Domain–specific Disambiguation for Typing with Ambiguous Keyboards

Karin Harbusch Saša Hasan Hajo Hoffmann
Michael Kühn Bernhard Schüler
Universität Koblenz-Landau
Fachbereich Informatik
Universitätsstr. 1, D–56070 Koblenz, GERMANY
{harbusch,hasan,rriipp,kuehn,bernie}@uni-koblenz.de

Abstract

In this paper, we investigate whether and how domain–specific corpora increase precision of word disambiguation for typing on an ambiguous keyboard. Basically, the disambiguation for our ambiguous keyboard with three letter keys is based on language–specific word frequencies of the lexicon CELEX (in this study English and German is dealt with). The more specific frequency information is extracted from texts in the special domains of school homework in three subjects and articles in two different scientific areas. All in all, we could not always reach a better performance by deploying domain–specific predictions. As a general solution we propose an interpolated language model combining both the general and the specific language model. For all our domains good results — compared to an ideal prediction on the basis of all available models — could be achieved by this method.

1 Introduction

Ambiguous keyboards enable motor impaired people and users of watch–sized devices to communicate by means of a reduced number of letter keys (down to 3). The number of keystrokes is not necessarily reduced as compared to typing on an ordinary keyboard. If arbitrary words are coded this way, different words receive the same encoding, i.e. the decoding function of such a keyboard is not injective. Thus, the user has to select the intended word from a list of word suggestions (ordered, e.g., by word frequencies in a given lexicon). Nevertheless, with nine letter keys only eight percent of 24,500 English words are actually encoded ambiguously and thus need a disambiguation step (Witten, 1982).

The advantages of an ambiguous keyboard with word disambiguation for users of Augmentative and Alternative Communication (AAC) devices are outlined by Kushler (1998). The efficiency of an ambiguous keyboard approaches one keystroke per letter. Beside literacy, no memorization of special encodings is required. Attention to the display is wanted only after the word has been typed.

The disambiguation for an ambiguous keyboard that is *personalised* to the user's vocabulary or *domain–specific* in a given communicative situation promises increased efficiency. One approach to a user–tailored disambiguation is a system that automatically adapts to the currently observed language usage. However, as observed by Raskin (2000) with regard to user interfaces, adaptation in a system makes it more difficult for the user to know how the system works and to operate it blindly. Another obstacle of a purely adaptive system derives from the short size of the typed texts in a given situation (the homework corpus we examine here has a size of 8,027 words).

In this paper, we investigate an approach to personalise the disambiguation component of our communication aid in a non–adaptive, user–controlled manner by employing several *domain–*

specific language models. Although the models themselves are static, a combination of them leads to a better disambiguation with respect to the chosen domain.

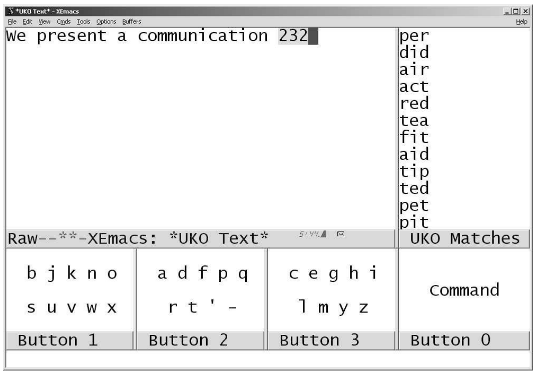
The paper is organised as follows. In Section 2, we delineate our prototypical ambiguous keyboard (called *UKO–II*). We explore various domains in Section 3 as for domain–specific models to supplement personalised disambiguation. In Section 4, the evaluation results are presented. In Section 5, we outline related work and finally, we address open questions and future work.

2 The UKO–II communication aid

Our communication aid UKO–II resides on the programmable and extendable XEmacs text editor[1]. It provides many text entry and manipulation functions useful in our context. Operating system support (e.g. sockets), basic applications like mail and a development environment including extensive documentation are at the programmer's fingertips. All components of the communication aid dealing with input/output have been implemented as Emacs Lisp modules[2].

