

# Sag' mir, was ich schreiben will!

## Beschleunigung der Texteingabe durch Textvorhersage

Wolfgang L. Zagler

### Kommunikationsbarrieren

Die auf geschriebene und gesprochene Sprache aufbauende zwischenmenschliche Kommunikation setzt sowohl sensorische (Sehen, Hören) wie auch aktuatorische (Sprechen, Schreiben) und intellektuelle Fähigkeiten voraus. Von besonders schmerzlicher Einschränkung ihrer Kommunikation sind jene behinderten Menschen betroffen, die zwar über die üblichen sensorischen und intellektuellen Fähigkeiten verfügen, jedoch unter Einschränkungen bzw. Ausfällen im aktuatorischen Bereich leiden. Mit anderen Worten: Gesprochenes und Geschriebenes kann ohne jegliche Einschränkungen aufgenommen und auch verstanden werden. Sobald aber (was gar nicht so selten vorkommt) eine motorische Störung gemeinsam mit dem Verlust der eigenen Stimme auftritt, verliert die betreffende Person faktisch jede Möglichkeit, sich selbst zu äußern, da sowohl das Sprechen wie auch das Schreiben im herkömmlichen Sinn unmöglich ist.

Da die übliche Bedienung einer Tastatur zur Texteingabe nicht angewendet werden kann, müssen alternative Techniken eingesetzt werden. Wenn die Bewegungsfähigkeit im Kopfbereich noch ausreicht, kann eine Tastatur mittels eines am Kopf befestigten oder mit den Zähnen gehaltenen Stabes betätigt werden. Um das seitliche Abrutschen zu verhindern werden Tastaturen üblicherweise mit einer Lochmaske versehen.

Ist auch diese Bewegungsfähigkeit nicht mehr gegeben, können vielfältige Einzelschalter (wie z.B. Blasschalter oder Lid-schlagschalters) eingesetzt werden, um minimale Bewegungen zur Steuerung des Kommunikationsvorganges auszunützen. Die Auswahl der einzelnen Kommunikationsbausteine (Buchstaben, Silben, Phrasen) erfolgt durch automatisches Scannen, also dadurch, dass sie zeitlich nacheinander zur Auswahl angeboten werden. Durch einen ersten Tastendruck wird der Scanvorgang angestoßen, und ein Cursor wandert schrittweise von einem Menüelement zum nächsten. Ist das gewünschte Menüelement erreicht, so kann dieses durch einen zweiten Tastendruck angesprochen werden. Die Taktfrequenz, mit der der Cursor (oft auch Fokus genannt) automatisch von einem Menüelement zum nächsten weitergeschaltet wird, wird Scanfrequenz genannt, ihr Kehrwert, die Verweildauer auf einem Me-

nüelement wird als Scanperiode bezeichnet.

Die richtige Wahl der Scanperiode stellt einen kritischen Punkt dar. Ist die Scanperiode zu groß, so kostet der Scanvorgang unnötig Zeit. Ist die Scanperiode zu klein, so ist das System nur mit größter Konzentration bedienbar oder wird sogar unbeherrschbar, wenn die Reaktionszeit des Benutzers nicht ausreicht.



**Bild 1:** Kommunikation mittels einer Buchstabentafel

**Bild 1** zeigt eine mittels Scanning benutzbare Buchstabentafel. Diese wird auf dem Bildschirm eines Computers dargestellt und erlaubt motorisch behinderten Personen die Kommunikation über einen Einfach-Sensor wie z. B. einen über den Lid-schlag ausgelösten Schalter. Die Buchstabentafel enthält 32 Elemente. Bei linearem Scannen werden im Durchschnitt 16,5 Scanschritte zur Auswahl eines Elementes benötigt. Bei einer Scanperiode von 1s sind das 16,5 Sekunden. Für ein Wort bestehend aus sechs Buchstaben werden demnach ca. 100 Sekunden benötigt! Mit anderen Worten: Die Schreibgeschwindigkeit liegt bei etwa 3 Buchstaben / min (die Eingabe des Leerzeichens nach dem Wort muss auch berücksichtigt werden).

