

# Plädoyer für eine konstruktivistische Maschinelle Übersetzung

Michael Carl

Institut für Angewandte Informationsforschung,  
Martin-Luther-Straße 14,  
66111 Saarbrücken, Germany,  
carl@iai.uni-sb.de

## Abstract

Dieser Beitrag beschäftigt sich mit dem Problem der Kompositionalität in der Maschinellen Übersetzung (MÜ). Anstatt eine endgültige Lösung zu suchen, wird das Problem relativiert, indem es in Bezug zu der Mächtigkeit von Dekompositions- bzw. Adaptionskomponenten der jeweiligen MÜ-Systeme gesetzt wird. Eine entsprechende formale Definition der Terminologie wird gegeben, und die hinreichenden und notwendigen Bedingungen für eine kompositionelle Übersetzung werden genannt. Einige MÜ-Systeme werden anhand dieser Terminologie klassifiziert. Das Papier schließt mit der begründeten These, daß Example Based Machine Translation (EBMT) ein geeignetes Paradigma darstellt, mittels dessen MÜ-Systeme gemäß externer Anforderungen (wie Abdeckung, Verlässlichkeit, Handhabbarkeit, Rechenzeit) optimiert werden können.

## 1 Einleitung

In der Maschinellen Übersetzung (MÜ) ist Kompositionalität eines der meist diskutierten und am wenigsten verstandenen Themen. Das Hauptproblem beruht dabei auf der Wahl der Übersetzungseinheiten: wann und wie soll ein quellsprachlicher Text in Einheiten zerlegt werden, von denen der zielsprachliche Text erzeugt werden kann.

Eine der Grundvoraussetzungen aller MÜ-Systeme ist, daß Einheiten der zu übersetzenden Quellsprache auf Einheiten der Zielsprache abgebildet werden können. Wort-für-Wort-Übersetzung erzeugt aber nur in Ausnahmefällen das erwünschte Resultat und ist somit keine angemessene Basis für die MÜ. Andererseits — so die Annahme — ist Sprache aber kompositionell; das Problem besteht also darin, wie quellsprachliche Texte in Einheiten zerteilt werden können, von denen dann ein akzeptabler zielsprachlicher Text erzeugt werden kann.

In diesem Beitrag wird versucht, sich dem Kompositionalitätsproblem mit den Begriffen ‘Dekomposition’ und ‘Adaption’ anzunähern. Der Ansatz kann konstruktivistisch genannt werden, weil hier weniger

eine Theorie vertreten wird, als vielmehr nach dem “wie” (d.h. der Konstruktion) der Übersetzung gefragt wird. In der MÜ ist Kompositionalität — so die Annahme — kein unabhängiges Phänomen, sondern wird in vielfältiger Weise von den Programmkomponenten und den Zielen des Programmeinsatzes gesteuert. Wie und unter welchen Umständen welche Art von Übersetzungseinheiten sinnvoll ist, kann somit letztlich nur in Bezug auf die Einsatzumgebung und die Zielsetzung beantwortet werden, in der ein MÜ-System Anwendung findet. In diesem Sinne soll hier ‘konstruktivistisch’ verstanden werden: die Struktur der Umwelt (z.B. die Art der Übersetzungseinheiten) wird durch die Erfahrungen und begrifflichen Einteilungen des Subjekts erzeugt (z.B. durch seine Anforderungen an ein MÜ-System), und nicht umgekehrt.

Um das Problem entsprechend dieser Sichtweise genauer zu beschreiben, nutze ich Terminologie aus dem Bereich des fallbasierten Schließens (FS), anhand derer dann typische MÜ-Systeme klassifiziert werden können.