2.1 Text entry interface

UKO–II provides 2 + n ($n \geq 2$) windows (in Figure 1) n is 4). The window in the left upper panel represents the text editing buffer. The window in the right upper panel lists all word suggestions for the ambiguously typed word currently under consideration. In the lower part all keys are shown. The number n (≥ 2) of buttons has been specified in advance. This parameter[3] depends on the user's motor functions or on the buttons available on the device. A genetic algorithm is used to calculate a near–optimal distribution of letters in order to minimise the length of suggested word lists (Kühn and Garbe, 2001).



We present a communication 232

per
did
air
act
red
tea
fit
aid
tip
ted
pet
pit

Figure 1: UKO-II Emacs text editing interface.

In our project, the keyboard is tailored to a user with cerebral palsy. No more than four buttons can be accessed directly. Three buttons provide ambiguous letter keys labeled by sets of letters for typing in the text editing window. The fourth button invokes meta–level commands such as letter deletion and word disambiguation. In this paper, we assume n = 4 throughout.

Words are entered by pressing the corresponding ambiguous key once for each letter. Only after the word is completed, the user disambiguates the input by selecting the intended word in a list of hits provided by the language model. Figure 1 depicts the situation after the word "aid" has been typed — by pressing the second, the third and the second button again (key sequence "232") — and before the user selects the targeted word in the list of suggestions. If the target word is not known to the system, it is possible to spell the word and include it in the lexicon for future use[4]. Other actions in the command mode provide text navigation and editing as well as invocation of the speech output system. These actions are triggered either by overloading the three letter keys with commands, or by entering and disambiguating a command name.

2.2 The static word list model

The disambiguation step, resulting in the list of word suggestions as illustrated in the right upper window in Figure 1, is based on a word list ordered by the word frequencies obtained from large corpora. In the UKO–II system, the ordering is based on the word frequency information for English and German in the CELEX lexical database (Baayen et

[1] http://www.xemacs.org/

[2] See http://www.gnu.org/manual/emacs-lisp-intro/ for an introduction to programming in Emacs Lisp

[3] If less than four physical buttons are available, the keys have to be selected on a virtual keyboard (*scanning*).

[4] Currently, no default frequency is assigned to these words. Instead, these personal word entries are added to the end of the list computed by the word model. Obviously, more sophisticated combinations will be incorporated in the next release of UKO.

al., 1995) taken from mixed corpora of written and spoken sources. The English word form frequencies are based on the 17.9 million word COBUILD corpus prepared by the University of Birmingham in 1991. The German word frequencies stem from the 6 million word corpus collected by the Institute of German Language in Mannheim in 1984[5]. Table 1 gives an overview of the size of the lexica. By *token* we refer to the number of occurrences of a word in the text whereas by *type* we indicate the set of different words (Manning and Schütze, 2001).

Word list	Word tokens	Word types
English	16,747,857	74,206
German	3,943,923	316,376
German (New)	4,013,854	321,380

Table 1: Word form tokens and types in the English and German word–frequency lists in CELEX.

Table 2 gives the maximum and the average length of the word lists, respectively, assigned to ambiguous codings with their variances. The word lists can grow quite long with 50 entries in the English word list with the ambiguous code '3111'[6] or even 75 entries in the modified German word list with ambiguous encoding '12223'[7]. But on average each ambiguous coding fits to two corresponding words if we take an ambiguous keyboard with three letter keys.

Word list	Max.	Avg.	Var.
English	50	2.017	11.755
German	75	1.915	8.669
German (New)	75	1.919	8.776

Table 2: Number of hits for ambiguous codings in the English and German word lists.

The ambiguous encodings used in our experiments are typed on language–specific keyboards[8] with three letter keys (see Figure 2).

English

b j k n o s v w x u	a d f p q r t - '	c e g h i l m y z

German

a g j l m q r w z ä	c f h k o s t u v x y ü ß	b d e i n p ö -

Figure 2: Keyboard layout used in UKO–II for English and German.