Wie sehr die Kommunikationsraten behinderter Menschen hinter den uns vertrauten Geschwindigkeiten zurückbleiben, verdeutlicht nachfolgende Tabelle:

Gesprochene Sprache	800 bis 900 Buchstaben pro Minute
Nichtbehinderte Schreiber über Tastatur	200 bis 300 Buchstaben pro Minute
Tastatureingabe über Mundstab	75 bis 120 Buchstaben pro Minute
Eingabe über Einzelschalter und Scannen	3 bis 10 Buchstaben pro Minute

## 2 Redundanz der Sprache und Vorhersage

Ein möglicher Ausweg aus dem Kommunikationsdilemma ist die Tatsache, dass jede Sprache ein hohes Maß an Redundanz aufweist (in der Größenordnung von 75%). Zur Vermittlung einer sprachlichen Information ist es im Prinzip nicht erforderlich, alle Buchstaben eines Textes komplett einzugeben. Diese Aussage lässt sich sehr leicht auch experimentell verifizieren. Man bittet dazu eine Person, anstelle zu sprechen, in langsamer Folge auf Buchstaben einer Alphabettafel zu zeigen. Dabei wird folgendes vereinbart: Der Empfänger der Nachricht darf versuchen, aus den bereits angezeigten Buchstaben eines Wortes bzw. auch aus den bereits bekannten Wörtern des Satzes zu raten, wie der Text weitergeht. Ist die Annahme richtig, dann zeigt der „Sprecher“ auf das Leerzeichen und beginnt mit dem Buchstabieren den nächsten Wortes, ist die Annahme falsch, wird das Buchstabieren bis zum nächsten Raten fortgesetzt.

Wertet man eine derart geführte Kommunikation aus, wird man feststellen, dass weit weniger als die Hälfte der Buchstaben eines Textes gezeigt werden mußten. Der Rest war aus der Zusammenhang zu erraten.

## 3 Wenn der Computer das Raten übernimmt

Man hat daher nach Möglichkeiten gesucht, dieses Erraten von Texten aus dem Zusammenhang auch auf den Computer zu übertragen, um auf diese Weise die für die Erstellung eines Textes erforderliche Anzahl von Anschlägen zu reduzieren. Unter der englischen Sammelbezeichnung „*Predictive Typing*“ kommen dabei folgende Verfahren zur Anwendung:

### 3.1 Word Completion

Unter dem Terminus technicus „*Word Completion*“ wird folgendes Verfahren verstanden: Der Benutzer / die Benutzerin beginnt das gewünschte Wort - wie gewöhnlich - mit dem ersten Buchstaben zu tippen. Nach Anschlag des ersten Buchstaben werden vom *Word Completion*-Algorithmus unter Zugriff auf ein Wörterbuch, das Informationen über die Verwendungshäufigkeit enthält, Vorschläge für die Fortsetzung des Wortes gemacht. Ist das gewünschte Wort bereits in der präsentierten Vorschlagsliste enthalten, so kann es (z.B. durch Betätigen einer Funktionstaste) direkt aus der Liste ausgewählt werden und es ist vollständig eingegeben. Befindet es sich nicht unter den vorgeschlagenen Wörtern, so wird vom Benutzer / der Benutzerin der nächste Buchstabe getippt und vom Algorithmus wird eine neue, enger fokussierte Vorschlagsliste erstellt usw.

### 3.2 Word Prediction

„*Word Prediction*“ bezeichnet folgendes Verfahren: Nach vollständiger Eingabe eines Wortes, wird vom *Word Prediction*-Algorithmus eine Vorschlagsliste für das folgende Wort erstellt. Ist das gewünschte Wort bereits in dieser Liste enthalten, so kann der Benutzer dieses direkt auswählen und es ist fertig eingegeben. Anderenfalls beginnt der Benutzer mit dem Tippen des ersten Buchstabens, die weitere *Prediction* erfolgt nach dem *Word Completion* Verfahren.

### 3.3 Multiple Word Prediction

Durch „*Multiple Word Prediction*“ werden zwei oder mehrere aufeinanderfolgende Wörter auf einmal vorhergesagt. Funktionsweise ähnlich wie *Word Prediction*.

### 3.4 Abbreviation Expansion

„*Abbreviation Expansion*“ bedeutet die automatische Übersetzung von Abkürzungen in Volltext und ermöglicht die schnelle Eingabe von häufig benötigten Wörtern oder Phrasen.

