



universität
wien

DIPLOMARBEIT

Titel der Diplomarbeit:

**Freie Translation Memory Systeme
für die Übersetzungspraxis:
Ein kritischer Vergleich**

Verfasser:

Sébastien Guillaudeau

angestrebter akademischer Grad:

Magister der Philosophie (Mag. phil.)

Wien, 2009

Matrikel-Nummer: 0203401
Studienkennzahl lt. Studienblatt: A 324 331 351
Studienrichtung lt. Studienblatt: Übersetzer Ausbildung Deutsch / Spanisch
Betreuer: Univ.-Prof. Dr. Gerhard Budin

Danksagung

Bedanken möchte ich mich bei Dr. Elina Lagoudaki für die Zusendung ihrer Doktorarbeit, deren Lektüre mich in meinen Recherchen orientiert hat. Für die freundliche Beantwortung all der zahlreichen Fragen, die ich in den vergangenen Monaten gestellt habe, danke ich den Entwicklern und Benutzern von OmegaT sowie Herren Oleg Tsygany und Dmitri Gabinski vom Anaphraseus-Entwicklerteam. Für seine konstruktive Kritik und das akribische Korrekturlesen des Manuskriptes bin ich Herrn Mag. Simon Böckle sehr zu Dank verpflichtet.

I would like to thank Elina Lagoudaki for sending me her PhD thesis which was a great help for my research. I am very grateful to the group of users and developers of OmegaT as well as to Oleg Tsygany and Dmitri Gabinski from the Anaphraseus development team for having kindly answering all my questions over the past months. Furthermore I owe many thanks to Simon Böckle for his constructive criticism and his meticulous review of the manuscript.

Inhaltsverzeichnis

Abbildungsverzeichnis	vi
Tabellenverzeichnis	vii
1 Einleitung	1
2 Begriffsklärungen	2
2.1 Translation Memory Systeme	2
2.1.1 Geschichtlicher Überblick	2
2.1.2 Kategorisierungsmodelle	6
2.1.3 Translator's workstation und TM-System	11
2.1.4 Definition von TM-Systemen	11
2.2 Open Source bzw. Freie Software	12
2.2.1 GNU/Linux	13
2.2.2 Die vier Freiheiten	13
2.2.3 Lizenzmodell: die GPL	14
2.2.4 Free Software Foundation	14
2.2.5 Open Source Initiative	15
2.2.6 Zusammenfassung	15
3 Untersuchungsvorbereitung	17
3.1 Gegenstand	17
3.1.1 OmegaT	17
3.1.2 Anaphraseus	18
3.1.3 Weitere Anwendungen	18
3.2 Vergleichskriterien	19
3.2.1 EAGLES-Framework	19
3.2.2 François Massion	19
3.2.3 Elina Lagoudaki	20
3.2.4 Zusammenfassung	21
4 Funktionale Merkmale: Vor dem Übersetzen	22
4.1 Projektmanagement	22
4.1.1 Projekt anlegen	22
4.1.2 Projektelemente ändern	24
4.1.3 Sprachen	25
4.1.4 Segmentierung	26

4.2	Unterstützte Dateiformate	29
4.2.1	Dokumentvorlagen	29
4.2.2	Translation Memory Dateiformate	32
4.2.3	Terminologiedatenbanken	34
4.2.4	Integrierte Wörterbücher	37
4.3	Vorübersetzen	38
4.4	Statistik	40
4.5	Alignment	43
5	Funktionale Merkmale: Während des Übersetzens	48
5.1	Editieren	48
5.1.1	Editortyp	48
5.1.2	Auswahl zwischen Absatz- und Satzsegmentierung	49
5.1.3	Segmente verkleinern/vergrößern	50
5.1.4	Kopieren/Einfügen	51
5.1.5	Einfügen von Sonderzeichen	51
5.1.6	Formatierung	52
5.1.7	Undo-Funktion	54
5.2	Anzeige	55
5.2.1	Formatierungen	55
5.2.2	Zieltextvorschau	55
5.2.3	Anzeige des Übersetzungsfortschritts	56
5.3	Terminologie verwenden	57
5.3.1	Nachschlagen – Einfügen in den Zieltext	57
5.3.2	Speichern von neuen Begriffspaaren	58
5.3.3	Listen von Stoppwörtern	59
5.4	Konkordanzsuche	59
5.5	Sonstige Produktivitätssteigerung	61
5.5.1	Automatische Propagation	61
5.5.2	Automatisches Ersetzen von Variablen	62
5.5.3	Rechtschreibprüfung	62
5.5.4	MT-Modul	64
5.6	Zusammenfassung	64
6	Funktionale Merkmale: Nach dem Übersetzen	66
6.1	Qualitätsprüfung	66
6.1.1	Rechtschreibprüfung	67
6.1.2	Terminologieprüfung	67
6.1.3	Tagschutz und -prüfung	67
6.1.4	Zahlen	68
6.1.5	Vollständigkeit der Übersetzung	68
6.2	Lieferung der Zieldateien	68
6.2.1	Einsprachige Dateiformate	69

6.2.2	Zweisprachige Dateiformate	70
6.3	Pflege des Datenbestands	71
6.4	Zusammenfassung	72
7	Nichtfunktionale Merkmale: Effizienz	73
7.1	Suchalgorithmus	73
7.1.1	Ähnlichkeitsmessung	73
7.1.2	Trefferquoten	74
7.1.3	Precision – Recall	77
7.1.4	Glossare	79
7.2	Analyse	80
7.3	Alignment	80
7.4	Zeit- und Ressourcenbeanspruchung	81
7.5	Formatierungen	82
8	Nichtfunktionale Merkmale: Benutzbarkeit	83
8.1	Anpassbarkeit	83
8.2	Erlernbarkeit	84
8.2.1	Dokumentation	84
8.2.2	Einfachheit	84
8.2.3	Support	85
8.3	Erweiterbarkeit	86
8.4	Übertragbarkeit	87
8.5	Zuverlässigkeit	87
9	Zusammenfassung	89
	Anhang	91
	Bibliographie	92
	Glossar	97
	Reguläre Ausdrücke	103
	Vergleichsmatrix	105
	Verzeichnis der erwähnten freien Anwendungen	108
	Lebenslauf	109
	Abstract	110

Abbildungsverzeichnis

2.1	Klassifizierung der Übersetzungstypen	8
2.2	Einordnung der TM-Systeme in den Übersetzungstechnologien	9
2.3	Kategorisierung von Softwares nach Lizenztypen	16
3.1	OmegaT-Übersetzungsumgebung	17
4.1	Projektanlegen in OmegaT	23
4.2	Projektfenster von Anaphraseus	24
4.3	Sprachenauswahlfenster in OmegaT	25
4.4	Segmentierungsregeln in OmegaT	27
4.5	Segmentierungsregeln in Ratel	28
4.6	Optionen bei der Erstellung eines Projektes in Rainbow	31
4.7	Hauptfenster von Olifant mit Dialogfenster zum Entfernen von TMX-Tags	34
4.8	Glossardatei in OmegaT	35
4.9	Dialogfenster zum Auswählen der Glossardatei in Anaphraseus	36
4.10	Statistische Analyse in OmegaT	41
4.11	Benutzeroberfläche und Skript von bligner	44
4.12	Hauptfenster von Bitext2tmx	46
5.1	Segment vergrößern in Anaphraseus	50
5.2	Zeichentabelle in Writer	52
5.3	Darstellung von Formatierungszeichen in OmegaT	53
5.4	Statistische Hinweise im OmegaT-Hauptfenster	56
5.5	Terminologische Treffer (Glossar und Wörterbuch) in OmegaT	57
5.6	Terminologischer Treffer (Glossar) in Anaphraseus	58
5.7	Konkordanzsuche in OmegaT	60
5.8	Einstellung des Editierverhaltens von OmegaT	61
5.9	Rechtschreibprüfung während der Übersetzung in OmegaT	63
6.1	Tagprüfung in OmegaT	68
6.2	Rückkonvertierung von Dateien ins ursprüngliche Format mit Okapi Rainbow	70
8.1	Anzahl der Nachrichten auf der OmegaT-Mailing Liste	86

Tabellenverzeichnis

2.1	Einordnung der TM-Systeme innerhalb der CAT	10
4.1	Analysefunktionen in OmegaT	42
5.1	Verfügbare Kennzahlen über den Übersetzungsfortschritt in OmegaT	56
5.2	Unterschiedliche Einheiten und Schreibweisen je nach Sprache und <i>locale</i> . . .	62
7.1	Vergleich der Trefferquoten bei OmegaT, Anaphraseus und Trados	76
7.2	Messung der <i>Precision</i> bei OmegaT und Anaphraseus in Bezug auf die TM .	78
7.3	Messung des <i>Recall</i> bei OmegaT und Anaphraseus in Bezug auf die Glossare	79
7.4	Analysevergleich OmegaT-Trados	80
7.5	Alignment-Ergebnisse TagAligner/Bligner	81

1 Einleitung

»Why should we be interested in using computers for translation at all?

The first and probably most important reason is that there is just too much that needs to be translated, and that human translators cannot cope. A second reason is that on the whole technical materials are too boring for human translators, they do not like translating them, and so they look for help from computers.«¹

John Hutchins

Ob privat oder gewerblich, Computer werden dort eingesetzt, wo man von ihnen eine Unterstützung bei der Bewältigung von bestimmten Aufgaben erhält. So haben auch Übersetzer² im Laufe der vergangenen zwei Jahrzehnte ihre Arbeitsweise durch den Einsatz von Computerprogrammen verändert. Eine erste Neuerung war die Verwendung von Textverarbeitungsprogrammen und von elektronischen Wörterbüchern. Später kam das Internet als reichhaltige Informationsquelle und als Beschleuniger des Informationsaustausches hinzu. Gleichzeitig wurden Programme speziell für die Unterstützung des Übersetzungsvorgangs entwickelt und vermarktet, zu denen auch die Translation Memory Systeme gehören.

Die translationswissenschaftliche Forschung hat Translation Memory Systeme nach unterschiedlichen Methoden untersucht und Ergebnisse geliefert, die zu deren Weiterentwicklung und Verbesserung beigetragen haben. Eine dieser Methoden ist die **empirische Analyse** dessen, wie sie von ihren Anwendern eingesetzt werden. Eine weitere Methode besteht in einer **vergleichenden Analyse** unterschiedlicher Translation Memory Systeme nach deren funktionalen und nichtfunktionalen Merkmalen. Dennoch schließen die bis dato veröffentlichten empirischen und vergleichenden Studien eine besondere Softwarekategorie von ihren Analysen aus: die sogenannten »freien« Translation Memory Systeme, bei denen es sich um Programme handelt, die unter besonderen Lizenzbestimmungen herausgegeben werden.

Die vorliegende Arbeit setzt sich daher zum Ziel, diese Lücke durch die gezielte **vergleichende Analyse** von freien Translation Memory Systemen ansatzweise zu schließen und wird versuchen, auf folgende drei Fragen Antworten zu geben: Was zeichnet freie Translation Memory Systeme aus? Inwiefern können sie von Übersetzern zur Unterstützung ihrer Arbeit eingesetzt werden? Welche Nutzen und Nachteile können sich aus deren Einsatz ergeben?

1 John Hutchins: »Current commercial machine translation systems and computer-based translation tools: system types and their uses«, S.1.

2 Im Sinne einer geschlechtsneutralen Formulierung beziehen sich die in dieser Arbeit verwendeten männlichen Formen sowohl auf Frauen als auch auf Männer.

2 Begriffsklärungen

2.1 Translation Memory Systeme

2.1.1 Geschichtlicher Überblick

Über Entstehungszeit und Urheberschaft der Idee von Translation Memory (TM) und Translation Memory Systemen (TMS) sind sich die Forscher uneinig. Da die gesamte Forschungsgeschichte hier nicht ausführlich nachverfolgt werden kann, weil es über das Thema dieser Arbeit hinausginge, können hier nur einige Etappen dieser Geschichte wiedergegeben werden. In einem 1998 veröffentlichten Artikel³ zeichnet John Hutchins die Entstehung und Fortentwicklung der Translation Memory nach und beschreibt, wie die TM-Technologien aus der Forschung über die Maschinelle Übersetzung (Englisch: *Machine Translation*, abgekürzt in MT) hervorgegangen sind.