Systeme, die auf fallbasiertem Schließen beruhen (FS-Systeme), sind Problemlösungssysteme, die auf bekanntes Wissen in Form von Fällen zurückgreifen. Sie können somit als eine Verallgemeinerung von MÜ-Systemen angesehen werden. FS-Systeme lösen neue Probleme durch die Adaption der Lösungen alter (und darum bekannter) Probleme, welche in einer Falldatenbank gespeichert sind. Richter (Ric95) nennt vier Wissenskontainer, die in jedem FS-System enthalten sind: 1. die Falldatenbank: sie enthält bekannte Probleme zusammen mit ihren Lösungen. 2. Ein Abstandsmaß: es erlaubt, für ein neues Problem ähnliche Kandidaten in der Falldatenbank zu suchen, 3. der Adaptionsmechanismus: er modifiziert (adaptiert) die Lösungen der gefundenen Kandidaten entsprechend den Erfordernissen des neuen Problems und 4. die Fallsprache: sie umfasst das Vokabular, in der die Fälle beschrieben sind. Nach Richter kann theoretisch jeder dieser Wissenskontainer das gesamte für die Problemlösung notwendige Wissen enthalten. Wie in diesem Beitrag gezeigt wird (und von Richter unterstrichen

The proposal will not now be implemented

Les propositions ne seront pas mises en application maintenant

(1)

The proposal                      will not now be implemented

Les propositions                      ne seront pas mises en application maintenant

(2)

The proposal    will not *now* be    implemented

Les propositions    ne seront pas    mises en application    maintenant

(3)

The    proposal            will    not    now    be    implemented

Les propositions    ne seront pas    mises en application    maintenant

(4)

wird), ist die bessere Möglichkeit allerdings, das Wissen so in die Kontainer zu verteilen, daß optimale Ergebnisse erzielt werden können. Zusätzlich zu den von Richter genannten Wissenskontainern führe ich den Begriff der Dekomposition (siehe auch (CC96)) ein, mit dessen Hilfe ein Problem in eine Menge von Komponenten zerteilt werden kann. Die Lösung der Teile unter Berücksichtigung ihrer Beziehungen untereinander führt dann zur Lösung des komplexen Problems.

## 2 Einige Beispiele

In diesem Abschnitt beleuchte ich einige Dekompositionsmöglichkeiten anhand des folgenden englisch/französischen Übersetzungsbeispiels<sup>1</sup>:

en: The proposal will not now be implemented

fr: Les propositions ne seront pas mises en application maintenant

Die entsprechend erforderliche Adaptionenkomplexität wird folgendermaßen klassifiziert:

Besteht der zu übersetzende Satz nur aus einer Komponente wie in Bild 1 (d.h. es findet keine Dekomposition statt), dann ist auch keine Adaption erforderlich. Dies ist ein typisches Verhalten von sogenannten Translation Memories (TM) (z.B. TRADOS, TRANSIT), die nur aus Falldatenbank und Abstandsmaß bestehen<sup>2</sup>.

Im Bild 2 wird der Satz in zwei Komponenten zerteilt. Während das englische Subjekt (die erste Komponente) im Singular steht, ist seine französische Übersetzung im Plural. In beiden Sprachen gibt es jedoch eine Kongruenz zwischen Subjekt und Prädikat, sodaß das (Hilfs-) Verb der zweiten Komponente die Merkmale (Numerus und Person) der ersten Komponente trägt. Der Adaptionenmechanismus muß deshalb entsprechend mächtig ausgestattet sein, um diese Kongruenz wiederherzustellen.

Dekomponiert (zerteilt) man den Satz entsprechend Bild 3 in die drei Komponenten /The proposal/, /will not now be/ und /implemented/, dann wird die Adaption sehr viel komplexer. Es muß nicht nur die Kongruenz zwischen der 1. und der 2. Komponente (und auf der französischen Seite auch der 3. Komponente) wiederhergestellt werden.

<sup>1</sup>Das Beispiel stammt aus dem Hansards-Korpus und wird in (BCDP<sup>+</sup>90) ausführlich diskutiert. Zum Zwecke der Illustration will ich im Weiteren davon ausgehen, daß die Übersetzung korrekt und wünschenswert ist.