## 4 Die Besonderheiten der deutschen Sprache

Wie bereits durch die bewußte Verwendung der englischen Fachausdrücke angedeutet, wurden Methoden des „*Predictive Typings*“ vor allem in englischsprachigen Ländern entwickelt, wo Textvorhersageprogramme Einsparungen an Anschlägen (Keystroke-Saving-Rate = KSR) von 50% bis 55% erreichen. Diese Programme beruhen im großen und ganzen auf Vorhersage und Ergänzung von Wörtern. Diese Erfolge sind nicht zuletzt auf die Tatsache zurückzuführen, dass Englisch zu den „nicht-flektierten“ Sprachen zählt, dass also englische Wörter lediglich geringe Va-

riationen in der Endung aufweisen. Somit können durch das Textvorhersageprogramm die einzelnen Wörter samt ihrer Endung aus einer kleinen Auswahl treffsicher vorhergesagt werden.

Als Beispiel soll hier die Erzeugung des Wortes „*investigates*“ (in „*He investigates the benefits of ...*“) dienen. Die grundsätzlich möglichen Endungen von *to investigate* beschränken sich auf folgende vier:

<i>investigate</i>	<i>investigates</i>
<i>investigating</i>	<i>investigated</i>

Betrachten wir hingegen das deutsche Wort „*forschen*“. Die Möglichkeiten der Endungsbildung sind mit 16 unterschiedlichen Wortformen bedeutend vielfältiger:

<i>forschen</i>	<i>forsche</i>	<i>forschst</i>
<i>forscht</i>	<i>forschte</i>	<i>forschtest</i>
<i>forschten</i>	<i>forschtet</i>	<i>forschend</i>
<i>forschende</i>	<i>forschender</i>	<i>forschendes</i>
<i>forschendem</i>	<i>forschenden</i>	<i>Forschet!</i> (Imperativ)

Wird keine Unterscheidung zwischen Groß- und Kleinschreibung getroffen, fallen alle Formen von „*Forschung*“ ebenso darunter wie das Eigenschaftswort „*forsch*“ mit allen seinen Endungen. Da für das deutsche Wort wesentlich mehr Buchstaben eingegeben werden müssen, bis sich die Unterschiede in den Endungen auf die Bildung der Vorschlagsliste auswirken, ist der durch Textvorhersage erzielte Gewinn geringer als in der englischen Sprache.

## 5 Elemente von Algorithmen zur Textvorhersage

Im folgenden werden einige Grundbausteine von Algorithmen zur Textvorhersage erläutert. Diese Grundbausteine sind Ergebnis von verschiedenen Forschungsarbeiten und werden auch bereits in mehreren Produkten eingesetzt.

### 5.1 Berücksichtigung der Häufigkeit von Einzelwörtern

Dies ist die einfachste und naheliegendste Strategie. Sie wird von praktisch allen bekannten Algorithmen genutzt. Die Vorschlagsliste wird basierend auf einem Frequenzwörterbuch erstellt indem jene Wörter vorgeschlagen werden, die dem eingegebenen Anfangsmuster entsprechen und am häufigsten vorkommen. Diese Häufigkeit wird durch eine Zählvariable angegeben, die jedem Wort zugeordnet ist. Man spricht von „*Unigrammen*“.

### 5.2 Berücksichtigung von bedingten Wahrscheinlichkeiten

Es werden Bigramme eingeführt, die die bedingten Wahrscheinlichkeiten angeben, dass ein bestimmtes Wort auf ein bestimmtes anderes folgt. Dadurch kann einerseits neben *Word Completion* auch *Word Prediction* durchgeführt werden. Andererseits entsteht ein zusätzlicher Gewichtungsfaktor für *Word Completion*.

Wird ein Wörterbuch vollständig mit Bigrammen versehen, so kann man sich diese in einer Matrix angeordnet denken. Diese Matrix wird vom in x und in y Richtung linear aufgetragenen Wörterbuch aufgespannt. Jedes Matrixelement gibt die Wahrscheinlichkeit an, dass auf das x-Wort das entsprechende y-Wort folgt. Bei einem Wortschatz von  $N$  Wörtern gibt es deshalb  $N^2$  Bigramme. Um hier die technischen Grenzen nicht zu sprengen werden in der Praxis oft nur Bigramme für z. B. jeweils die 10 häufigsten Nachfolger eines Wortes gespeichert. Dann entstehen  $10 \times N$  Bigramme.