2.1.1.1 ALPAC

Im Jahre 1966 setzt sich das *Automatic Language Processing Advisory Committee* (ALPAC) mit der Evaluierung damals existierender vollautomatisierter Übersetzungssysteme auseinander und hält in seinem Gutachten fest, dass die MT-Systeme erhebliche Qualitätsmängel aufweisen. Trotz – oder gerade wegen – dieses Rückschlages erkennt Hutchins hier der erste Impuls für die Entstehung der TM-Technologie:

»The consequences of the report were massive reductions in the support for MT in the United States. However, ALPAC was not entirely negative; it encouraged support for basic computational linguistics and, in addition (today often forgotten), the development of computer-based aids for translators.« (Hutchins 1998: 3)

Der hier verwendete Begriff von »*computer-based aids for translators*« soll bewusst auf die damals entstehende Diskrepanz zwischen (voll-)automatisierter Übersetzung einerseits und rechnerunterstützter Humanübersetzung andererseits hinweisen.

³ Dieses Kapitel basiert weitgehend auf John Hutchins (1998): »The origins of the translator's workstation«. In: *Machine Translation* 13(4), S. 287–307. URL: <http://www.hutchinsweb.me.uk/MTJ-1998.pdf> (besucht am 13.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsa4YUCc>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.

2.1.1.2 Friedrich Krollmann

Der für den damaligen Übersetzerdienst der deutschen Bundeswehr tätige Friedrich Krollmann entwickelte seit 1965 ein lexikografisches System namens LEXIS. 1971 erwähnt er wünschenswerte Erweiterungen terminologischer Datenbanken, die die Arbeit der Übersetzer erleichtern sollen. Unter diesen Empfehlungen finden sich mehrsprachige Wörterbücher und vor allem sogenannte »*translation archives*« (Krollmann 1971: 118 f.). Diese auf Magnetbändern gespeicherten Übersetzungsarchive sollten als Quelle für wiederverwendbare Übersetzungssegmente dienen und den Übersetzer in seiner Arbeit unterstützen. Des Weiteren plädiert er für eine Vernetzung aller Datenbankentypen im Sinne einer automatischen Abfrage, die einen Abgleich zwischen Quell- und ZIELtext ermöglichen soll. Bedingt durch den damaligen Stand der Datenverarbeitungs- und Kommunikationstechnik konnten diese Ideen jedoch erst später umgesetzt werden.

2.1.1.3 Erhard Lippmann

Anfang der 1970er Jahre waren Computer Anlagen mit großem Platzbedarf, auf denen Programme installiert waren, die man mittels Terminals bediente. Man musste Programme im Rechenzentrum installieren, die Anwendung laufen lassen und auf die Ergebnisse warten, was mitunter Tage dauern konnte. Um dies zu umgehen, hatte Krollmann die Nützlichkeit eines neuen computertechnologischen Ansatzes erkannt: des »Time-Sharing«, der den gleichzeitigen Zugriff mehrerer Benutzer auf Daten und deren Vernetzung ermöglichte. Alle Vorgänge sollten auf den Benutzerterminals erledigt werden können. Der im IBM-Forschungszentrum tätige Erhard Lippmann untersuchte als erster die Anwendung des Time-Sharing auf die rechnerunterstützte Humanübersetzung. In einem 1971 veröffentlichten Forschungsbericht hielt er diese neuen Techniken der Interaktion zwischen Mensch und Maschine fest, wobei er hier auch betont, dass, »*im Gegensatz zu MT, dieses System nicht versucht, die menschliche Übersetzung dadurch zu simulieren, dass es eine eigenständige Übersetzung liefert. Vielmehr dient es einer Erweiterung der Fähigkeiten des Benutzers.*« (Lippmann 1971: 10)⁴ So konnten zum Beispiel lexikalische Vorschläge direkt am Bildschirm angezeigt werden – im Gegensatz zum LEXIS-System, bei dem alle lexikalischen Vorschläge ausgedruckt werden mussten. Dennoch konnte dieser erste dezentrale Ansatz mangels technischer Voraussetzungen, insbesondere im Bereich der Textverarbeitung, vorerst nicht umgesetzt werden.

⁴ Alle Übersetzungen von Zitaten aus dem Englischen sind – wenn nicht anders vermerkt – vom Autor dieser Arbeit.

2.1.1.4 Peter Arthern

Die praktische Anwendung eines Übersetzungsarchivs wurde 1979 von Peter Arthern formuliert. Zu jener Zeit untersuchte die EG-Kommission das potentielle Nutzen computerbasierter Terminologiesysteme in Hinblick auf eine Verbesserung der Übersetzungsvorgänge. In seiner Veröffentlichung erklärte Arthern, dass die MT-Technologien (insbesondere Sys-tran) noch nicht reif waren, um auf alle von der Kommission produzierten Texte angewandt zu werden. Des Weiteren stellte er fest, dass diese Texte *»sich in hohem Maße wiederholten, ganze Stellen aus anderen EG-Texten zitierten«*, und dass Übersetzer *»Texte oft noch einmal zu übersetzen hatten, obwohl dies bereits geschehen war.«* (Arthern 1979: 94) Von dieser Feststellung ausgehend, empfiehlt er ein System, bei dem alle Original- und übersetzten Texte gespeichert sind, gekoppelt mit der Fähigkeit, beliebige Stellen rasch wiederzufinden und unmittelbar in neue Dokumente einzufügen. Sein Konzept, das er *»translation by text-retrieval«* nennt, soll imstande sein, den neuen Text mit im System gespeicherten Texten *»Satz für Satz«* zu vergleichen und nächstähnliche Sätze auszugeben. Je umfangreich das Übersetzungsarchiv wäre, umso geringer der Übersetzungsaufwand, bis hin zur einfachen Ersetzung von Namen und Datumangaben (vgl. ebd.: 95). In seinem Konzept sieht er außerdem die Möglichkeit vor, MT als zusätzlich unterstützende Module zu implementieren.

2.1.1.5 Martin Kay

Sowohl John Hutchins (1998) als auch Harold Somers (2003) sind sich darüber einig, dass die Geburtsstunde der später als Translation Memory bekannten Technologie in der Veröffentlichung des – ursprünglich wenig bekannten – Forschungsberichts von Martin Kay 1980 anzusiedeln ist. In diesem Bericht übt Kay Kritik am damals weit verbreiteten Ansatz der MT-Forschung, wonach diese menschliche Übersetzer über kurz oder lang gänzlich ersetzen müsste. Er plädiert hingegen für die Entwicklung von Übersetzungs-Tools, die tatsächlich von Übersetzern anwendbar seien (vgl. Hutchins 1998: 8). Kays Grundidee lag darin, bestehende Textverarbeitungsanwendungen um übersetzungsorientierte Funktionen zu ergänzen. Dieses von ihm genannte *»translator's amanuensis«* würde *»ständig unter der engen Kontrolle eines menschlichen Übersetzers stehen.«* Es wäre hier, um *»ihm bei der Steigerung seiner Produktivität zu helfen, und nicht um ihn zu ersetzen«* (Kay 1980: 20). Wesentliche Komponente dieses Systems wären (vgl. Hutchins 1998: 8):

- ein guter mehrsprachiger Texteditor
- ein Terminal mit geteiltem Bildschirm

- eine Funktion zum automatischen Nachschlagen eines beliebigen Wortes oder Ausdrucks in einem Wörterbuch
- eine Funktion zum Nachschlagen von vorher vom Übersetzer getroffenen Entscheidungen, um die Konsistenz der Übersetzung zu gewährleisten, und
- eine Funktion zum automatischen Übersetzen von Stellen, wobei der Übersetzer im Nachhinein oder während des Vorganges intervenieren kann.

Dabei gilt Kay nicht als der Urheber der TM, die ja auf Peter Arthern zurückgeht. Dennoch gab seine Synthese wichtige Impulse für die Entwicklung praxisbezogener, vom Übersetzer unmittelbar anwendbarer Tools, die in ihrer Gesamtheit betrachtet eine »*translator's workstation*« bilden.

2.1.1.6 Alan Melby

Anfang der 1980er Jahre empfiehlt Alan Melby – Hutchins zufolge als erster – die Verwendung der zweisprachigen Konkordanzsuche als Unterstützung für die Übersetzungsarbeit. Dabei werden Quell- und Zieltexte in Einheiten, sogenannte »*translation segments*« (Melby 1981) geteilt und gegenübergestellt. Zur Erstellung von Translation Memories wird dieser Vorgang später Alignment genannt. Der Vorteil der Konkordanzsuche liegt darin, dass der Übersetzer den Kontext jeder Okkurrenz angezeigt bekommt und somit entscheiden kann, ob das gesuchte Wort oder der gesuchte Ausdruck bereits übersetzt, und wenn ja, richtig übersetzt wurde.

1982 erarbeitet Alan Melby – parallel zu Kays Empfehlungen, die zu jener Zeit wenig bekannt sind – ein Modell für einen »Übersetzer-Arbeitsplatz«. Im Zuge seiner Forschung über MT an der Brigham Young Universität hatte er nämlich festgestellt, dass MT qualitativ nicht zufriedenstellende Ergebnisse lieferte; somit plädierte er für ein Modell, das eine »*reibungslose Integration von menschlicher und maschineller Übersetzung*« (Melby 1982: 217) über »*drei Stufen der Unterstützung*« (Hutchins 1998: 10) gewährleistet. Dabei soll – wie bei Kay – die Steuerung des Übersetzungsprozesses gänzlich beim Übersetzer bleiben. Diese »*three levels of assistance*« beschreibt Melby wie folgt (1992: 147):

- »level one includes word processing, telecommunications, and terminology management;«
- »level two adds text analysis, automatic dictionary look-up, and synchronized bilingual text retrieval;«
- »level three provides an interface to machine translation systems.«

Voraussetzung für Stufe 2 und 3 ist, dass die Textvorlage computerlesbar ist.

2.1.1.7 Erste kommerzielle Anwendungen

All diese von der Forschung geforderten Neuerungen wurden erstmals Mitte der 1980er Jahre von der Firma ALPS (Automated Language Processing Systems) in ihren *Multilingual Word Processor* integriert. Hutchins bezeichnet dieses System als »a true forerunner of the translation workstation« (Hutchins 1998: 13) und listet dessen wesentliche Merkmale wie folgt auf:

- Anzeige von Quell- und ZIELtext Seite an Seite am Bildschirm
- Automatische Übernahme von Formatierungen
- Möglichkeit, gesamte Stellen (z.B. Tabellen) zu kopieren und einzufügen
- Anlegen eines Glossars durch Abgleich des Quelltextes gegenüber eines gespeicherten Wörterbuchs, wobei die in diesem »textbezogenen Glossar« befindlichen Termini in einem »Referenzfenster« am Bildschirm angezeigt werden
- Anlegen einer zweisprachigen Datei mit allen Wiederholungen des Textes (»repetitions extraction«) und deren Übersetzungen
- optionale automatische Übersetzung

Zum Durchbruch kam es jedoch erst Anfang der 1990er Jahre im Zuge der technischen Fortschritte und der Demokratisierung des Personal Computers. Zu jener Zeit erfreute sich die TM-Technologie bereits eines immer größer werdenden Bekanntheitsgrads innerhalb der Gemeinschaft der professionellen selbstständigen Übersetzer. Programme wurden speziell für diese Zielgruppe – und nicht mehr ausschließlich für zahlungsstarke Übersetzungsfirmen (*Language Service Provider* – LSP) bzw. Übersetzungsabteilungen von Großunternehmen – entwickelt. So kommen Anfang der 1990er Jahre mehrere Produkte auf den Markt: *TranslationManager/2* (IBM), *Transit system* (STAR AG), sowie *Translator's Workbench* (Trados). Hutchins zufolge geht sogar der Begriff »Translation Memory« auf Trados zurück.

2.1.2 Kategorisierungsmodelle

Die Erörterung aller theoretischen Ansätze in Bezug auf eine Kategorisierung der Übersetzungstechnologien und den Platz der Translation Memory Systeme innerhalb derselben ginge über den Rahmen dieser Arbeit hinaus. Außerdem soll hier nicht näher über MT-spezifische Problemstellungen eingegangen werden, denn es ist nicht der Gegenstand dieser Arbeit. Trotz dieser Einschränkungen ist es unabdingbar, klare Kategorien auf den Weg hin zu einer Definition zu präsentieren.