We denote by $\mathrm{rank}(w, c, p)$ the position of a word w in the corresponding word list of code c ordered by a language model with the probability function p. A performance metric of such a language model for disambiguating a given text is the *precision of rank* r, i.e. the percentage of correct disambiguations if only the r most likely suggestions according to the underlying language model are considered. Assuming $r = 1$, the precision of rank 1 — also called the *simple precision* (cf. Jurafsky and Martin (2000, p. 639) for considerations in the area of word sense disambiguation) — turns out to be around 68.75% for English and 73.5% for German CELEX word frequency lists, respectively. Table 3 presents the upper values of the precision of rank 1 to 5 for the CELEX lexica in English and German. Notice that these measurements are upper bound values as the language model and the test set cannot be assumed to be identical. Thus actual typing performs considerably weaker (cf. upper left panel in Figure 3).

Rank	1	2	3	4	5
English	68.75	86.77	92.73	95.24	96.63
German	73.50	87.67	92.76	95.41	96.85

Table 3: Precision of rank = $1, \ldots, 5$ for the CELEX lexica in English and German.

In Section 4, we evaluate domain–specific test corpora with the CELEX frequencies and compare these values to values computed for several more specific language models.

In general, a static base language model as presented here will be limited in its representativity for arbitrary texts. As our goal is to improve the precision of the disambiguation step, we propose to complement the base language model with lan-

[5] Since the orthography in the German word list does not respect the spelling reform resolved in 1996, we duplicate in "German (New)" all words that are spelled differently after the reform with the corresponding frequencies of the old spellings. In the following, we always use the modified German word frequency list.

[6] *look, loss, moon, guns, hook, cook, . . .*

[7] *mußte, Woche, müsse, wußte, müßte, Wüste, . . .*

[8] One of our test subjects switches between the two keyboards for typing in English and German, respectively. As fluent typing with one keyboard may outweigh the efficiency effect for an optimised language–specific keyboard, UKO–II can also be tailored to use only one keyboard layout for both languages; i.e. only the language models and vocabularies are exchanged.

guage models extracted from domain–specific corpora as outlined in the next section.

3 Domain–specific corpora

Let us first determine the meaning of “domain” we assume here. According to Rose and Haddock (1997), homogeneity is an important feature. In Lesher and Rinkus (2002), genres and topics are differentiated. Lehnert noted from (Soderland, 1997) characterises a specific domain as a corpus of texts together with a predefined set of concepts that are of interest in that domain (e.g., newspaper articles in which the target concept is management succession events: identifying persons moving into corporate management positions or moving out). This means that a domain in the terminology of Lehnert coincides with a topic in the Lesher and Rinkus classification. Additional properties for domain classification according to Lesher and Rinkus are style and formalness. As our experiments show varying behaviour for different genre and topic domains of uniform style, we investigate the conciseness of our domains in more detail. Basically, we define a domain as a set of documents on the same topic and in the same style.

In our evaluation, we use two English and two German text collections. Every fifth word is extracted and added to the test set (*cross–validation* by splitting the files into 80 percent as training material and the remaining 20 percent of texts for testing). In the following, we describe each of the four text corpora in more detail.

ACL and COLING. These two English text collections originate from the proceedings of the ACL-02 conference (ACL, 2002) and the COLING 2002 (COLING, 2002)[9]. The ACL and COLING collection consists in total of 455 papers in portable document format. Since not all papers could be converted to plain text successfully, the usable corpus finally comprises 354 texts in the domain of computational linguistics (208 for ACL and 146 for COLING). After the process of tokenization and filtering, the overall number of word tokens of the corpus is approximately 1.5 million (0.45 million tokens were filtered out). A more detailed overview on the sizes is shown in Table 4.

BIDOK. The third part of our evaluation material is a selection of German texts from the *BIDOK* library (BIDOK, 2003). BIDOK is an Internet library containing scientific articles about the integration of disabled people. These articles are publicly available and primarily include German texts about theoretical basics, discussions, analyses and advice concerning the integration of handicapped persons.