### 5.3 Berücksichtigung syntaktischer Zusammenhänge durch Wortklassifizierung und Verbundwahrscheinlichkeiten

Durch Wortklassifizierung nach syntaktischen Grundsätzen und entsprechende Kennzeichnung im Wörterbuch („*word tagging*“) werden syntaktische Informationen gespeichert.

Zusätzlich werden die Wahrscheinlichkeiten für das Auftreten bestimmter Sequenzen von syntaktischen *tags* gespeichert. Diese können in Form von Bigrammen oder Trigrammen abgespeichert werden. Da die Anzahl von verschiedenen *tags* begrenzt ist (z. B. unter 100), ist auch der Umfang derartiger Matrizen begrenzt und es stellt heute kein technisches Problem mehr dar, auch Trigramme zu berücksichtigen

Die syntaktische Verbundwahrscheinlichkeit stellt einen zusätzlichen Gewichtungsfaktor für die Aufnahme eines Wortes in die Vorschlagsliste dar, der sowohl bei *Word Prediction* als auch bei *Word Completion* eingesetzt werden kann.

Eine stark vereinfachte Anwendung dieser Methode ermöglicht das Gewinnen von Ausschließungskriterien für die Vorschlagsliste. So ist z. B. das unmittelbare Aufeinanderfolgen von zwei Hauptwörtern in der deutschen Sprache sehr unwahrscheinlich. Eine einfache Strategie lautet deshalb, unmittelbar nach einem Hauptwort weitere Hauptwörter grundsätzlich aus der Vorschlagsliste auszuschließen. Damit kann die Wahrscheinlichkeit, dass sich das gewünschte Wort in der Vorschlagsliste befindet, erhöht werden. Etwas allgemeiner können durch Betrachtung aller syntaktischer Bigramme,

die verschwindend klein sind, Ausschlusskriterien gefunden werden.

**5.4 Berücksichtigung syntaktischer Zusammenhänge durch Grammatiken**

Diese Verfahren verwenden einen auf einer Grammatik basierenden Parser, um die wahrscheinlichsten Kategorien für nachfolgende Wörter zu bestimmen. Der Parser greift dabei auf syntaktische Kategorien und syntaktische Regeln (die „Grammatik“) zu, die für die jeweilige Sprache definiert sein müssen. Mit diesen Verfahren ist es möglich, Geschlecht und Einzahl / Mehrzahl zu berücksichtigen. So kann erreicht werden, Vorschläge mit den wahrscheinlichsten morphologischen Eigenschaften zu machen. Nichtsdestotrotz muss die Möglichkeit für den Benutzer / die Benutzerin gegeben sein, die Wortendung zu verändern.

Die Anwendung eines Parsers setzt ein Wörterbuch voraus, das morphologische Informationen enthält. Die Komplexität dieser Methode ist größer als bei den bisher skizzierten Verfahren, da alle Wörter eines Satzes, die vor dem gerade bearbeiteten Wort stehen bei der Vorhersage berücksichtigt werden.

**5.5 Berücksichtigung semantischer Zusammenhänge**

Die Berücksichtigung semantischer Zusammenhänge ist nicht sehr verbreitet, da die Implementierung sehr komplex ist. Auch hier können Parser eingesetzt werden. Jedes Wort hat assoziierte semantische Kategorien. Das Hauptproblem liegt in der Definition und Zuordnung von semantischen Kategorien.

**6 Wem nützt „Predictive Typing“?**

Wir wollen jetzt berechnen, ab welcher KSR sich die Verwendung von Textvorhersage für eine behinderte Person bezüglich der Schreibgeschwindigkeit bezahlt macht.

**Definitionen**

*k* sei die Schreibgeschwindigkeit einer Person gemessen in produzierte Anschläge/s

*B* sei die Anzahl von Buchstaben in einem Text

*W* sei die Zahl der Wörter, üblicherweise gleich Zahl der Leerzeichen plus Zahl der Absätze (wenn man davon ausgeht, dass nach jedem Wort entweder ein Leerzeichen oder ein <CR> folgt und wenn man vernachlässigt, dass nach dem letzten Wort eines Textes kein Leerzeichen und kein <CR> steht und Satz- und Sonderzeichen nicht berücksichtigt)

*A* sei die Zahl der Anschläge, die erforderlich ist um diesen Text auf einer Tastatur zu erzeugen, wobei  $A = B + W$  ist

Damit ergibt sich die Zeit, die zur herkömmlichen Erstellung eines Textes erforderlich ist zu

$$T_k = \frac{B+W}{k} = \frac{A}{k} \quad [s]$$

wobei  $T_k$  die Gesamtzeit für konventionelle Texteingabe bezeichnet.