2.1.2.1 Maschinelle Übersetzung – Humanübersetzung – Computer-Aided Translation

Zur Klassifizierung aller Übersetzungsprozesse hat sich in der Forschung die grundsätzliche Unterscheidung zwischen maschineller Übersetzung einerseits und Humanübersetzung andererseits durchgesetzt. Aufbauend auf Melbys »Grad der Automatisierung« in der Übersetzung erarbeiteten John Hutchins und Harold Somers 1992 eine Typologie, die folgende vier Kategorien berücksichtigt:

Humanübersetzung (HT – für »*human translation*«) bezeichnet den Vorgang, bei dem der menschliche Übersetzer seine Tätigkeit ohne »*computerbasierte linguistische Unterstützung*« (Hutchins und Somers 1992: 149) ausübt. Textverarbeitungsprogramme gelten zwar als computerbasierte Unterstützung, können aber nicht als »*linguistisch*« angesehen werden.

Maschinengestützte Humanübersetzung (MAHT – für »*machine-aided human translation*«) beschreiben Hutchins und Somers als den Übersetzungsvorgang, bei dem der Übersetzer ebenfalls im Mittelpunkt steht, jedoch zusätzlich »*computerbasierte linguistische Unterstützung*« in Anspruch nimmt. Dazu gehören Rechtschreib-, Grammatik- und/oder Stil-Prüfungstools, sowie online oder auf CD-ROM erhältliche ein- und zweisprachige Nachschlagewerke. Wenn diese Tools in einem integriert sind, bilden sie eine »*translator's workstation*« (ebd.: 150). Diese bietet oft zusätzlich die Möglichkeit, textbezogene Glossare zu erstellen, Wörter in lokalen oder externen Terminologiedatenbanken automatisch nachzuschlagen und einen maschinellen halbautomatischen Übersetzungsentwurf zu produzieren.

Humangestützte maschinelle Übersetzung (HAMT – für »*human-aided machine translation*«) wird hier als der Vorgang beschrieben, bei dem der Computer die Übersetzung ganzer Texte übernimmt. Der menschliche Übersetzer greift entweder »*während des Prozesses, in einer ›interaktiven‹ Weise, oder außerhalb des Prozesses, in Phasen der ›Präedition‹ oder ›Postedition‹*« (ebd.: 150) ein.

Vollautomatisierte hochqualitative Übersetzung (FAHQT – für »*fully automatic high quality translation*«) bezeichnet den Vorgang, bei dem der Computer ganze Texte übersetzt und druckreife Übersetzungen liefert, ohne dass der Mensch intervenieren muss.

Bei diesem Modell werden MAHT und HAMT als zwei getrennte Kategorien angeführt, die sich durch den Automatisierungsgrad und auch dadurch unterscheiden, ob primär der Mensch oder die Maschine die Übersetzung fertigt. Beide werden dennoch in einer Hauptkategorie zusammengefasst: die »*Computerunterstützte Übersetzung*« (»*Computer-Assisted Translation* (CAT)«; siehe Abbildung 2.1 auf der nächsten Seite). Interessant ist außerdem

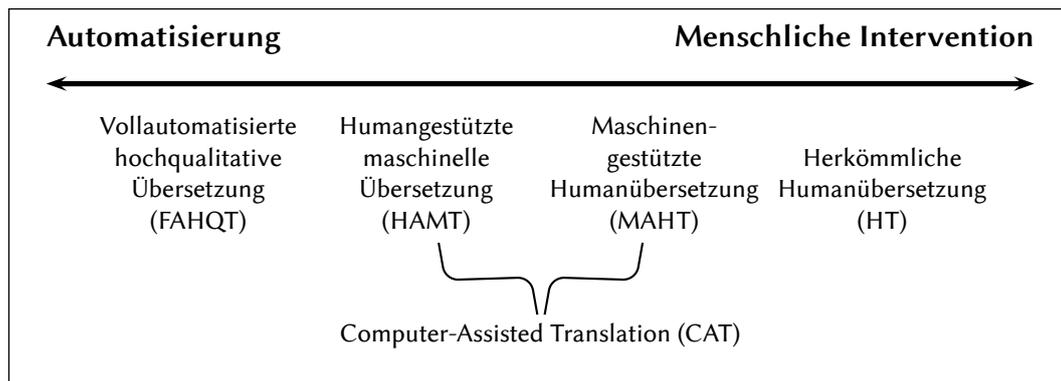


Abbildung 2.1: Klassifizierung der Übersetzungstypen (nach Hutchins/Somers 1992: 148)

die Feststellung, dass dieses Modell zwar den Begriff *translator's workstation* in die MAHT-Kategorie einordnet, die Begriffe *translation memory* oder *translation memory system* aber – noch – nicht ausdrücklich erwähnt.

An dieser Stelle soll darauf hingewiesen werden, dass in der Forschungsliteratur recht unterschiedliche Standpunkte in Bezug auf diese Kategorisierung vertreten sind. Dies liegt einerseits darin, dass es recht schwierig ist, klare Grenzen zwischen Systemen zu ziehen, die diese oder jene Funktionalität aufweisen; so betonen Hutchins und Somers: »*not only are the acronyms and names confusing but it is sometimes difficult to categorise systems as one or the other, hence the term >Computer-Aided Translation< (or >Computer-Assisted Translation<) is often used to cover all types.*« (Hutchins und Somers 1992: 147) Andererseits wurde dieses Modell angesichts des technologischen Fortschritts (insbesondere im Bereich der EDV und durch die Demokratisierung des PC) und neuerer Forschungsergebnisse immer wieder überarbeitet.

So subsumiert zum Beispiel Lynne Bowker (2002: 4) HAMT und FAMT in *machine translation*. Des Weiteren setzt sie MAHT mit CAT gleich, wobei sie – im Gegensatz zum Hutchins/Somers-Modell – elektronische Ressourcen, Internet, sowie Rechtschreib- und Grammatikprüfsysteme aus dieser Kategorie ausschließt und sie der HT zuschreibt.

Diesem Ansatz schließt sich Chiew Kin Quah an:

»As for human-aided machine translation and machine-aided human translation, the boundary between these two areas is especially unclear. Both classes are considered to be computer-aided translation [...] However [...] a different classification has been proposed where only machine-aided human translation is viewed as synonymous with computer-aided translation.« (Quah 2006: 7)

Elina Lagoudaki geht der Kategorisierungsproblematik aus dem Weg, indem sie nur mehr grundsätzlich zwischen menschlicher und maschineller Übersetzung unterscheidet (siehe Abbildung 2.2 auf der nächsten Seite). So erklärt sie:

»In fact, virtually all researchers in TT who have offered proposals for categorising translation systems have based their typology on the Hutchins and Somers model [e.g. Sager, 1994: 290] . According to this model, fully automatic Machine Translation systems are distinct from translation support systems, in the sense that the former attempt to produce translations automatically without the intervention of a human, whereas the second category covers interactive tools that are designed to support the human translator during the translation activity [Hutchins, 2004:2]. This distinction, based on the degrees of translation automation, is still relevant today and allows for a definition of a group of interactive translation support systems that includes Translation Memory systems, Terminology Management systems, concordance tools, localisation tools and a new type of translation workflow automation systems: Translation Management systems.« (Lagoudaki 2008: 25)

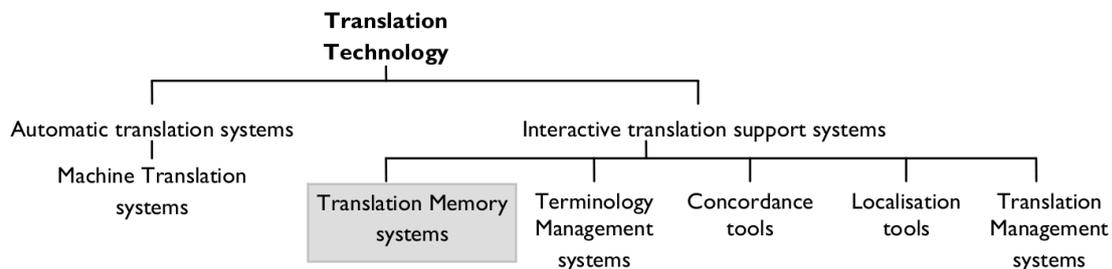


Abbildung 2.2: Einordnung der TM-Systeme in den Übersetzungstechnologien (Lagoudaki 2008)

Lagoudaki räumt jedoch ein, dass es selbst innerhalb dieser Gruppe von »interaktiven übersetzungsunterstützten Systemen« – wie sie CAT-Systeme nennt – schwierig sei, strenge Grenzen zu ziehen, da einige Systeme bereits mehrere der von ihr genannten Komponenten in sich vereinen. Sie fügt hinzu, dass dieses Phänomen sich in Zukunft sogar verstärken werde (vgl. ebd.: 26).

2.1.2.2 Ressourcen – Anwendungen

Wie im vorigen Kapitel erörtert, ist eine Kategorisierung der CAT-Technologien an Hand des »Grads der Automatisierung«, die sie bieten, teilweise unzureichend, was sich in den unterschiedlichen Standpunkten widerspiegelt: Je nachdem wie breit man sie definiert, können sie nämlich sämtliche bzw. wenige digitale Hilfen, deren sich der Übersetzer im Rahmen seiner Tätigkeit bedient, umfassen.

Aus diesem Grund weist Lagoudaki auf weitere Unterscheidungsmerkmale hin: ob es sich erstens um *Ressourcen* oder um *Tools* handelt (vgl. ebd.: 25) und, zweitens, ob diese Ressourcen bzw. Tools eher für den allgemeingebrauch oder speziell für die Übersetzung – oder

verwandte Tätigkeiten – entwickelt sind (vgl. Lagoudaki 2008: 27). So empfiehlt Lagoudaki eine Typologie, die auf brauchbaren Unterscheidungskriterien basiert und die Modelle von Bowker (2002: 7) und Esselink (2003: 79) ergänzt. Wie aus Tabelle 2.1 zu entnehmen, unterscheidet sie zwischen der Translation Memory, die eine Datei bzw. Datenbank ist und somit eine Ressource darstellt, und den Translation Memory Systemen, die wiederum zu deren Handhabung konzipierte Softwareanwendungen sind.

Digitalressourcen	Computeranwendungen	
	Übersetzerspezifisch	Allgemein
Wörterbücher und Thesauri	Translation Memory Systeme	Textverarbeitungsprogramme
Glossare	Terminologiemangement-Systeme	Webpage Design/Editor
Terminologische Datenbanken	Translation-Management-Systeme	Grafikbearbeitungssoftware
TM-Datenbanken	Lokalisierungstools	Rechtschreib- und Grammatikprüfer
Einsprachige Textkorpora	(Konkordanz-Tools)	Internetsuchmaschinen
Zweisprachige Textkorpora	(Machine-Translation-Systeme)	Desktop-Search-Software
Enzyklopädien		Desktop Publishing Software
Web		Optische Zeichenerkennung (OCR), u.v.m.

Tabelle 2.1: Einordnung der TM-Systeme innerhalb der CAT (nach Lagoudaki)

Des Weiteren unterstreicht Lagoudaki, dass ihre Definition bzw. Kategorisierung von CAT weiter gefasst ist, als die von Bowker (2002) – die ihrerseits alle Digitalressourcen ausschließt – und enger als die von anderen Forschern, die ihrer Auffassung nach CAT-Tools als Synonym für TM-Systeme betrachten (vgl. ebd.: 27).

2.1.3 Translator's workstation und TM-System

Wie bereits auf Seite 4 erwähnt, geht der Begriff von *Translator's workstation* auf Martin Kay zurück. Es handelt sich um ein System, das zahlreiche CAT-Technologien (Ressourcen und Anwendungen) in sich integriert, wobei die wichtigste davon die Translation Memory ist. Letztere wird im Sprachgebrauch mittlerweile sogar metonymisch für *Translator's workstation* verwendet. So schreibt Hutchins: »*Although the translator workstation integrates a variety of computer-based translation tools, such is the central role of the translation memory that they are now often referred to simply as >translation memory systems< «* (Hutchins 2005: 13).