We have chosen the BIDOK library because of its large amount of thematically sorted texts that are easily available. The whole collection consists of 711 texts divided into 14 domains. Two domains could not be dealt with in the preprocessing step because of technical problems. We run our evaluation tests on each of the remaining twelve domains.

The texts selected from the BIDOK corpus have an overall size of 5.5 million words. 1.36 million tokens are filtered out in the preprocessing step described in Section 4.

School. The *School* corpus consists of homework of a German 10th–grade pupil in the subjects Biology, Physics and Chemistry in the school year 2001–2002. These texts contain expressions such as formulae which cannot be typed directly using the ambiguous keyboard and thus are disregarded for the experiments[10]. The School corpus texts have an overall size of 8,027 tokens after elimination of 400 tokens with punctuation marks, formulae and numbers.

Beside cross–validation for all four corpora, the school corpus is also evaluated by taking approximately the first 80 percent of those texts in the training set and the rest in the test set in a second experiment (we refer to this experiment by School (b), label S’ in Figure 3 whereas the cross–validation of school homework is referred to by School (a) and S, respectively).

4 Experimental results

In our test series, we investigate the usefulness of complementing the general model that we retrieve from the CELEX corpus by a domain–specific

[9] Both paper collections are available on CD–ROM.

[10] Studies on using LaTeX (Lamport, 1994) for typing formulae are left for future work.

model. For all text collections, some preprocessing steps are necessary.

We only consider words that the user is able to access directly through the ambiguous keyboard. The letters “A–Z”, the apostrophe and the hyphen occur on the keys of the English keyboard. The German keyboard additionally contains the umlauts “ÄÖÜ” and the character “ß”. All other characters are substituted (e.g. *é* by *e*) or removed in a special step along with punctuation marks and numbers. Additionally, we ignore the distinction between upper and lower case letters.[11]

A summary of the corpora sizes after removing non–word tokens is given in Table 4.

	ACL	COLING	BIDOK	School
Total	1,172,609	760,123	5,521,501	8,027
Discarded	274,547	176,168	1,359,167	440
Training	712,415	433,324	3,550,882	6,342
Test	185,647	150,631	611,452	1,245

Table 4: Corpora sizes (tokens) after removing non–words.

As described in Section 2, we use the evaluation metric of *precision of rank r*. We run four experiments on each of the corpora described in Section 3. Furthermore, we assume a cross–validation for all corpora. Additionally, the School corpus is divided sequentially in a 80:20 percent manner (cf. S’ in Figure 3).

(1) The first run is performed with the general language model derived from the CELEX word statistics.

(2) In the second experiment, the language model derives from the training texts in the domain–specific corpus currently under consideration.

(3) The third run exploits both the models in (1) and (2) for each word and takes the model resulting in the better rank for this word (*oracle*). The oracle serves as an ideal baseline to compare our language model in (4) to[12].

(4) Finally, in a fourth experiment, the two models are combined by interpolating with an experimentally derived optimal weight λ. λ in $[0, 1]$ is chosen optimally for any corpus by comparing the precision for fifty values with step length 0.02. We calculated the following λ values: 0.98 (ACL), 0.98 (COLING), 0.96 (BIDOK), 0.46 (School (a)) and 0.32 (School (b)).

Formally, we denote by $p_1(w|c)$ the conditional probability of disambiguating the ambiguously typed word (*code*) c by word w in the general language model. By $p_2(w|c)$ we denote the conditional probability of word w given code c in the domain–specific model. Accordingly, the investigated rank evaluations (cf. Sec. 2.2) for experiment (1) to (4) look as follows:

(1) $\mathrm{rank}(w, c, p_1)$

(2) $\mathrm{rank}(w, c, p_2)$

(3) $\min(\mathrm{rank}(w, c, p_1), \mathrm{rank}(w, c, p_2))$ and

(4) $\mathrm{rank}(w, c, p_3)$ where
$p_3(w|c) = \lambda\, p_2(w|c) + (1 - \lambda)\, p_1(w|c)$.