Wird mit Textvorhersage gearbeitet, kommt es zu einer Einsparung an Anschlägen.

Anstelle von *A* Anschlägen sind nur mehr *P* Anschläge erforderlich. Diese *P* Anschläge setzen sich aus den Anschlägen für die Eingabe von Buchstaben und den Anschlägen zur Selektion aus der Vorschlagsliste zusammen.

Den Einsparungsfaktor wollen wir mit *e* bezeichnen. Er errechnet sich aus:

$$e = \frac{\text{gesparte Anschläge}}{\text{Zahl der Anschläge}}$$

$$e = \frac{A - P}{A}$$

Es ist üblich diesen Wert als „Keystroke Saving Rate“ (KSR) zu bezeichnen und in [%] anzugeben:

$$KSR[\%] = 100 \cdot e$$

Die Einsparung an Anschlägen führt zunächst zu einem Zeitgewinn. Dieser wird jedoch dadurch verringert, dass für den Blickwechsel von der Tastatur zur Bildschirm und zurück sowie für das Durchsuchen der Vorschlagsliste zusätzliche Zeit benötigt wird, die es beim Schreiben ohne Textvorhersage nicht gibt. Wir bezeichnen diese zusätzliche Zeit, die vor jedem einzelnen Anschlag für das Treffen der richtigen Entscheidung (weilerschreiben oder aus der Liste auswählen) aufgewendet werden muss mit *t* [s].

Damit ergibt sich für die Produktion eines Textes der Länge *A* folgender Gesamtaufwand den wir mit  $T_p$ , Gesamtzeit für Texteingabe mit *Prediction* bezeichnen wollen.

Die Gesamtzeit ist also die Summe aus der durch Vorhersage um den Faktor  $1 - e$  reduzierten Anschläge dividiert durch die Schreibgeschwindigkeit und der Zeit, die bei jedem Anschlag für das Treffen der Auswahl erforderlich ist.

Durch Gleichsetzen der Gesamtzeiten für konventionelle und *predictive* Eingabe läßt sich nun ein Grenz-Einsparungsfaktor  $e_g$  berechnen. Liegt unter der Vorgabe bestimmter Werte von *k* und *t* der vom

Textvorhersageprogramm geleistete Einsparungsfaktor unterhalb von  $e_g$ , dann benötigt die *predictive* Eingabe mehr Zeit als die konventionelle und bringt daher keinen Zeitgewinn.

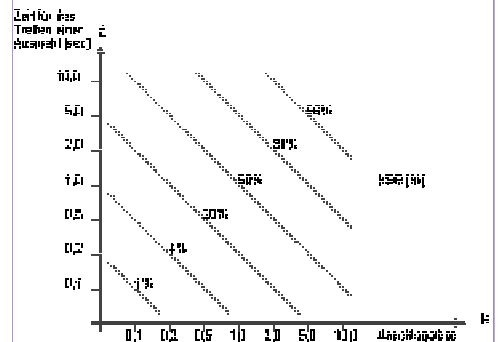
$$T_k = T_p$$

$$\frac{A}{k} = \frac{(1 - e_g) \cdot A}{k} + (1 - e_g) \cdot A \cdot t$$

und daraus:

$$e_g = \frac{kt}{kt + 1}$$

Dieser Zusammenhang kann als Diagramm dargestellt werden. Auf der x-Achse ist die Schreibgeschwindigkeit *k* aufgetragen, auf der y-Achse die Zeit *t*, die für das Treffen der Auswahl erforderlich ist. Die Parameterschar gibt die in [%] ausgedrückte Grenz-KSR an.



**Bild 2: Rentabilität von Textvorhersage**

Beispiel: Eine behinderte Person ist in der Lage, alle zwei Sekunden einen Anschlag zu produzieren. Die Zeit, die sie benötigt, die Auswahl aus der Liste zu treffen sei 1,5 Sekunden. Der Schnittpunkt liegt im Diagramm etwas unterhalb der Linie für KSR 50% (genau bei 43%). Das heißt, jede Textvorhersage, die in der Lage ist, eine Einsparung an Anschlägen 43% zu bewirken, ist für diese Person ein Vorteil.