Es darf die Frage aufgeworfen werden, inwiefern TW und TMS als deckungsgleiche Begriffe aufzufassen sind. Während Somers (2003), Hutchins (2005) und Quah (2006) noch von zwei separaten Systemen sprechen und TMS als Teil eines typischen TW beschreiben, geht Lagoudaki (2008) einen anderen Weg. Aus ihrer Definition von TMS geht hervor, dass sie beide Begriffe weitgehend als Synonym betrachtet⁵. So beschreiben selbst Softwareentwickler ihr Produkt⁶ wahlweise als »translation memory application«, »translation memory tool« oder »computer-aided translation tool«. Eines jedoch ist unbestreitbar: So wie es unterschiedliche Arten von *workstations* für unterschiedliche Berufe und Einsatzzwecke gibt (vgl. dazu Hutchins 1998: 2), so richten sich TM-Systeme primär an Übersetzer⁷ und haben eine (oder mehrere) Translation Memory(ies) als Kernkomponente.

2.1.4 Definition von TM-Systemen

Aus all diesen Erkenntnissen definiert Lagoudaki ein Translation Memory System wie folgt:

»Software application that includes a repository (most commonly in the form of a database) in which previous translations and their corresponding source text are stored in a structured and aligned way, so that any new text to translate is searched automatically and matched to the system's available resources in order for the system to be able to suggest a translation. The application normally comes with a set of standard tools (such as an alignment tool and various file format converters), as well as additional translation support tools (such as a terminology management tool, a concordance tool,

5 Selbst Hutchins und Somers nennen Translator's Workstation alternativ auch »Translator's Workbench«. Da dies ebenfalls der Name eines Produktes der Firma Trados ist, ist es durchaus vorstellbar, dass im Sprachgebrauch der Produktname zum Gattungsnamen geworden ist. Es wäre somit nicht verwunderlich, wenn sich die Bezeichnung Translation Memory System deshalb nach und nach durchsetzt, da sie möglicherweise »korrekter« klingt.

6 In diesem Beispiel: OmegaT.

7 So sind folgerichtig Lokalisierungstools primär für Lokalisierer konzipiert, obwohl diese Tools durchaus auf TM-Technologien zurückgreifen können (vgl. dazu Lagoudakis Modell, Abbildung 2.2 auf Seite 9).

etc.), thus offering an interactive translation support environment. It is normally used by translation professionals, especially by translators.« (Lagoudaki 2008: 27)

An Hand dieser Definition können die **wesentlichen** Merkmale eines TM-Systems folgendermaßen aufgelistet werden:

- TM-Systeme sind Softwareanwendungen.
- Sie bieten eine »interaktive übersetzungsunterstützende Umgebung« (vgl. TW).
- Sie werden hauptsächlich von Übersetzern verwendet.
- Sie greifen auf Ressourcen in Form einer Datei oder Datenbank (die Translation Memory) zurück, in denen Originaltexte (bzw. Textstellen) und deren Übersetzungen gegeneinander und strukturiert gespeichert sind.
- Die Suche innerhalb dieser Ressourcen erfolgt automatisch.
- Ein Datenvergleich findet statt und
- Übersetzungsvorschläge werden formuliert.

Übliche Merkmale eines TM-Systems sind des Weiteren:

- Ein Tool zum Alinieren von Texten
- Verschiedene Tools, um unterschiedliche Dateiformate zu unterstützen.

Optionale Merkmale sind:

- Ein Terminologie-Management-Tool
- Eine Funktion zur Konkordanzsuche

2.2 Open Source bzw. Freie Software

Wie Georg Hüttenegger in der Einleitung zu seinem 2006 veröffentlichten Buch *Open Source Knowledge Management* anmerkt, hat der Begriff »Open Source« *»keine eindeutige Definition, sondern stellt eine Art Oberbegriff dar. Generell geht es darum, dass der Quellcode (Source) eines Produktes zugänglich ist.«* (Hüttenegger 2006: 4) Dieser Definitionsmangel ist weitgehend darauf zurückzuführen, dass sich hinter dem Begriff »Open Source« eine Gemeinschaft von verschiedenen Akteuren – Programmentwicklern, Anwendern, Unternehmern – verbirgt, die unterschiedliche Standpunkte vertreten. Betrachtet man dennoch die Entstehungsgeschichte dieses Begriffes, lassen sich grundsätzliche Merkmale erkennen.

2.2.1 GNU/Linux

Wer sich je die Frage gestellt hat, warum heutzutage nahezu alle PC werkseitig mit Microsofts Betriebssystem Windows® ausgestattet sind, wird ähnlich wie der damals seit 1971 am Labor für Künstliche Intelligenz des MIT beschäftigte Richard Stallman 1984 vorgegangen sein, mit dem einzigen Unterschied, dass das damals verbreitetste Betriebssystem UNIX war. So schreibt er:

»1984 war es unmöglich, einen modernen Computer zu nutzen, ohne ein proprietäres Betriebssystem installieren zu müssen, ein Betriebssystem [!] , das man nur mit einer stark einschränkenden Lizenz erhalten konnte. Niemand durfte Software frei mit anderen Computermittbenutzern teilen, und kaum jemand war in der Lage, die Software den eigenen Bedürfnissen anzupassen. Die Besitzer der Software hatten Mauern errichtet, um uns voneinander zu trennen. Das GNU-Projekt wurde gegründet, um all das zu ändern. Das erste Ziel: Die Entwicklung eines Unix-kompatiblen, portablen Betriebssystems, das zu 100% Freie Software sein sollte. Nicht 95% frei, nicht 99,5%, sondern 100%, sodaß die Anwender die Freiheit haben, das gesamte System weiter zu verteilen, es zu verändern und an jedem seiner Teile mitzuarbeiten. Der Name des Systems – GNU – ist ein rekursives Akronym mit der Bedeutung ›GNU ist Nicht Unix«. So wird zum einen Unix anerkannt, während zum anderen klar gemacht wird, daß GNU etwas anderes ist. Technisch gesehen ist GNU wie UNIX. Aber ungleich UNIX gibt GNU den Anwendern Freiheit.« (Stallman 1999a)

Aus diesen fünfzehn Jahre nach Beginn seines Projektes formulierten Sätzen wird klar, worum es Stallman geht, nämlich um ein bestimmtes, auf die Informationstechnologien angewandtes Freiheitsverständnis. So kündigte er in jenem Jahr seine Stelle beim MIT und begann, am GNU-Projekt, der Entwicklung eines freien Betriebssystems mit einer Reihe von Gleichgesinnten – die sich *Hacker* nennen – zu arbeiten.

In einem weiteren 1999 verfassten Aufsatz erklärt Stallman, dass das GNU-System um 1990 beinahe fertig war. Allerdings fehlte noch ein wesentlicher Teil: der Kernel.

»Im Jahr 1991 entwickelte Linus Torvalds einen Unix-kompatiblen Kernel und nannte ihn Linux. Um 1992 resultierte das Kombinieren von Linux mit dem noch nicht ganz fertigen GNU System in einem komplett freien Betriebssystem. [...] Wir nennen diese Systemversion GNU/Linux, um seine Zusammenstellung als eine Kombination von dem GNU System mit Linux als Kernel auszudrücken.« (Stallman 1999b)

2.2.2 Die vier Freiheiten

Die von Stallman geforderten Freiheiten für den Anwender sind folgende (Stallman 1996a):

- Freedom 0: The freedom to run the program, for any purpose.

- Freedom 1: The freedom to study how the program works, and adapt it to your needs. (Access to the source code is a precondition for this.)
- Freedom 2: The freedom to redistribute copies so you can help your neighbor.
- Freedom 3: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (Access to the source code is a precondition for this.)

2.2.3 Lizenzmodell: die GPL

Demnach gilt wiederum jede Software als »proprietary«, die dem Anwender keine oder nur einige dieser Freiheiten auf Grund eines restriktiven Copyrights gewährt. Um diese Freiheiten zu garantieren und zu schützen, werden Freie Softwares unter das in Anlehnung am Copyright-Begriff genannte »Copyleft« gestellt.

»Um ein Programm unter das Copyleft zu stellen, stellen wir es zuerst unter das Copyright; dann fügen wir als Rechtsmittel Vertriebsbestimmungen hinzu, die allen erlauben, den Code des Programms *oder jedes davon abgeleiteten Programms* zu verwenden, zu ändern und weiter zu verteilen, aber nur wenn die Vertriebsbestimmungen unverändert bleiben. So werden der Code und die gewährten Freiheiten rechtlich untrennbar.« (Stallman 1996b)

Rechtlich ist der Copyleft-Grundsatz durch verschiedene Lizenzen verankert, von denen die *General Public Licence* (GPL) und die *Less General Public Licence* (LGPL) die bekanntesten und verbreitetsten sind.

Selbst wenn Freie Softwares meist kostenlos verfügbar sind, soll an dieser Stelle betont werden, dass das Adjektiv »frei« (English: *free*) auf Freiheit hinweist: »*Free software* is a matter of liberty, not price. To understand the concept, you should think of *free* as in *free speech*, not as in *free beer*.« (Stallman 1996a) Nichtsdestotrotz dürfen – und werden – Freie Softwares gegen Entgelt vertrieben. Ganze Geschäftsmodelle gründen sogar auf dem entgeltlichen Vertrieb und Support solcher Softwares.

2.2.4 Free Software Foundation

1985 gründete Stallman die *Free Software Foundation* (FSF) mit dem Ziel – wie es auf deren deutschsprachigen Seite heißt: »*durch ihre Aktivitäten das Verständnis und die Unterstützung für Freie Software in der Öffentlichkeit, Politik und Gesetzgebung maßgeblich zu verbessern.*«⁸ Diese – nach eigenen Angaben – »*nicht gewinnorientierte und in einigen Staa-*

⁸ FSFE - Free Software Foundation Europe. URL: <http://fsfe.org/index.de.html> (besucht am 23.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXexyHb>.

ten als gemeinnützig anerkannte Nichtregierungsorganisation« gibt Empfehlungen heraus, organisiert Tagungen und pflegt eine Liste von Lizenzen, die in ihren Augen die oben genannten Freiheiten schützen.

2.2.5 Open Source Initiative

Neben der Bezeichnung Freie Software entstand der Begriff »Open Source« als die Nichtregierungsorganisation *Open Source Initiative* (OSI) 1998 in Kalifornien gegründet wurde. Den geschichtlichen Hintergrund dieser Gründung bildet die Veröffentlichung des Quellcodes ihres Webbrowsers durch die Firma Netscape. Einige Freie Software- und GNU/Linux-Anhänger erkannten in diesem Schritt einen Wendepunkt und plädierten von nun an für eine Annäherung zwischen Freie Software-Gemeinschaft und Industrie. Da in ihren Augen die Bezeichnung »frei« die Geschäftsleute abschreckte, wurde der Begriff »Open Source« geprägt:

»The ›open source‹ label was invented at a strategy session held on February 3rd, 1998 in Palo Alto, California. [...] [Eric] Raymond had been invited out by Netscape to help them plan their browser source-code release. The strategy session grew from a realization that the Netscape announcement had created a precious window of time within which we might finally be able to get the corporate world to listen to what the hacker community had to teach about the superiority of an open development process. The conferees decided it was time to dump the moralizing and confrontational attitude that had been associated with ›free software‹ in the past and sell the idea strictly on the same pragmatic, business-case grounds that had motivated Netscape. They brainstormed about tactics and a new label. ›Open source‹, contributed by Chris Peterson, was the best thing they came up with. Over the next week Raymond and others worked on spreading the word. Linus Torvalds gave an all-important imprimatur the following day. [...] Richard Stallman flirted with adopting the term, then changed his mind.«⁹

Genauso wie die FSF gibt die OSI Empfehlungen heraus und pflegt eine Liste von Lizenzen, die mit ihren Grundsätzen vereinbar sind. Man erkennt jedoch, dass OSI-anerkannte Lizenzen nicht zwangsläufig FSF-anerkannt sein müssen, da Meinungsverschiedenheiten beide »Schulen« voneinander trennen.