In Figure 3, we present the evaluation results for the precisions of rank 1 to 5 using evaluations (1) to (4) for the corpora ACL, COLING, BIDOK and School (a). Additionally, the fifth bar depicts School (b) with a sequential partition. Table 5 delineates the *out of vocabulary* (OOV) — i.e. the percentage of missing word tokens or types missing in a model combination — resulting from models (1), (2) and (4). We observe that for the larger corpora (cf. ACL, COLING and BIDOK), the second method using only the domain–specific model already outruns the general language model computed by the CELEX word frequencies. Unfortunately, this does not hold for the smaller corpus with school homework texts where the test set has been split chronologically at the end of the corpus (cf. fifth bar). Here the domain–specific model performs worse in comparison to the general model. We can compensate for this disadvantage by combining the general and the domain–specific model. How far we can get is illustrated

[11] In UKO–II, the vocabulary actually distinguishes between upper and lower case letters in the following manner. Only the more frequent spelling is stored in the lexicon and suggested accordingly (e.g., *Tasten* “keys” outstrips *tasten* “to fumble”). In order to change the initial character a command via the meta–function key switches the mode.

[12] Note that without knowing the intended word, this value can only be computed for a code in a labelled corpus but not for an ambiguously typed code.

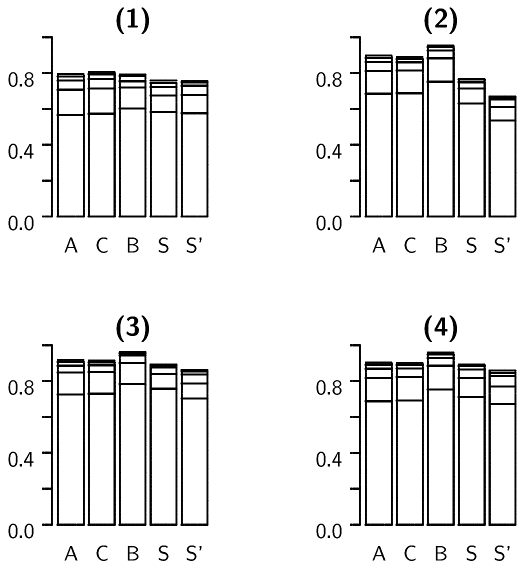


Figure 3: Cross–validation evaluation based on the precisions of rank 1 trough 5 for the corpora ACL (A), COLING (C), BIDOK (B) and School (a) (S) using the methods (1) to (4) described above. Additionally, School (b) (S') shows the evaluation results using a sequential test–training partition.

by the ideal baseline (3). It delineates the optimal value for any combination of the two language models in (1) and (2). The interpolated model performs only slightly poorer and is at least as good as the domain–specific model, even for the larger corpora.

Table 5 shows the percentage of unknown word types and tokens for each corpus using models (1), (2) and (4). The clear improvement of the OOV token rates for every corpus supports the desirability of combined general and domain–specific language models. There is no clear advantage for the general or the domain–specific model in terms of the OOV token rates, if we compare the first two rows (ACL, COLING) to the other corpora. Furthermore, we observe that at least with respect to the OOV type rates for the School corpora, we can reduce the percentage of unknown words in a test text considerably by memorising new words after their first occurrence. Then, the number of unknown words in a text correspond to the number of OOV types. This method can be exploited by a dynamic language model, as is planned in a later stage of our project. Interestingly, this does not hold for the ACL corpus and the combined model (4), where apparently many unknown words occur only once (the so–called *hapax legomena*; Manning and Schütze (2001, p. 22)) in the test set and thus do not benefit from dynamic updating the language model.

OOV rate %	(1)	(2)	(4)
types			
ACL	13.71	21.98	8.46
COLING	10.28	25.86	7.95
BIDOK	4.26	1.72	1.32
School (a)	3.38	1.96	1.40
School (b)	3.66	2.78	1.85
tokens			
ACL	17.85	23.36	8.66
COLING	17.11	32.61	11.16
BIDOK	19.03	2.25	1.63
School (a)	14.52	5.26	3.13
School (b)	13.17	6.25	4.10

Table 5: Out of Vocabulary (OOV) word rates (types and tokens) for the general (1), the domain–specific model (2) and model combination (4).