Wir haben die Rentabilität von Textvorhersage bisher nur an der Möglichkeit, die Schreibgeschwindigkeit zu steigern, gemessen. Zwei Dinge dürfen dabei aber nicht vergessen werden:

1. Auch wenn kein Zeitgewinn erzielt werden kann, ist die Eingabe mit Textvorhersage oft bequemer und belastet den Schreiber / die Schreiberin weniger als die konventionelle Eingabe.
2. Durch die Textvorhersage werden dem Benutzer / der Benutzerin ganze Wörter vorgeschlagen. Für Personen, die aus welchen Gründen auch immer, Probleme mit der Rechtschreibung haben, ist das ein großer Vorteil. Die Anfangsbuchstaben eines Wortes werden oft leichter gemerkt und das Auswählen eines ganzen Wortes aus einer

Liste von Vorschlägen fällt leichter als die buchstabenweise Generierung eines Textes. In gewisser Weise findet eine automatische Rechtschreibprüfung des Textes statt. Es wird berichtet, dass behinderte Personen mittels Textvorhersage qualitativ bessere Texte herstellen konnten, als bei herkömmlicher Eingabe.

## 7 Berechnung der maximal möglichen KSR

Zur Abschätzung der erzielbaren KSR führen wir als weitere Größe die durchschnittliche Wortlänge  $l_{ow}$  ein. Wenn jedes Wort sofort – das heißt noch bevor der erste Buchstabe eingegeben wurde – in der Vorschlagsliste aufscheint, kann es durch einen einzigen Tastendruck pro Wort ausgewählt werden. Dieses Idealmaß wird oft als (ideale) Benchmark für Textvorhersagesysteme angesehen. Es soll hier mit  $e_1$  bezeichnet werden (Index  $I$  für einen Anschlag / eine Selektion).

$$e_1 = 1 - \frac{1}{l_{ow} + 1}$$

Der Term  $(l_{ow} + 1)$  ist die durchschnittliche Wortlänge plus nachfolgendes Leerzeichen, das ja bei der Textvorhersage bei jedem ausgewählten Wort automatisch generiert wird.

Mit einer für die deutsche Sprache üblichen  $l_{ow}$  von 5 ergibt sich der Benchmarkwert zu:

$$e_1 = 1 - \frac{1}{5 + 1} = 0,83$$

bzw. die maximal theoretisch erreichbare KSR mit 83%.

Nach eigenen Abschätzungen sind aber 2-3 Anschläge der realistische Wert, womit sich die Leistung eines realen Textvorhersageprogramms zwischen den Werten für die  $e_2$  und  $e_3$  einstellen wird:

$$e_2 = 1 - \frac{2}{5 + 1} = 0,67$$

$$e_3 = 1 - \frac{3}{5 + 1} = 0,50$$

Mit anderen Worten werden reale Textvorhersageprogramme eine KSR zwischen 55% und 60% kaum überschreiten können. Mit diesen Werten ist jedoch das Fernziel unter Einbeziehung syntaktischer Zusammenhänge gesteckt. Eigene Messungen, die noch ohne Verwendung syntaktischer Methoden erfolgten, zeigten, dass durchschnittlich mit einem Anschlag mehr gerechnet werden muss. Die zu erwartende Leistung wird demnach zwischen:

$$e_3 = 1 - \frac{3}{5 + 1} = 0,50$$

$$e_4 = 1 - \frac{4}{5 + 1} = 0,33$$

also die KSR 33% bis 50% liegen.

## 8 Schlussfolgerung und Herausforderung

Die hier durchgeführte Abschätzung einer maximal möglichen bzw. realistisch zu erwartenden KSR und jenes Grenz-Einsparungsfaktors, ab dem kein Zeitgewinn mehr vorhanden ist, macht deutlich, dass für wirklich nützliche Textvorhersageprogramme alle Register gezogen werden müssen und dass neben einer effektive Programmierung der grundlegenden Algorithmen selbst großes Gewicht auf die Einbeziehung linguistischer Zusammenhänge und die Optimierung der Mensch-Maschine-Schnittstelle gelegt werden muss.