2.2.6 Zusammenfassung

Obwohl beide Bezeichnungen »Freie Software« und »Open Source Software« scheinbar austauschbar sind, wurden in diesem Kapitel einige geschichtlich bedingte Unterschiede fest-

⁹ Open Source Initiative (OSI): *History of the OSI*. URL: <http://www.opensource.org/history> (besucht am 24.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZYQqtV>.

gestellt: Während der Begriff »Freie Software« der gesellschaftspolitischen oder gar humanistisch-philosophischen Anschauung der Hackergemeinschaft zuzuschreiben ist, entstand der Begriff »Open Source« im Zuge pragmatischer und geschäftsorientierter Überlegungen. Der einen Betrachtungsweise ist es zu verdanken, dass das freie Betriebssystem GNU/Linux sowie eine Reihe Freier Softwares entstanden ist¹⁰. Die andere Betrachtungsweise bewirte, dass Privatunternehmen sich durch ihre Finanz- und Innovationskraft vermehrt an der Open Source-Idee beteiligten, wodurch sie einem breiteren, nicht nur auf einige begabte Programmierer beschränkten Publikum zugänglich gemacht wurde. Letztendlich entstand dadurch ein Markt, dessen Wachstum auf jährlich 22,4% geschätzt wird und 2013 eine Größe von USD 8,1 Mrd. erreichen dürfte.¹¹

Schließlich soll darauf hingewiesen werden, dass sich zahlreiche Gemeinschaftsmitglieder über die Meinungsverschiedenheiten zwischen FSF und OSI hinwegsetzen und ihre Projekte FOSS (*free and open source software*) nennen. Um die unterschiedlichen Softwarekategorien zu veranschaulichen, wird auf Chao-Kueis Diagramm verwiesen (Abbildung 2.3).

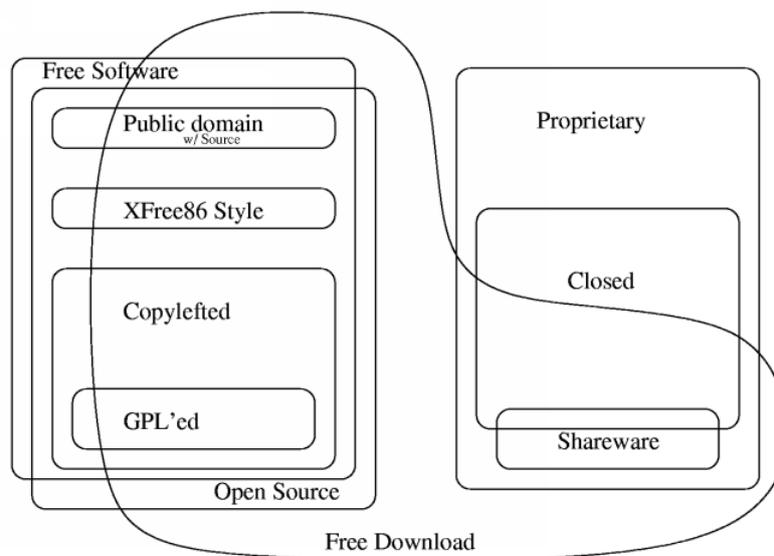


Abbildung 2.3: Kategorisierung von Softwares nach Lizenztypen (Autor: Chao-Kuei)¹²

10 Allein auf der größten Open Source Entwicklungsplattform sourceforge.net werden über 150 000 Projekte beherbergt.

11 Vgl. IDC: *Open Source Software Market Accelerated by Economy and Increased Acceptance From Enterprise Buyers*, IDC Finds. URL: <http://www.idc.com/getdoc.jsp?containerId=prUS21947509> (besucht am 24.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXKcCfX>.

12 Aus: Free Software Foundation: *Categories of Free and Non-Free Software - GNU Project*. URL: <http://www.gnu.org/philosophy/categories.html> (besucht am 24.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsX1nCGe>.

3 Untersuchungsvorbereitung

3.1 Gegenstand

Im Rahmen dieser Arbeit werden ausschließlich freie¹³ Translation Memory Systeme, die für den Einsatz durch Übersetzer konzipiert sind. Software, die hauptsächlich im Bereich der Software- oder Websitelokalisierung (wie Virtaal) eingesetzt, nicht weiterentwickelt werden (wie Transolution) oder noch nicht veröffentlicht wurden (wie OpenTMS) sind von dieser Untersuchung ausgeschlossen.

3.1.1 OmegaT

OmegaT ist eine freie Softwareanwendung, die seit 2000 entwickelt wird. In Java geschrieben ist sie plattformunabhängig und läuft sowohl unter MS-Windows als auch unter Mac und GNU/Linux. Sie bietet eine eigene Übersetzungsumgebung, die aus einem Texteditor (Hauptfenster) sowie integrierten TM-, Glossar- und Wörterbücher-Suchfunktionen (Nebenfenster) besteht. Im Rahmen dieser Arbeit wurde die aktuellste Version (2.0.4) mit dem »tokenizer«-Plug-in getestet.

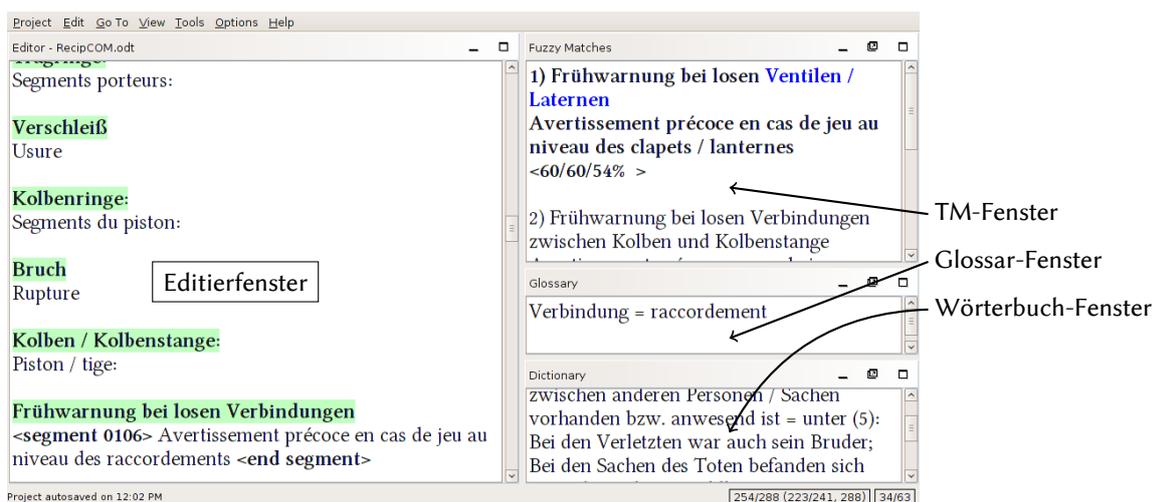
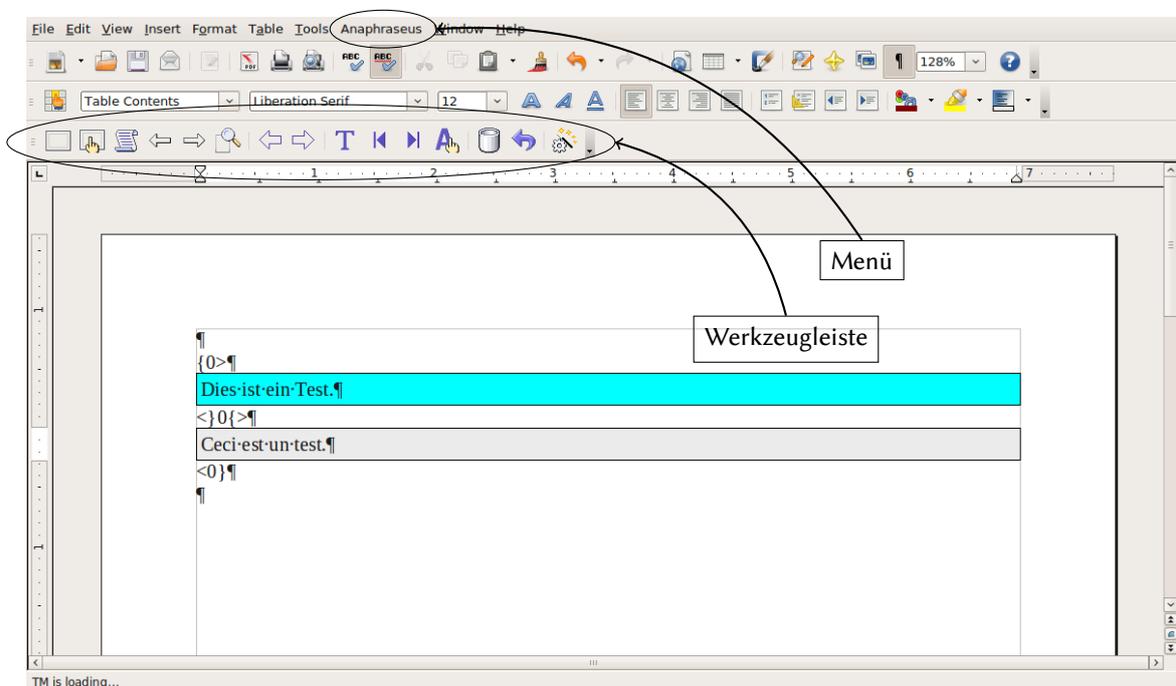


Abbildung 3.1: OmegaT-Übersetzungsumgebung

13 Freie Software im Sinne der FSF.

3.1.2 Anaphraseus

Bei Anaphraseus handelt es sich um eine Makro-Sammlung für das OpenOffice.org Textverarbeitungsprogramm Writer. Somit ist es keine eigenständige Softwareanwendung, vielmehr ein Ergänzungsmodul, das die Verwendung von TM-Technologien ermöglicht. Es wurde in Anlehnung an Wordfast – das seinerseits eine Makro-Sammlung für Microsoft Word ist – entwickelt. Die Erstversion wurde im Oktober 2007 von Oleg Tsygani veröffentlicht (Medvedev 2008: 2) und wird seitdem ständig weiterentwickelt. Es ist plattformunabhängig (sofern OpenOffice.org installiert ist) und ebenfalls frei, da es unter der GPL steht. Getestet wurde die Version 1.23b.



3.1.3 Weitere Anwendungen

Neben den oben genannten zwei TM-Systemen existiert eine Reihe freier Anwendungen, die als Ergänzungstools für bestehende TM-Systeme konzipiert sind (z.B. in Form von Formatkonvertierungstools, Alignmenttools, u.v.m). Sofern es sinnvoll ist, auf alternative Systeme einzugehen, werden sie an geeigneter Stelle beschrieben werden.

3.2 Vergleichskriterien

Bei der Auswahl von objektiven Kriterien zum kritischen Vergleich der oben genannten freien TM-Systeme wurden nachstehende Forschungsergebnisse berücksichtigt.

3.2.1 EAGLES-Framework

Ein erster umfassender Versuch zur Festlegung von Methoden und Kriterien für die Evaluierung von Systemen und Produkten aus dem Bereich der Sprachentechnologie wurde auf Wunsch der Europäischen Kommission zwischen 1993 und 1998 von der EAGLES-Arbeitsgruppe gemacht. Die Ergebnisse¹⁴ ergänzen die in den Normen ISO/IEC 9126-1 und ISO/IEC 14598-1 formulierten Kriterien und Methoden zur Evaluierung von Softwareprodukten. (Vgl. King 1998: 152) Diese Kriterien lassen sich in sechs Hauptkategorien unterteilen:

Funktionalität Welche Funktionen besitzt das TM-System? Erfüllen sie die Bedürfnisse und Erwartungen der Anwender?

Benutzbarkeit Wie »anwenderfreundlich« ist das TM-System in Hinblick auf Erlernbarkeit, Anpassungsmöglichkeiten, usw.?

Effizienz Wie leistungsfähig sind die verfügbaren Funktionen? Wie hoch sind die zum Betrieb der Anwendung benötigten Zeit- und Computerressourcen?

Erweiterbarkeit Inwiefern ist es dem Anwender möglich, das TM-System für seine eigenen Bedürfnisse zu erweitern?

Übertragbarkeit Inwiefern ist das TM-System mit anderen Systemen kompatibel?

Zuverlässigkeit Wie zuverlässig ist das TM-System in Bezug auf System- oder Programmabsturz?