These results support the need for the development of a personalised version of UKO–II where the user may select one or possibly more topic domains that the system is trained with, before s/he starts writing.

5 Related work

Ambiguous keyboards in general and competing approaches are only touched lightly here for reasons of space (see also (Harbusch and Kühn, 2003)). An important recent development in the area of ambiguous keyboards is the work by Tanaka–Ishii *et al.* (2002). They present an ambiguous text input system with five or fewer letter keys. Word predictions are computed on the basis of *prediction by partial matching (PPM)* on the word level. The letters are assigned to the keys in alphabetical order. This approach compares favorably to ours. However, in our approach the keys have been assigned non–alphabetically after optimising with respect to a large corpus.

Other work on typing with word disambiguation focusses on the nine letter keys of a standard phone keyboard, e.g. Forcada (2001); Rau and Skiena (1996), and can be traced back to the early 1980s (Witten, 1982, pp. 120–122). Work in alternative and augmentative communication preceding Kushler (1998) deals with key–by–key disambiguation for efficient text input (Levine and Goodenough-Trepagnier, 1990; Arnott and Javed, 1992).

Many approaches have been published recently on the acquisition of homogeneous domains. (Seymore and Rosenfeld, 1997) investigate the problem of automatic topic identification and present a method for fine–tuned topic adaptation by interpolating with thousands of topic domain models. For cases where the topic domain is unknown or the topic changes within a document, it appears easier to apply dynamic adaptation of the domain models, e.g. by *on-line algorithms* (Kalai et al., 1999). Resnik and Melemed (1997) report on semi–automatic extraction of domain–specific translation lexica. In a post–editing process, generally used words are explicitly filtered out again (lexicon of technical terms only). We did not make such a distinction because we tried to circumvent a manual post–processing step as this requires judging whether a term that may occur in the general lexicon, is also used as a technical term (e.g., "feature" as in Head–Driven Phrase Structure Grammar; Sag and Wasow (1999)). Tests with a hand–coded domain–specific lexicon are left for future work.

Finally, we discuss the application of domain–specific knowledge in language modeling. This paradigm is widely used in various areas such as pattern recognition and image processing, text and language modeling, diagnosis systems, computational biology, or genetics (e.g., in the area of information retrieval domain–specific search engines are appearing on the market (cf. Kruger (2000)) or Oyama *et al.* (2001)). A directly comparable approach, in the domain of augmentative communication aids, is presented by Lesher and Rinkus (2002). They report substantial improvements based on domain–specific topic databases. Their claim is that the method also works in other domains of classification such as style, formalness, or genre. Our studies do not completely support with this claim. Particularly for smaller corpora or inhomogeneity within a corpus (cf. school corpus), we obtained lower values for a pure domain–specific disambiguation. For the interpolated language model, the claim is confirmed by our evaluations.

6 Conclusion

We investigated the use of domain–specific language models for the disambiguation of ambiguously typed text and presented two basic ways of combining domain–specific language models with a general model. We have run experiments with two English and two German text collections and compared the results. Although the domain–specific model alone does not outperform the general language model if the training set is not representative, a combination with the base model can compensate this. An ideal combination of the two models is the oracle (cf. (3)). In general, the interpolation of a domain–specific and a general model turns out to be a promising approach, even if the available domain–specific training corpora are of limited representativity.

An open question results from our observations in the previous Section. How can a domain–specific lexicon be acquired? Evaluations have to show the performance with a lexicon providing only technical terms interpolated with the general CELEX lexicon.

Future work will address the integration of an adaptive component updating a text–specific language model where recency information of words is considered for the ranking of word suggestions. We will explore a more fine–grained topic distinction which is available for our corpora. The use of syntax for disambiguation is examined in a parallel ongoing project. The long–term goal is a disambiguation component that utilises several layers of language models for text entry with an ambiguous keyboard. On an implementation level, our communication aid will be extended to allow the entry of formulae, numbers and other special expressions.