<sup>2</sup>Einige TM bieten allerdings auch zusätzlich ein integriertes MÜ-System an (im Interaktiv oder im Batch Modus)

Ausserdem muß auch die diskontinuierliche 2. Komponente behandelt werden. Da *now* als Übersetzung von *maintenant* gelten kann, ist die 2. französische Komponente durch die zwischengerutschte 3. Komponente unterbrochen. Der Adaptionsmechanismus muß hier also zusätzlich in der Lage sein, eine kontinuierliche englische Komponente auf eine diskontinuierliche französische Komponente abzubilden und umgekehrt.

In Bild 4 entspricht fast jedes Wort einer Komponente. Die Falldatenbank ist hier atomar. Der Adaptionsmechanismus jedoch muß eine sehr hohe Komplexität aufweisen und etwa einen Parser oder ähnliche Werkzeuge umfassen. Viele MÜ-Systeme fallen in dieses Schema.

Bevor auf die einzelnen Systeme näher eingegangen wird, etabliere ich nun zunächst eine Terminologie, mit der dann im darauffolgenden Abschnitt die einzelnen Komponenten genauer erfaßt werden können.

### 3 Terminologie

Ich gebe nun eine formale Definition der Terminologie und zeige mathematisch, daß eine konzeptuelle Äquivalenz in den einzelnen Berechnungsschritten erforderlich und hinreichend ist, um einen Satz kompositionell zu übersetzen.

#### Abstandsmaß

Eine Falldatenbank *FDB* und ein Abstandsmaß *dist* klassifizieren nach Globig und Wess (GW95) eine Menge von Begriffen *C*. Das Abstandsmaß ist dabei dermaßen konzipiert, daß der Abstand zwischen Instantiierungen gleicher Klassen Null beträgt. Wenn der Abstand zwischen einem Problem *p* und einem Fall *f* gleich Null ist, dann handelt es sich um Instantiierungen gleicher Klassen:

$$dist(p, f) = 0 \Rightarrow C(p) \equiv C(f) \quad (5)$$

Während Globig und Wess zwei Eigenschaften des Abstandsmaßes einführen (es kann **informiert** oder **universal** sein), will ich für unsere Zwecke etwas andere Kriterien gebrauchen:

Ein Abstandsmaß *dist*<sub>1</sub> ist besser **informiert** als ein Abstandsmaß *dist*<sub>2</sub> wenn beide die gleiche Menge von Begriffen *C* klassifizieren und alle Falldatenbanken *FDB*<sub>*i*</sub>, *i* > 1 größer sind als *FDB*<sub>1</sub>:

$$\forall i, i > 1, \quad C : (FDB_1, dist_1), C : (FDB_i, dist_2) \\ \vee |FDB_1| < |FDB_i| \quad (6)$$

Je besser ein Abstandsmaß informiert ist, desto mehr Wissen speichert es über die Begriffsmenge *C*. Das informierte Abstandsmaß ermöglicht die minimale Falldatenbank, da für jeden Begriff nur eine Instantiierung gespeichert werden braucht:

$$\begin{array}{ll} \text{wenn} & C(p) \equiv C(f) \\ \text{dann} & dist(p, f) = 0 \\ \text{sonst} & dist(p, f) = 1 \end{array}$$

Das uninformierte Abstandsmaß impliziert die maximale Falldatenbank, da jeder Fall eine andere begriffliche Klasse beschreibt:

$$\begin{array}{ll} \text{wenn} & dist(p, f) = 0 \\ \text{dann} & C(p) \equiv C(f) \\ \text{sonst} & C(p) \neq C(f) \end{array}$$

#### Dekomposition

Ein Problem *p* ist **dekomponierbar** dann und nur dann, wenn es in eine Menge von Komponenten *k*<sub>1</sub>...*k*<sub>*n*</sub> zerteilt werden kann und die Schnittmenge der zu den Komponenten gehörigen Begriffe gleich dem Begriff des Problems *p* ist:

$$dcomp(p) = k_1 \dots k_n \Leftrightarrow C(p) \equiv \bigcap_{i=1}^n C(k_i) \quad (7)$$