3.2.2 François Massion

In seinem 2005 veröffentlichten Buch *Translation Memory Systeme im Vergleich* präsentiert François Massion die Ergebnisse einer Reihe von Tests, denen er sechs marktführende kommerzielle TM-Systeme unterzogen hat, um diese miteinander zu vergleichen. Obwohl er weder Open Source Anwendungen getestet noch ausdrücklich in diesem Werk darauf hinweist, nach welchen Kriterien er die zu untersuchenden Merkmale und Funktionen ausgewählt hat,

¹⁴ Nachzulesen unter *EAGLES Report*. URL: <http://www.issco.unige.ch/en/research/projects/ewg95/node1.html> (besucht am 25. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsX7TBvF>.

bietet seine Untersuchungsmethode durchaus brauchbare Evaluierungskriterien an, die »*unterschiedliche Tests und messbare Parameter definiert, um möglichst weitgehend subjektive Urteile zu vermeiden.*« (Massion 2005: 19)

3.2.3 Elina Lagoudaki

In ihrer 2008 am Londoner Imperial College unter der Betreuung von Mark Shuttleworth abgefassten Dissertation widmet sich Elina Lagoudaki ebenfalls der kritischen Untersuchung von – überwiegend jedoch nicht ausschließlich – kommerziellen TM-Systemen. Ihre Ergebnisse beruhen auf einer anwenderorientierten empirischen Methode in der Form einer Online-Erhebung, an der 874 Personen aus dem Bereich der Sprachtechnologien (90% davon Übersetzer) zwischen dem 1. Juli und dem 1. September 2006 teilnahmen. Ziele dieser Erhebung waren (Lagoudaki 2006: 10):

- »to establish the needs of translation professionals via their practices and working habits during the translation process;«
- »to reveal the tasks related to TM use;«
- »to distinguish the profiles of different TM user groups according to criteria such as the type of tasks they perform, their professional status, their years of working experience, their computer usage competence, etc.«
- »to provide an insight into the work environment in which translation professionals carry out the translation activities today;«
- »to estimate the TM technology penetration in the translation market;«
- »to help understand the reasons behind low usage of TM technology and to discover missed opportunities for reaching potential users;«
- »to uncover user satisfaction levels for existing TM systems;«
- »to open the way to new ideas about future systems and identify possibilities for expanding the functionality and scope of use of TM systems.«

Wie aus ihrer Zielsetzung zu erkennen ist, verfolgt Lagoudaki einen praxisorientierten Ansatz, und versucht, »den Bedürfnissen«, »der Zufriedenheit« und den Erwartungen der TM-System-Anwender auf den Grund zu gehen.¹⁵ Um die Kriterien auszuwählen, nach denen sie ihre Studie gestaltet, stützt sie sich auf das Modell der EAGLES-Arbeitsgruppe, wie es von King (1998) dargestellt und von Monika Höge in ihrer 2002 an der Universität Helsinki abgefassten Dissertation »Towards a Framework for the Evaluation of Translators' Aids« weitergeführt wurde, aber auch auf die subjektive Einschätzungen einer kleinen Gruppe von Testpersonen:

15 Auf die aus dieser Studie gewonnenen statistischen Erkenntnisse wird an geeigneter Stelle eingegangen.

» [...] users were asked to rate the TM system they were using most frequently in terms of functionality, reliability, efficiency in match recall, efficiency in match precision, efficiency in speed, usability, learnability, price to usefulness ratio and customer support. [...] The selection of criteria was based on the quality metrics proposed in the EAGLES framework [...] and on Höge's framework for evaluating translators' aids systems [...], and was finalised in the discussion that took place at the brainstorming session. The session's participants concluded that these were the most important criteria that TM users consider when evaluating their TM tool, either when comparing it to other tools or when they assess its quality against their expectations or its stated purpose.« (Lagoudaki 2008: 101f)

3.2.4 Zusammenfassung

In den folgenden Kapiteln werden die TM-Systeme nach Merkmalen untersucht, die von der EAGLES-Arbeitsgruppe, von Massion und von Lagoudaki als relevant identifiziert wurden. Diese Merkmale werden in der Folge in funktionale und nichtfunktionale unterteilt.

4 Funktionale Merkmale: Vor dem Übersetzen

Im Vorfeld zum eigentlichen Übersetzungsvorgang müssen beim Einsatz von TM-Systemen einige Schritte vom Benutzer erledigt werden. Über welche unterstützende Funktionen die untersuchten Systeme verfügen, wird in diesem Abschnitt beschrieben.

4.1 Projektmanagement

Keines der untersuchten Systeme bietet eine Workflow-Lösung an – wie sie beispielsweise in der webbasierten kommerziellen Anwendung accross (Enterprise-Version) implementiert ist (Massion 2005: 40f.) –, bei der Personal- und Zeitressourcen verwaltet werden können. Da derartige Produkte jedoch vor allem für größere Sprachdienstleister gedacht sind, wird hier eine andere Definition für den Begriff »Projekt« verwendet.

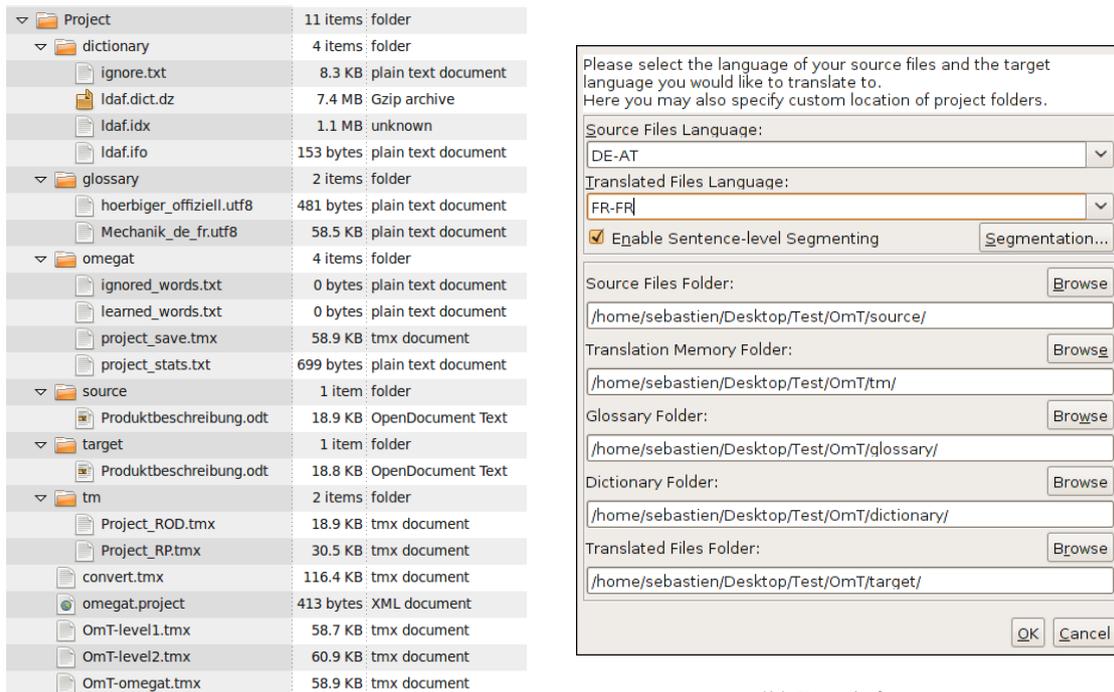
Bei den untersuchten TM-Systemen bezieht sich das Wort »Projekt« *»auf eine Sammlung von Ordnern und Dateien in Zusammenhang mit einer Übersetzung. Ein »Projekt« ist üblicherweise Synonym für Übersetzungsauftrag.«* (Welsh und Prior 2009: 7)

4.1.1 Projekt anlegen

OmegaT Nachdem der Speicherort für den Projekthauptordner ausgewählt wurde, erscheint das in Abbildung 4.1b auf der nächsten Seite dargestellte Dialogfenster. Hier werden projektweite Parameter festgelegt: Ausgangs- und Zielsprache in Zwei- bzw. Vierbuchstabenkodierung (ISO-Kodierung)¹⁶, Auswahl zwischen Absatz- oder Satzsegmentierung, Segmentierungsregeln, sowie Namen und Speicherort der Projektunterordner¹⁷. Anschließend öffnet sich ein Dialogfenster mit Optionen für das Importieren der zu übersetzenden Ausgangstexte. Alternativ dazu können die zu übersetzenden Textdateien in den Ordner /source/ und die sonstigen projektrelevanten Dateien – Translation Memory, Glossar, Wörter-

¹⁶ Letztere Option ermöglicht die Unterscheidung zwischen Sprachvarianten (sogenannten *locales*) und ist für die Verwendung von integrierten Wörterbüchern (Kapitel 5.3.1 auf Seite 57) sowie der integrierten Rechtschreibprüfung (Kapitel 5.5.3 auf Seite 63) relevant.

¹⁷ Die Standardeinstellungen treffen bei Namen und Speicherort der Projektunterordner auf die meisten Projekte zu.



(a) Vollständige Ordnerstruktur

(b) Projektfenster

Abbildung 4.1: Projektanlegen in Omega T

buch – in die eigens zu diesem Zweck angelegten jeweiligen Ordner abgespeichert und das Projekt durch Drücken der  Taste neu geladen werden.

Die von OmegaT angelegte projektbezogene Ordnerstruktur bietet eine übersichtliche Art und Weise an, die unterschiedlichen Dateien zu ordnen. Außerdem können beliebig viele Ausgangsdateien in den Ordner `/source/` gespeichert werden und diese wiederum in beliebig tiefe Unterordner. Laut Lagoudaki entspricht diese Funktionalität den Wünschen von Studienteilnehmern: »A further request expressed by a number of translators and project managers was the possibility of keeping the files ordered in folders and subfolders within a project, thus making it easier to organise large and complex projects.« (Lagoudaki 2008: 164) Dies ermöglicht auch die gleichzeitige Suche nach bestimmten Termini innerhalb sämtlicher Ausgangs-, Ziel-, TM- und Glossardateien (siehe Konkordanzsuche auf Seite 59): »Loading more than one file in a project is also very useful [...] allowing the translator to search all the files at once for every occurrence of a particular term in order to understand how it is used.« (ebd.: 163)

Anaphraseus Genau wie Wordfast, von dem es sich inspiriert, bietet Anaphraseus kein Projektmanagement an. Ein »Projekt« bezieht sich somit lediglich auf die Datei, die gerade in OpenOffice.org Writer geöffnet ist. Es müssen dennoch Ausgangs- und Zielsprache (ISO-Kodierung) festgelegt, sowie Projektdateien wie TM und Glossar entweder erstellt oder ausgewählt werden.

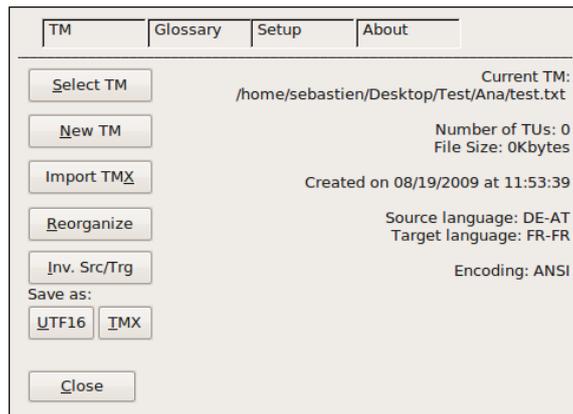


Abbildung 4.2: Projektfenster von Anaphraseus

4.1.2 Projektelemente ändern

Eine weitere Erwartung von Studienteilnehmern beschreibt Lagoudaki wie folgt: *»The system should also allow the user to add, remove or update files in his projects as they progress. This kind of flexibility can be very useful when translators are working for clients who keep revising the documents they have sent as work progresses.«* (Lagoudaki 2008: 164)

OmegaT In OmegaT ist es ohne Weiteres möglich, Ausgangsdateien durch neuere zu ersetzen, nachdem das Projekt angelegt und sogar mit der Übersetzung begonnen wurde. Allerdings ist es dabei erforderlich, das Projekt erneut zu laden (durch Drücken der  Taste) und sich Segment für Segment durch den Text zu arbeiten. Segmente, die bereits übersetzt wurden und in den neuen Ausgangsdateien in der gleichen Form und mit dem gleichen Inhalt wie in den alten Ausgangsdateien vorkommen, werden vom System automatisch ersetzt. Lediglich die modifizierten Segmente bleiben zu bearbeiten. Diese Funktion ist besonders nützlich, wenn der Übersetzer während des Projektes eine aktualisierte Vorlage vom Auftraggeber erhält.

Anaphraseus Dadurch, dass es sich mit Anaphraseus bei jeder einzelnen geöffneten Datei um das jeweilige »Projekt« handelt, ist diese Möglichkeit nicht gegeben.