Um für die deutsche Sprache den Nachholbedarf geringer werden zu lassen, wurde unter Förderung der Oesterreichischen Nationalbank das Projekt PROPHET durchgeführt, in dem grundlegende Arbeiten zur Textvorhersage in der deutschen Sprache durchgeführt wurden. In Simulationen an verschiedenen Texten wurde der Einfluß zahlreicher Parameter (wie Größe und Verwaltung des Wörterbuches, Länge der Vorschlagsliste etc.) gemessen. Ohne grammatikalische oder syntaktische Beziehungen auszunützen, konnten KSR-Werte in der Größenordnung von 40 bis 45% erreicht werden. Bei Texten, deren Wortschatz durch das Wörterbuch vollständig oder zumindest fast vollständig abgedeckt wurde, stieg die KSR sogar bis weit über 50%.

Es konnte somit durch Simulation gezeigt werden, dass brauchbare KSR-Werte auch in der deutschen Sprache erreicht werden können, dass aber das Ziel, Werte um die 55% bis 60% zu erhalten, erst durch Auswertung syntaktischer Beziehungen und die zusätzliche Verwendung von „*Abbreviation Expansion*“ erreicht werden kann.

## 9 Danksagung und Kontakte

Das Projekt PROPHET – Deutschsprachiges Textvorhersageprogramm für kommunikationsbehinderte Menschen wurde vom Jubiläumsfond der OeNB unter der Nummer 6273 gefördert. Ab Oktober 1998 wird der Abschlussbericht vorliegen.

### fortec -

Forschungsgruppe für Rehabilitationstechnik  
Institut für Allgemeine Elektrotechnik und Elektronik  
Technische Universität Wien  
1040 Wien, Gußhausstraße 27/359-3B  
Tel: +43 (1) 504-1830,  
Fax: +43 (1) 504-1830 /12  
E-Mail: [zw@fortec.tuwien.ac.at](mailto:zw@fortec.tuwien.ac.at)  
<http://fortec.iaee.tuwien.ac.at/fortec>

# Filmcomputer

gesammelt von Teemu Schabl

Dinge, die Computer in Filmen können:

- 1 Textverarbeitungsprogramme zeigen niemals einen Cursor
- 2 Man muss niemals die Leertaste benutzen, um lange Sätze einzugeben.
- 3 Filmcharaktere machen niemals Tippfehler.
- 4 Alle Bildschirme zeigen 5cm-Buchstaben
- 5 High-Tech-Computer, wie die von der NASA, dem CIA oder solchen Regierungsinstitutionen, haben einfach zu bedienende grafische Oberflächen.
- 6 Diejenigen, die keine grafische Oberfläche haben, haben unglaublich gute textbasierte Eingabeaufforderungen, die englische Sätze korrekt verstehen und ausführen.
- 7 Kommandozeileninterpreter geben einem Zugriff zu jeglicher Information, die man benötigt, wenn man einfach "ACCESS THE SECRET FILES" auf der nächsten Tastatur eingibt.
- 8 Man kann ebenfalls einen Computer mit einem zerstörerischen Virus infizieren, indem man einfach "UPLOAD VIRUS" eingibt (->"Fortress").
- 9 Alle Computer sind verbunden. Man kann auf Informationen auf dem Desktop-Computer des Bösewichts zugreifen, selbst wenn dieser ausgeschaltet ist.
- 10 Leistungsstarke Computer piepen jedesmal, wenn man eine Taste drückt oder der Bildschirm sich ändert. Einige Computer verlangsamen sogar die Ausgabe auf dem Bildschirm, damit es nicht schneller geht als man lesen kann (wirklich hervorragende Computer emulieren dazu noch das Geräusch eines Nadeldruckers).
- 11 Alle Computerpanels arbeiten mit Tausenden Volt und haben explosive Geräte direkt unter ihrer Oberfläche. Fehlfunktionen werden von einem hellen Lichtblitz, einer Rauchwolke, einem Funkenschauer, und einer Explosion, die dich wegschleudert, angezeigt.
- 12 Leute, die etwas auf einem Computer eingeben, können ihn ohne Datenverlust sofort ausschalten.
- 13 Ein Hacker kann jederzeit in die gesicherten Computer der Welt einbrechen, indem sie das geheime Passwort mit zwei Versuchen erraten. *weiter auf Seite 43 >*