Die **Granularität** einer Dekomposition *dcomp*<sub>1</sub> ist feiner als die Granularität einer Dekomposition *dcomp*<sub>2</sub> in bezug auf eine Menge *P* von Problemen *p*, wenn die Anzahl der Komponenten, die durch *dcomp*<sub>1</sub> erzeugt werden, größer ist als die Anzahl der Komponenten, die durch *dcomp*<sub>2</sub> erzeugt werden:

$$\forall p \in P : |dcomp_1(p)| > |dcomp_2(p)| \quad (8)$$

#### Falldatenbank

Eine Falldatenbank *FDB* **deckt** eine Menge *P* von Problemen *p* **ab** dann und nur dann, wenn für jedes Problem *p* ∈ *P* mindestens ein Fall *f* ∈ *FDB* existiert, und beide, *p* und *f*, Instantiierungen desselben Begriffs *C* sind:

$$cover(P, FDB) \Leftrightarrow \forall p \in P, \exists f \in FDB : (9) \\ C(p) \equiv C(f)$$

Ein Fall *f* ist **atomar**, wenn er nicht kompositionell ist. Eine Falldatenbank *FDB* ist **atomar**, wenn sie nur atomare Fälle enthält. Sonst ist sie kompositionell.

Atomarität der *FDB* und Informiertheit des Abstandsmaßes sind im Prinzip unabhängig; sie weisen in der Praxis jedoch eine enge Korrelation auf: je atomarer die *FDB* konzeptioniert ist, desto weniger informiert wird das Abstandsmaß ausfallen. Umgekehrt gilt, daß, je informierter das Abstandsmaß ist, desto kompositioneller kann die *FDB* sein. Dieser empirische Zusammenhang und mögliche Konsequenzen daraus werden im nächsten Abschnitt ausgeführt. In einer realistischen Anwendung wird — bei gleicher Abdeckung — die Anzahl der Begriffe, die mittels einer kompositionellen *FDB* klassifiziert werden, allerdings sehr viel größer sein, als die Anzahl der Begriffe, die mittels einer atomaren *FDB* klassifiziert werden, weil eine kompositionelle *FDB* ungleich mehr Fälle enthält, als eine atomare *FDB*. Die Atomarität der Falldatenbank und die Granularität der Dekomposition sind hingegen abhängige Größen. Je atomarer die *FDB* ist, desto feiner wird die Granularität der Dekomposition ausfallen.

Eine atomare *FDB* impliziert eine feinste Granularität. Enthält die *FDB* nur maximal kompositionelle Fälle, dann wird die Dekomposition sehr grob sein, d.h. im Extremfall wird keine Dekomposition stattfinden.

### Adaptation

Eine Menge von Fällen  $f_1 \dots f_n$  ist **adaptierbar** dann und nur dann, wenn aus ihnen eine Lösung  $l$  komponiert werden kann, wobei die Schnittmenge der zu den Begriffen gehörigen Fälle  $f$  gleich dem Begriff der Lösung  $l$  ist:

$$\text{adapt}(f_1 \dots f_n) = l \Leftrightarrow \bigcap_{i=1}^n C(f_i) \equiv C(l) \quad (10)$$

Ein Adaptionmechanismus  $\text{adapt}_1$  ist **komplexer** als ein Adaptionmechanismus  $\text{adapt}_2$  in bezug auf eine Menge  $P$  von Problemen  $p$ , wenn  $\text{adapt}_1$  feiner granuliert Dekompositionen der Probleme  $p \in P$  komponiert als  $\text{adapt}_2$ :

$$\begin{aligned} \forall p \in P & : C(\text{adapt}_1(\text{dcomp}_1(p))) & (11) \\ & \equiv C(\text{adapt}_2(\text{dcomp}_2(p))) \\ & \vee \text{dcomp}_1() \text{ feiner als } \text{dcomp}_2() \end{aligned}$$

Aus 8 und 11 folgt, daß, je atomarer eine *FDB* ist, desto komplexer der Adaptionmechanismus ist.

