improvement in total *ksr*: from 41.475 without collocation to 41.653 with the collocation-based model. The distribution of *ksr* changes across articles was as follows: in 644 cases, the collocation-based model led to an improvement in *ksr* over the baseline; in 274 cases, the collocation-based model caused a decline in *ksr*; in 81 cases, there was no effect on *ksr*.

A paired two-sided Wilcoxon signed rank test comparing the *ksr*'s for each article with and without the collocation model showed that the difference between the two conditions is highly significant ($V = 91420.5, p < .00000$).

Thus, the results of the second experiment confirmed the small but consistent positive effect of collocation-based prediction that we found in the fist experiment.

The effects of collocation-based prediction in both experiments are document-specific, probably depending on how well the topic of each test document is represented in the training corpus.

5 Conclusion

In this paper, we reported about some preliminary experiments in which we added a collocation-based module to the German FASTY system, in order to exploit semantic/topical dependencies among words.

The results we obtained in these experiments are encouraging, in that they indicate that adding collocation-based prediction to FASTY does improve *ksr*. However the magnitude of this improvement is small. Thus, we are currently exploring various strategies to boost up the performance of collocation-based prediction.

The way in which we interpolate the collocation-based model with the other prediction models appears to be extremely important. Thus, we should perhaps adopt a more sophisticated interpolation scheme than the current hand-tuning approach. For example, we could experiment with an EM-like algorithm or a Maximum Entropy model along the lines of (Rosenfeld, 1996).

The results also suggest that the semantic coverage of the long-distance bigram pool should perhaps be extended via training on larger corpora. However, care should be exercised in ensuring that the resulting collocation database is still manageable in size and that it does not introduce noise that disturbs the prediction process.

Moreover, rather than collecting triggers and targets directly from wordforms, the training corpus should perhaps be lemmatized, and the relevant statistics should be computed for lemmas. This is appealing because semantic relatedness does not depend on the particular inflected wordforms involved (especially if long distance dependencies are considered) but rather on lemmas. This would also lead to a more compact collocation database, which is certainly desirable.

While lemmatization does not pose a problem for the collection of long-distance bigrams, during the prediction process we need to present wordforms, rather than lemmas, to the user. Thus, strategies should be developed to come up with the most likely wordform(s) for each lemma. The FASTY system contains a grammar-based module (not described in the present paper) that checks whether predicted wordforms are grammatically possible in the current context. Perhaps, this module could be employed to generate appropriate wordforms.

Instead of relying simply on the patterns of co-occurrence of two words, we could also look at their contextual similarity, as measured by the cosine of vectors representing other words they occur with (see (Manning and Schütze, 1999), chapter 8).

Finally, semantically related words could be clustered into classes (for example, adopting the clustering algorithm we already use with some success for the compound head prediction component of FASTY – see (Baroni et al., 2002c)), so

that the trigger-target model could be replaced or combined with a more general class-based model.

We hope that further work along the lines we just sketched will transform the promising but meager improvements in *ksr* we report here into a more sizeable positive boost of the performance of the system.

At that point, we will make collocation-based prediction a stable component of the FASTY system, and we will adapt it to the other languages of the FASTY project.

Acknowledgements

We would like to thank the Austria Presse Agentur for kindly making the APA corpus available to us. This work was supported by the European Union in the framework of the IST programme, project FASTY (IST-2000-25420). Financial support for ÖFAI is provided by the Austrian Federal Ministry of Education, Science and Culture.

References

M. Baroni. 2002. FASTY: A multilingual approach to text prediction, *Elsnews* 11.2, 11-12.

M. Baroni, J. Matiasek. and H. Trost. 2002a. Predicting the Components of German Nominal Compounds. In F. van Harmelen (ed.) *ECAI 2002. Proceedings of the 15th European Conference on Artificial Intelligence*. IOS Press, Amsterdam.

M. Baroni, J. Matiasek and H. Trost. 2002b. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. *Proceedings of the ACL-2002 Workshop on Morphological and Phonological Learning*, 48-57.

M. Baroni, J. Matiasek and H. Trost. 2002c. Wordform- and class-based prediction of the components of German nominal compounds in an AAC system. *Proceedings of COLING 2002*.

P. Brown, P. Della Pietra, P. DeSouza, J. Lai, and R. Mercer. 1990. Class-based n-gram models of natural language. *Computational Linguistics*, 18, 467-479.

J. Carlberger. 1998. *Design and Implementation of a Probabilistic Word Prediction Program*, Royal Institute of Technology (KTH).

K. Church. 2000. Empirical estimates of adaptation: The chance of two Noriegas is closer to p/2 than p^2. *COLING 2000*.

K. Church and P. Hanks. 1989. Word association norms, mutual information, and lexicography. *Proceedings of ACL 27*, 76–83.

A. Copestake. 1997. Augmented and alternative NLP techniques for augmentative and alternative communication, *Proceedings of the ACL workshop on Natural Language Processing for Communication Aids*.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19.1, 61-74.

S. Hunnicutt and J. Carlberger. 2001. Improving word prediction using Markov Models and heuristic methods. *Augmentative and Alternative Communication* 17, 255-264.

L. Karttunen, K. Gaál, and A. Kempe. 1997. *Xerox Finite-State Tool*. Xerox Research Centre Europe, Grenoble.

K. McCoy, and P. Demasco. 1995. Some applications of Natural Language Processing to the field of Augmentative and Alternative Communication. *Proceedings of the IJCAI-95 Workshop on Developing AI Applications for People with Disabilities*.

C. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MASS.

J. Matiasek, M. Baroni and H. Trost. 2002. FASTY: A multi-lingual approach to text prediction, *Proceedings of the 8th International Conference on Computers Helping People with Special Needs.*

R. Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modeling. *Computer Speech and Language* 10, 187–228.