4.1.3 Sprachen

Von TM-Systemen wird erwartet, dass sie möglichst viele, wenn nicht alle Sprachen der Welt – mit ihren unterschiedlichen Zeichensystemen und Schreibrichtungen – unterstützen.

OmegaT Beim Anlegen eines Übersetzungsprojektes stehen in OmegaT eine Vielzahl an Sprachen (inkl. *locales*) für die Festlegung von Ausgangs- und Zielsprache zur Verfügung.

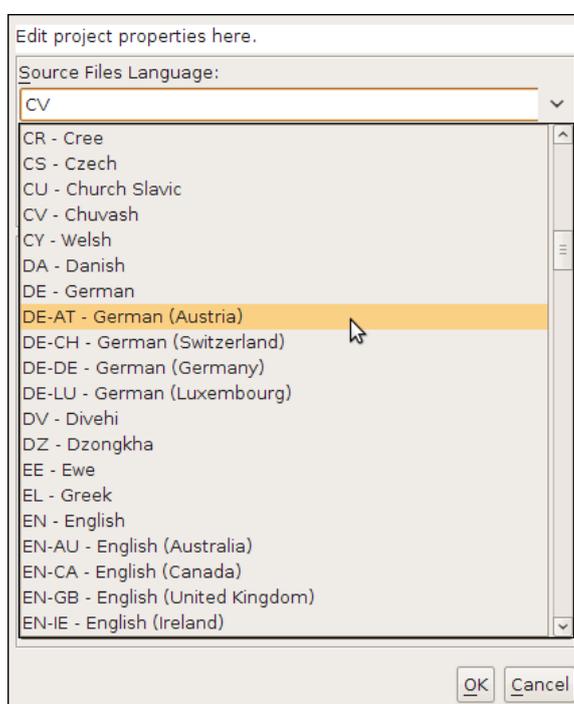


Abbildung 4.3: Sprachenauswahlfenster in OmegaT

OmegaT unterstützt standardmäßig Unicode (UTF-8) und bidirektionale Schreibsysteme, was die korrekte Speicherung und wiedergabe von nichtwestlichen Schriftzeichen ermöglicht. So kann in OmegaT über die Tastaturkombination **Strg** + **↑** + **O** die Schreibrichtung mitten im Satz geändert werden, um beispielsweise einen Tag oder einen Markennamen richtig einzufügen.

Anaphraseus In Anaphraseus ist die Auswahl an Sprachen unbegrenzt, sofern die eingegebenen ISO-Codes den TMX-Standards entsprechen. Das Sprachenauswahlfenster verweist sogar auf die von Wordfast verwendeten Codes.

4.1.4 Segmentierung

Abgesehen von der Größe der Übersetzungseinheiten, die im Fall von OmegaT auf Absätze oder Sätze einstellbar ist (siehe Seite 22), bieten die meisten TM-Systeme Funktionen zur feinen Einstellung der Satzsegmentierung an. Zur Segmentierungsfunktion schreibt Massion:

»Unter Segmentierung versteht man die Aufteilung eines Textes in Übersetzungseinheiten. Eine Übersetzungseinheit ist nicht zwangsläufig ein Satz, aber am geläufigsten ist die Segmentierung in Sätze. Unterschiedliche Segmentierungsregeln führen dazu, dass es zu Unterschieden beim Austausch von Translation Memories zwischen einzelnen Systemen gibt. Wenn beispielsweise das System A ein Tab als Segmentende festlegt, während das System B dies nicht tut, dann wird das Memory aus dem System A nicht alle Sätze des anderen Systems erkennen können.« (Massion 2005: 122)

In diesem Zusammenhang erinnert er, dass die LISA (*Localization Industry Standard Association*) einen Standard zum Austausch von Segmentierungsregeln erkennt: den SRX (für *Segmentation Rules eXchange*). Im Vorfeld zu seiner Untersuchung merkt er an: »Die untersuchten Systeme unterscheiden sich in der Art und Weise, wie Segmentierungsregeln definiert werden und wie flexible Ausnahmen zu den Regeln gehandhabt werden können.« (ebd.: 122) Dies gilt ebenfalls für die im Rahmen dieser Arbeit untersuchten freien TM-Systeme.

OmegaT Dieses System besitzt zahlreiche sprachenbezogene Segmentierungsregeln, die bei der Festlegung der Projektsprachen automatisch geladen werden. Sie werden nach Trefflichkeit von oben nach unten gereiht und auch in dieser Reihenfolge auf den Text angewandt.

Die `default`-Regel bewirkt, dass nach den Satzzeichen, die normalerweise das Ende eines Satzes darstellen (Punkt, Ausrufe- und Fragezeichen) auch tatsächlich die aktuelle Übersetzungseinheit geschlossen wird und eine neue beginnt. Die auf den Dateityp anwendbare `Text files segmentation`-Regel bewirkt ihrerseits zusätzlich, dass eine neue Übersetzungseinheit nach einem Absatzzeichen begonnen wird. Diese Regeln sind insofern anpassbar, als man zusätzliche Zeichen als Abgrenzungszeichen definieren kann.

Neben diesen Regeln, die den Text in Übersetzungseinheiten aufteilen, gibt es die weit- aus zahlreicheren Ausnahmeregeln, die das Gegenteil bewirken: So darf nach bestimmten Abkürzungen wie »bzgl.« oder »bzw.« keine neue Übersetzungseinheit begonnen werden. Einige dieser für die deutsche Sprache spezifischen Ausnahmeregeln sind in Abbildung 4.4 auf der nächsten Seite dargestellt. Auch in diesem Fall können Ausnahmeregeln beliebig definiert werden. Geschieht dies jedoch nachdem das Übersetzungsprojekt angelegt und mit der Arbeit begonnen wurde, ist es erforderlich, das Programm durch Drücken der -Taste erneut zu laden.

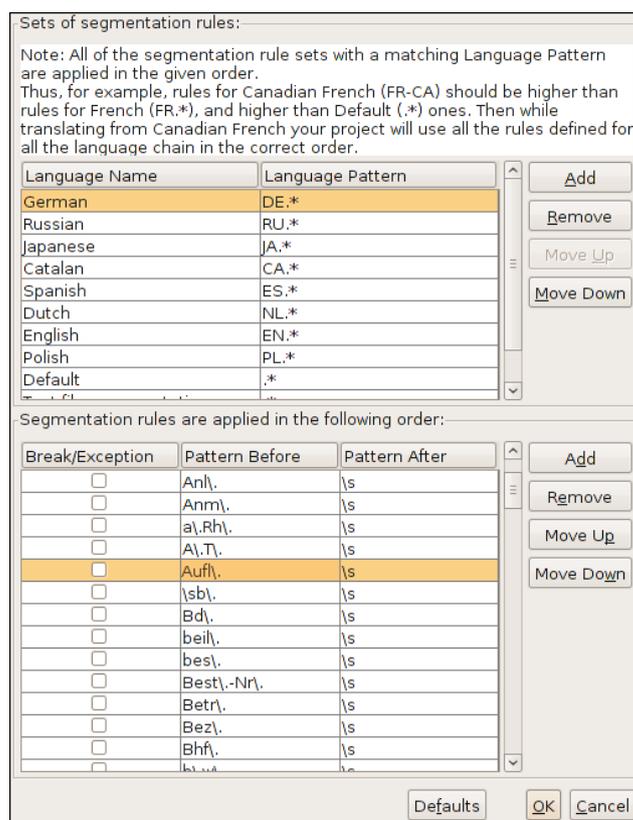


Abbildung 4.4: Segmentierungsregeln in OmegaT

Hier soll angemerkt werden, dass OmegaT den SRX-Standard nicht unterstützt, obwohl es dem in seiner Syntax recht nahe kommt. Außerdem ist es nicht möglich, Segmentierungsregeln zu importieren oder zu exportieren:

»The Sentence Segmentation has been implemented with the help of the Segmentation Rules eXchange (SRX) standard – please note that not all SRX features are supported. And it is not possible to import/export rules defined in SRX format. However, if you know how SRX works, you will already know a lot about how OmegaT does the segmentation.« (*OmegaT 2 User Manual*)

Anaphraseus Anaphraseus ist ein recht neues Programm und verfügt leider noch über keine Bedienungsanleitung. Somit konnte nicht in Erfahrung gebracht werden, ob es über eine Funktion zur Anpassung von Segmentierungsregeln verfügt. Ein einfacher Test zeigte, dass es keine sprachspezifischen Ausnahmeregeln besitzt: So wurde ein Satz, der die Abkürzung »z.B.« beinhaltet, in zwei Übersetzungseinheiten aufgeteilt.

Trotz dieses Mangels ist es in Anaphraseus – im Gegensatz zu OmegaT – möglich, Seg-

mente während des Übersetzungsvorgangs zu verkleinern oder zu vergrößern (siehe dazu Kapitel 5.1.3 und Abbildung 5.1 auf Seite 50).

Okapi Framework: Ratel Beim Okapi Framework handelt es sich um eine Reihe von freien (LGPL-lizenzierten) Anwendungen und Filtern, die allerdings bis vor Kurzem ausschließlich auf der proprietären Software-Plattform Microsoft .NET liefen. Seit 30. April 2009¹⁸ wird das Okapi Framework nur mehr in Java entwickelt. Demnächst sollen alle Anwendung in Java portiert werden und auf der freien Plattform Mono laufen.

Mit Ratel – das über die grafische Benutzeroberfläche Rainbow aufgerufen werden kann – lassen sich Segmentierungsregeln im standardisierten SRX-Format bearbeiten und in die in Kapitel 4.2.1 beschriebenen Konvertierungsvorgänge einbinden. Dank der Verwendung des SRX-Formats ist dieses Tool vielseitig und ermöglicht feine Einstellungen der Segmentierungsregeln (Abbildung 4.5).

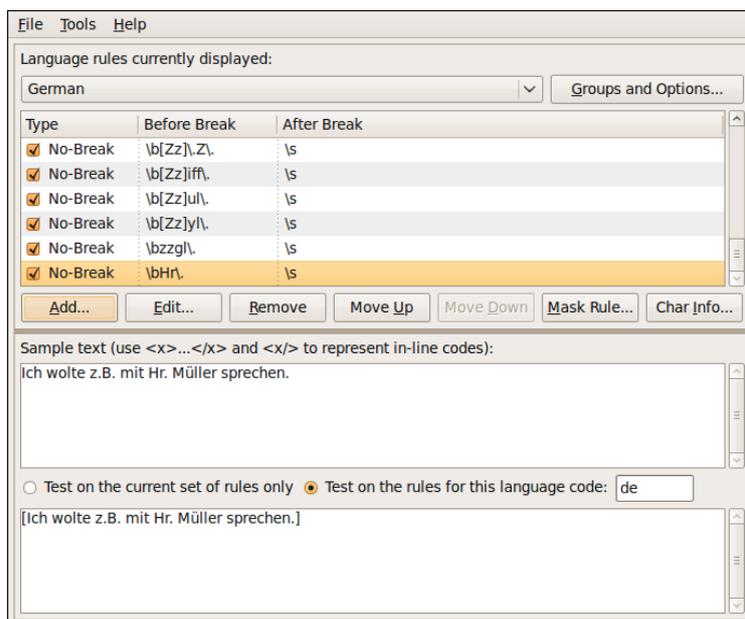


Abbildung 4.5: Segmentierungsregeln in Ratel

18 *Okapi Framework - Introduction*. URL: <http://okapi.sourceforge.net/> (besucht am 23.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYg8HP0>.

4.2 Unterstützte Dateiformate

4.2.1 Dokumentvorlagen

Auf die Frage, mit welchen Dateiformaten sie am meisten konfrontiert sind, antworteten die Studienteilnehmer wie folgt¹⁹ (Lagoudaki 2006: 15):

- Microsoft Word, Excel, PowerPoint und Reintext 96%
- PDF 43%
- XML und HTML 26%
- Hardcopy 19%
- DTP²⁰ 12%

Demnach versuchen alle TM-Systeme so viele – offene wie proprietäre – Dateiformate wie möglich zu unterstützen, um dieser Nachfrage gerecht zu werden.

OmegaT Mit Hilfe von integrierten Filtern unterstützt OmegaT folgende Formate uneingeschränkt und unmittelbar. Diese Dateien bedürfen keiner Konvertierung und können direkt in den `/source/`-Ordner abgelegt werden, wobei unterschiedliche Dateiformate gleichzeitig unterstützt werden können²¹:

- Reintext in verschiedenen Kodierungen (ASCII, UTF-8, usw.)
- OpenOffice.org (Textverarbeitung, Tabellenkalkulation, Präsentation)
- OpenDocument
- Microsoft Open XML
- HTML und XHTML
- Okapi einsprachige XLIFF-Dateien (Siehe auf der nächsten Seite).