### Kompositionalität

Ein Problem  $p$  ist kompositionell, wenn 1) es dekomponierbar ist in eine Menge von Komponenten, 2) die Komponenten durch die *FDB* abgedeckt sind und 3) die gefundenen Lösungen adaptierbar sind. Entsprechend der Gleichung 12 erhalten wir so eine Kette konzeptioneller Äquivalenzen für alle Berechnungsschritte:

$$C(p) \equiv \bigcap_i C(k_i) \equiv \bigcap_i C(f_i) \equiv C(l) \quad (12)$$

Die erste Äquivalenz wird durch die Dekomposition erreicht; die zweite Äquivalenz deckt die *FDB* ab, und die dritte Äquivalenz beruht auf den Anforderungen an den Adaptionmechanismus.

Im Weiteren gehe ich nun davon aus, daß die *FDB* alle Komponenten  $k$  abdeckt (d.h. für alle  $k$  gibt es ein  $f$  sodaß  $C(k) \equiv C(f)$ ) Ich untersuche die o.g. MÜ-Paradigmen in Bezug auf die Granularität der *FDB* und der Informiertheit des Abstandsmaßes anhand der eingeführten Terminologie.

## 4 Diskussion

In der Forschung entwickelte oder auf dem Markt verfügbare MÜ-Systeme realisieren eine Fülle von Dekompositions- und Adaptionmöglichkeiten mit entsprechend granulierten *FDB*en und informierten

Abstandsmaßen. Die Beispiele aus dem vorhergehenden Abschnitt sollen nun näher betrachtet werden.

Traditionelle MÜ-Systeme zeichnen sich durch eine atomare *FDB* und ein uninformatives Abstandsmaß aus<sup>3</sup>. Jeder Eintrag in der *FDB* (d.h. Lexikon) wird als distinkter Begriff aufgefasst, und es werden nur nicht-kompositionelle Begriffe in die Falldatenbank aufgenommen. Entsprechend Bild 4 werden Sätze in eine feinste Granularität dekomponiert. Die eigentliche Übersetzung findet dann im 'Adaptionmechanismus' statt. Alle drei MÜ-Generationen (cf. (WK95)) fallen hierunter.

- Der *direkte* Ansatz bildet lexikalische Items der Quellsprache auf lexikalische Items der Zielsprache ab und versucht dann, den zielsprachlichen Text wiederherzustellen.
- Der *interlingua* Ansatz (z.B. (Dor93)) beansprucht, eine sprachunabhängige Bedeutungsrepräsentation zu berechnen, von welcher ausgehend ein zielsprachlicher Text wieder generiert werden kann.
- Der *transfer* Ansatz kann zwischen beiden angesiedelt werden (z.B. (Str96)): anstelle von sprachunabhängigen Abstraktionen (Bedeutungen) werden Abstraktionen der Quellsprache berechnet, die dann auf Abstraktionen der Zielsprache abgebildet werden, um daraus den zielsprachlichen Text zu generieren.

Traditionelle MÜ-Systeme machen in aller Regel keinen Gebrauch von längeren Komponenten, die den Übersetzungsprozeß stark vereinfachen könnten. Sie tragen so nur in begrenztem Maße dem Rechnung, was Computer am leichtesten können: Speichern und Wiederfinden.