Acknowledgments

We would like to thank the reviewers for their valuable comments. All responsibility for errors rests with us, of course. Furthermore, we want to thank Gerard Kempen for helpful suggestions on the paper.

References

ACL. 2002. *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL '02)*, University of Pennsylvania, Philadelphia.

J.L. Arnott and M.Y. Javed. 1992. Probabilistic character disambiguation for reduced keyboards using small text samples. *AAC Augmentative and Alternative Communication*, 8(1):215–223.

R.H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. The CELEX lexical database (release 2.5). CD-ROM. Linguistic Data Consortium, University of Pennsylvania, PA.

BIDOK. 2003. BIDOK: Behindertenintegration – Dokumentation. Online library, http://bidok.uibk.ac.at/.

COLING. 2002. *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.

M.L. Forcada. 2001. Corpus-based stochastic finite-state predictive text entry for reduced keyboards: application to Catalan. *Procesamiento del Lenguaje Natural*, 27:65–70.

K. Harbusch and M. Kühn. 2003. Towards an adaptive communication aid with text input from ambiguous keyboards. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 03)*, Budapest, Hungary.

D. Jurafsky and J.H. Martin. 2000. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ.

A. Kalai, S. Chen, A. Blum, and R. Rosenfeld. 1999. On-line algorithms for combining language models. In *Proceedings of International Conference On Acoustics, Speech, and Signal Processing (ICASSP '99)*, Phoenix, AZ.

A. Kruger, C.L. Giles, F. Coetzee, E. Glover, G. Flake, S. Lawrence, and C. Omlin. 2000. DEADLINER: Building a new niche search engine. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2000)*, Washington, DC.

M. Kühn and J. Garbe. 2001. Predictive and highly ambiguous typing for a severely speech and motion impaired user. In C. Stephanidis, editor, *Universal Access in Human-Computer Interaction (UAHCI 2001)*. Lawrence Erlbaum, Mahwah, NJ.

C. Kushler. 1998. AAC using a reduced keyboard. In *Proceedings of the Technology and Persons with Disabilities conference 1998*. Online document, http://www.csun.edu/cod/conf/1998/proceedings/csun98_140.htm.

L. Lamport. 1994. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, MA, second edition.

G.W. Lesher and G.J. Rinkus. 2002. Domain–specific word prediction for augmentative communications. In *Proceedings of the RESNA 2002 Annual Conference*, Reno, NV.

S.H. Levine and C. Goodenough-Trepagnier. 1990. Customised text entry devices for motor-impaired users. *Applied Ergonomics*, 21(1):55–62.

C.D. Manning and H. Schütze. 2001. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, fourth edition.

S. Oyama, T. Kobubo, and T. Ishida. 2001. Keyword spices: A new method for building domain-specific web search engines. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1457–1466, Seattle, WA.

J. Raskin. 2000. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley, Reading, MA.

H. Rau and S. Skiena. 1996. Dialing for documents: An experiment in information theory. *Journal of Visual Languages and Computing*, 7:79–95.

P. Resnik and I D. Melamed. 1997. Semi–automatic acquisition of domain–specific translation lexions. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, DC.

T. Rose and N. Haddock. 1997. The effects of corpus size and homogeneity on language model quality. Technical Report HPL-97-70, HP Laboratories.

I.A. Sag and T. Wasow. 1999. *Syntactic theory: a formal introduction*. CSLI Publications, Stanford, CA, USA.

K. Seymore and R. Rosenfeld. 1997. Using story topics for language model adaptation. In *Proceedings of Eurospeech '97*, Rhodes, Greece.

S.G. Soderland. 1997. *Learning Text Analysis Rules for Domain–Specific Natural Language Processing*. Ph.D. thesis, University of Massachusetts, Amherst, MA.

K. Tanaka-Ishii, Y. Inutsuka, and M. Takeichi. 2002. Entering text with a four-button device. In *(COLING, 2002)*, pages 988–994.

I.H. Witten. 1982. *Principles of Computer Speech*. Academic Press, London, UK.