Im Gegensatz zu traditionellen MÜ-Systemen stehen Translation Memories (TM) (z.B. TRADOS (Hey96), TRANSIT). TM führen typischerweise keine Dekomposition des Quellsatzes durch (siehe Bild 1). Ihre *FDB* enthält lange, kompositionelle Fälle, während ein gut informiertes Abstandsmaß eine Anzahl von Übersetzungen aus der *FDB* sucht. Aufgrund der Tatsache, daß hier keine Adaption erfolgt, werden die angebotenen Lösungen jedoch zunehmend unvollständig bzw. unrichtig, je schlechter der Quellsatz mit einem Fall in der *FDB* übereinstimmt. Ein großer Nachteil von TM ist das relativ starke Anwachsen der *FDB*, während die Abdeckung sehr beschränkt bleibt.

Example-Based Machine Translation (EBMT) kann zwischen diesen beiden Ansätzen angesiedelt wer-

<sup>3</sup>Dies ist natürlich eine grobe Vereinfachung: Präprozessoren, (Multi-Word-Erkennung, Shallow Parsing, usw.) sowie auch Unterspezifikation weichen diese Grenzen auf. Solche Mechanismen werden allerdings oftmals lediglich als pragmatische Notwendigkeit angesehen, und nicht als eigentliche Komponente traditioneller MÜ-Systeme.

den. In manchen Systemen, cf. (SN90), (CC96), wird eine linguistische Analyse verwandt, um syntaktische Konstituenten zu erfassen. Der Adaptionsmechanismus hat hier zur Aufgabe, Teile (Wörter, Sequenzen von Wörtern oder Konstituenten) in der zielsprachlichen Struktur zu ersetzen. Die linguistische Analyse dient der Kenntlichmachung der zu ersetzenden Teile und einer Kriterienbildung für die Angemessenheit der Substitution in der zielsprachlichen Struktur.

In dem Pangloss EBMT System (Bro96), (NBD94) werden Fälle aus der *FDB* gesucht, die den quellsprachlichen Satz (oder Teile davon) enthalten. Mit Hilfe eines Thesaurus und eines zweisprachigen Lexikons werden dann die potentiellen Übersetzungen des Satzes (oder dessen Teile) aus den gefundenen Fällen extrahiert. Die Adaption dieser Satzstücke wird dann von einem statistischen Sprachmodell außerhalb des eigentlichen Pangloss EBMT System durchgeführt.

Weil EBMT (so wie auch FS-Systeme) verschiedenartige Wissenskontainer zur Verfügung stellen, können verschiedenartige Wissensquellen (Fälle und linguistisches Wissen) integriert werden. EBMT bietet dadurch den Vorteil, effizient von den diversen Wissensquellen Gebrauch zu machen und die Kontainerinhalte in bezug auf ein Ziel zu optimieren. Mögliche Optimierungskriterien sollen nun abschließend angegeben werden:

Die **Abdeckung** der *FDB* wird tendentiell mit kürzeren Fällen und mit einem informierten Abstandsmaß zunehmen. Da die Abdeckung des Systems im Wesentlichen von der Abdeckung der *FDB* abhängt, haben atomare Fälle sowie ein informiertes Abstandsmaß also das stärkste Potential, die Abdeckung des Systems zu vergrößern.

Die **Verlässlichkeit** der Ergebnisse beruht einerseits auf der Verlässlichkeit des Adaptionsmechanismus. Je kürzer die Komponenten sind, desto komplexer wird der Adaptionsmechanismus sein, weil — wie oben gezeigt — die Adaption mit kürzeren Komponenten sehr viel schwieriger wird. Mit zunehmender Komplexität des Adaptionsmechanismus steigt aber auch die Wahrscheinlichkeit fehlerhafter Adaption und damit sinkt die Verlässlichkeit der Ergebnisse. Andererseits steigt die Wahrscheinlichkeit fehlerhafter Klassifizierung, je informierter das Abstandsmaß ist. Möglichst lange Komponenten (und damit kompositionelle Fälle in der *FDB*) und ein uninformiertes Abstandsmaß haben also das Potential die Verlässlichkeit der Ergebnisse zu steigern.

Die **Handhabbarkeit** des Systems kann beschrieben werden als notwendige Anzahl der Schritte, um das Verhalten des Systems zu ändern. Handhabbarkeit hängt damit im wesentlichen von der Menge verschiedener Wissensquellen ab, die geändert werden müssen, dem nötigen Vorwissen, um diese Änderungen vorzunehmen und dem Grad, zu dem diese Änderungen automatisch durchgeführt werden

können. Offensichtlich ist die Veränderung von Adaptionswissen sehr viel aufwendiger (benötigt mehr Hintergrundwissen und mehr komplexe Operationen) als die Veränderung (Erweiterung) der *FDB*. Einfache Handhabbarkeit heißt in diesem Zusammenhang also Erweiterung der *FDB*.

**Rechenzeit** wird in allen Schritten benötigt. Für eine begrenzte Domain mag es sinnvoll sein, ein Maximum an (kompositionellen) Fällen in der Falldatenbank zu speichern da Retrieval — im Gegensatz zu Adaption — eine billige Funktion ist. Für sehr große oder unbegrenzte Domänen wird das allerdings kaum ein gangbarer Weg sein. Hier muß ein Kompromiß gefunden werden, um das Verhältnis von Falllänge zu Adaptionskomplexität zu optimieren.

## 5 Schlussfolgerung

In diesem Beitrag habe ich versucht, das Problem der Kompositionalität in der maschinellen Übersetzung (MÜ) von einer fallbasierten Perspektive aus anzugehen. Das zentrale Anliegen ist es, die Kompositionalitätsfrage zu relativieren, indem einerseits die Beziehung Granularität der Falldatenbank/Informiertheit des Abstandsmaßes und die daraus resultierende Komplexität des Adaptionsmechanismus untersucht wird und andererseits in Beziehung zu systemexternen Anforderungen gesetzt wird.

Ich habe eine formale Definition der Terminologie gegeben und prototypische MÜ-Systeme (traditionelle Systeme, Translation Memories (TM) und Example-Based Machine Translation (EBMT)) in die Terminologie eingeordnet. Der Beitrag wird abgeschlossen mit der These, daß EBMT ein geeignetes Paradigma darstellt, um MÜ-Systeme nach diversen benutzer-spezifischen Kriterien zu optimieren.

## References

- Peter F. Brown, J. Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, F. Jelinek, Robert L. Mercer, and P.S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85, 1990.
- D. Ralf Brown. Example-Based Machine Translation in the Pangloss System. In *COLING-96*, 1996.
- Bróna Collins and Pádraig Cunningham. Adaptation guided retrieval in EBMT. In *EWCBR-96 Workshop*, Lausanne, 1996.
- Bonnie Jean Dorr. *Machine Translation: A View from the Lexicon*. MIT Press, Cambridge, Massachusetts. London, England, 1993.
- Christoph Globig and Stefan Wess. Learning in case-based classification algorithms. In *LNAI 1168*, Berlin, New York, 1995. Springer.

Matthias Heyn. Integrating machine translation into translation memory systems. In *European Association for Machine Translation - Workshop Proceedings*, pages 111–123, Geneva, 1996. ISSCO.

Sergei Nirenburg, S. Beale, and C. Domashnev. A full-text experiment in example-based Machine Translation. In *International Conference on New Methods in Language Processing (NeMLaP) 94*, pages 78–87, Manchester, September 1994.

Michael M. Richter. The Knowledge Contained in Similarity Measures. <http://wwwagr.informatik.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html>, 1995.

S. Sato and M. Nagao. Towards memory based translation. In *COLING-90*, 1990.

Oliver Streiter. *Linguistic Reference Manual of the CAT2 Machine Translation System, Version 0.2*. Martin-Luther-Straße 14, 66111 Saarbrücken, BRD, January 1996.

Peter Whitelock and Kieran Kilby. *Linguistic and computational techniques in Machine Translation system design*. Computational Linguistics. UCL Press, London, 1995.