Im OmegaT-Handbuch wird außerdem darauf hingewiesen, dass das TM-System mittels externer Programme weitere Formate unterstützt. Es handelt sich dabei um:

- Proprietäre Microsoft Office Formate (bis *Office 2003*): die Konvertierung erfolgt über OpenOffice.org durch Öffnen in der jeweiligen Anwendung und anschließendes Speichern. Bei umfangreichen Projekten mit zahlreichen einzelnen MS-Office-Dateien kann

¹⁹ Mehrfachauswahl war möglich.

²⁰ Proprietäre Desktop-Publishing-Formate (FrameMaker, Illustrator, InDesign, QuarkXPress, PageMaker, usw.)

²¹ So können sich beispielweise HTML- und ODT- (OpenOffice.org Dokumentformat) Dateien im gleichen Ordner unter `/source/` befinden.

die Konvertierung mit Hilfe des Skripts mso2ooo (siehe auf der nächsten Seite) im Batchverfahren erledigt werden.

- XLIFF: die Konvertierung erfolgt über Anwendungen und Filter des Okapi Framework.
- Trados TagEditor TTX- und Wordfast Professional TXML-Dateien: diese proprietären Formate lassen sich mittels Toxic (Siehe auf der nächsten Seite) einbinden.

Okapi Framework: Rainbow Mit Hilfe der grafischen Benutzeroberfläche, die das Programm Rainbow bietet, lassen sich verschiedenartige Dateioperationen durchführen. Eine davon ist die Konvertierung zwischen zahlreichen Formaten:

IDML Für Adobe InDesign IDML-Dateien; Alphaversion

MIF Für Adobe FrameMaker Interchange Format; Pre-Alphaversion

MsOffice/OOXML – Für Microsoft Office 2007 Dateien (DOCX, PPTX und XLSX)

OpenOffice/ODF - Für OpenOffice.org Dateien (ODT, ODP, ODS und ODG)

PlainText – Für Reintext-Dateien

Table - Für Tabellen-ähnliche Dateien wie CSV, tab-delimited Dateien, usw.

TMX - Für TMX Dateien

XLIFF - Für XLIFF Dateien

XML - Für XML Dateien



Generic XLIFF: Alle zu übersetzenden Dateien werden ins XLIFF-Format konvertiert und in ein Projekt verpackt. Die Übersetzung erfolgt anschließend mit Hilfe eines beliebigen XLIFF-Editors

OmegaT: Erzeugt ein OmegaT-kompatibles Projekt mit fertiger Ordnerstruktur

Original + RTF Layer: Alle zu übersetzenden Dateien werden in eine RTF-Datei mit Trados-kompatiblen Formatvorlagen konvertiert.

Hierzu wird folgendermaßen vorgegangen:

1. In Rainbow wird die Adresse des lokalen Ordners, in dem sich die zu konvertierenden Dateien befinden, als Wurzeladresse angegeben und die Dateien zur Bearbeitungsliste hinzugefügt.
2. Über den Menüpunkt **Utilities > Translation Package Creation...** wird die Konvertierungsfunktion aufgerufen. Es stehen zahlreiche Optionen zur Verfügung:
 - a) Projekttyp (Generic XLIFF, OmegaT-XLIFF, Original + RTF)
 - b) Einbinden von Satzsegmentierungsregeln (Abbildung 4.6 auf der nächsten Seite)
 - c) Vorübersetzen²²
3. Über die Schaltfläche **OK** wird die Konvertierung gestartet.

²² Der Vorgang des Vorübersetzens wird in Kapitel 4.3 auf Seite 38 erklärt.

4. Anschließend stehen die fertig konvertierten Dateien in einem neu angelegten Ordner, dessen Adresse und Name einstellbar sind, zur Verfügung.

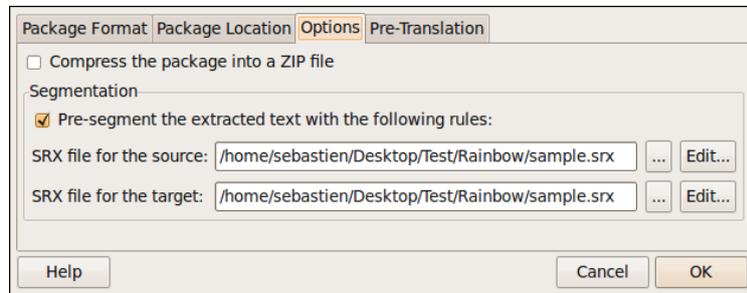


Abbildung 4.6: Optionen bei der Erstellung eines Projektes in Rainbow

Toxic Toxic ist ein von Marc Prior, einem OmegaT-Entwicklungsteammitglied, in tcl/tk geschriebenes, GPL-lizenziertes Programm mit grafischer Benutzeroberfläche, das die Konvertierung zwischen Trados TagEditor TTX-, sowie Wordfast Professional TXML-Dateien in ein von OmegaT lesbares Zwischenformat (TOX) übernimmt:

»Toxic« stands for Trados-OmegaT-eXchange. The purpose of the Toxic utility is to enable users of OmegaT to receive and deliver files in the Trados Tag Editor TTX format. Tradox TTX files are produced in Trados from a number of initial source-text formats. Toxic should therefore extend the range of file formats that can be handled in OmegaT. In addition, it enables OmegaT to be used in customer workflows which require Tag Editor. Toxic can also be used to translate files in the Wordfast Professional TXML format.« (Prior 2009)

mso2ooo mso2ooo ist ein 2006 von Leontius Adhika Pradhana in Python/OpenOffice-Makro geschriebenes Skript, das es ermöglicht, mehrere Microsoft Office Dateien, die im älteren proprietären Format vorliegen (DOC, XLS, PPT), im Stapelverfahren in das entsprechende OpenOffice.org Format (ODT, ODS, ODP) zu konvertieren. Dies ist besonders bei umfangreichen Projekten interessant, da das händische Konvertieren mühsam und zeitraubend werden kann.

Anaphraseus Bedingt durch seine Einbindung in OpenOffice Writer unterstützt Anaphraseus lediglich die Formate, die das freie Textverarbeitungsprogramm selbst unterstützt.

Zusammenfassung Sowohl OmegaT als auch Anaphraseus sind im Stande, zahlreiche Dateivorlagen zu öffnen und für die Übersetzung vorzubereiten, wobei Ersteres ein breiteres Spektrum abdeckt.²³ Es zeigt, wie sehr die jeweilige Entwicklergemeinschaft bemüht ist, Unterstützung für möglichst viele, teilweise als »Standards« etablierte Formate anzubieten und somit eine größere Unabhängigkeit von proprietären Tools und Formaten zu gewährleisten.

Trotz dieser Bemühungen werden PDF-Dateien von keiner Anwendung unterstützt. Je nach Qualität der Vorlage können dennoch CAT-bearbeitbare Texte mit Hilfe von Texterkennungsoftware (OCR) bzw. von Konvertierungsskripten aus PDF-Dateien extrahiert werden. Da diese Vorgänge nicht Gegenstand dieser Arbeit sind, wird hier lediglich auf zwei freie, plattformunabhängige Tools verwiesen: GOOCR (Texterkennung) und Pdftotext (Konvertierung von PDF- zu Reintext-Dateien).

4.2.2 Translation Memory Dateiformate

LISA-Standard: TMX Auf der Seite der LISA finden sich Empfehlungen für den Gebrauch eines Formatstandards für TM-Dateien: den TMX. So ist dort zu lesen:

»TMX (Translation Memory eXchange) is the vendor-neutral open XML standard for the exchange of Translation Memory (TM) data created by Computer Aided Translation (CAT) and localization tools. The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process. In existence since 1998, TMX is a certifiable standard format. TMX is developed and maintained by OSCAR (Open Standards for Container/Content Allowing Re-use), a LISA Special Interest Group.«²⁴

Von diesem Standard gibt es derzeit fünf Varianten (von 1.1 bis 1.4a). Grundsätzlich wird dabei unterschieden, ob die TM-Datei Formatierungsinformationen (*level 2*) oder keine derartigen Informationen (*level 1*) speichert.

OmegaT Intern verwendet OmegaT ein TM-Format, das sehr stark am TMX-Standard (*level 2*) angelehnt ist: das OmegaT-TMX-Format.

TM-Dateien, die von OmegaT erstellt wurden, können problemlos ins aktuelle Übersetzungsprojekt eingebunden werden. Des Weiteren unterstützt OmegaT TM-Dateien, die von

²³ Unerwähnt geblieben sind Software- und Skripttextdateien (wie Javascript), mit denen vor allem Software- bzw. Webitelokalisierer arbeiten. Für diese Berufsgruppe bietet jedoch OmegaT ebenfalls zahlreiche Möglichkeiten an.

²⁴ Localization Industry Standards Association (LISA): *Translation Memory eXchange (TMX)*. URL: <http://www.lisa.org/Translation-Memory-e.34.0.html> (besucht am 27.07.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXZU85X>.

anderen TM-Systemen erstellt wurden, sofern diese dem TMX-Standard (1.1 bis 1.4a, *level 1* oder *2*) entsprechen. Allerdings wird in der Bedienungsanleitung darauf hingewiesen, dass *level 2*-Dateien nicht vollständig von OmegaT unterstützt werden, und dass dies möglicherweise Konsequenzen auf die Trefferquote bei der TM-Durchsuchung haben kann. Somit ist es ratsam, TM-Dateien zu importieren, die keine Formatierungsinformationen beinhalten.²⁵

TM-Dateien werden auf einfache Weise importiert: Es genügt, sie in den Projektunterordner `/tm/` abzulegen und das Projekt neu zu laden. Interessanterweise können beliebig viele TMX-Dateien in diesen Ordner abgelegt werden: OmegaT durchsucht sie alle gleichzeitig nach möglichen Treffern. Sollte eine zu importierende TM-Datei fehlerhaft sein oder unnötige Formatierungsinformationen beinhalten, kann dies mit Hilfe der Anwendung Okapi Olifant korrigiert werden.

Anaphraseus Als Open Source-»Klon« von Wordfast verwendet Anaphraseus das Wordfast-eigene TXT-Format (Unicode-Format UTF-16, Wordfast Version 8) für die interne TM.

Anaphraseus kann TM-Dateien im standardisierten TMX- und dem OmegaT-TMX-Format importieren. Je nach Umfang der zu importierenden TM-Datei kann der Vorgang allerdings einige Minuten in Anspruch nehmen. In diesem Fall ist die Verwendung der Import/Exportfunktion von Okapi Olifant zu empfehlen.

Okapi Olifant Bei Olifant handelt es sich um eine freie Anwendung aus der Reihe des bereits erwähnten Okapi Framework, die über umfangreiche Funktionen zum Anlegen, Ändern, Pflegen und Konvertieren von TM-Dateien verfügt. Allerdings ist sie noch nicht auf Java portiert worden und läuft derzeit nur unter MS Windows.

Olifant kann nahezu alle Typen von TM-Dateien (TMX 1.1 bis 1.4b *level 1* und *2*, Wordfast TXT) öffnen, importieren sowie bearbeiten und benötigt dafür nur wenige Sekunden. Nach erfolgreichem Laden der TM-Datei können die darin enthaltenen Formatierungszeichen geändert oder entfernt werden (siehe Abbildung 4.7 auf der nächsten Seite). Dadurch kann auf einfache Weise aus einer *level 2*-TMX-Datei eine *level 1* erstellt werden, die bestimmte TM-Systeme wie OmegaT besser erkennen.

Des Weiteren kann mit Hilfe von Olifant eine TMX-Datei in eine Wordfast-lesbare TXT-Datei konvertiert werden und umgekehrt: Dies ist gerade für deren Verwendung durch Anaphraseus interessant, da die in diese Anwendung integrierte Konvertierungsfunktion (TMX bzw. OmegaT-TMX zu Wordfast-TXT) wie bereits erwähnt recht viel Zeit beanspruchen

²⁵ Vgl. Jean-Christophe Helary u. a.: *OmegaT 2 User Manual*. Hilfedatei zum Programm

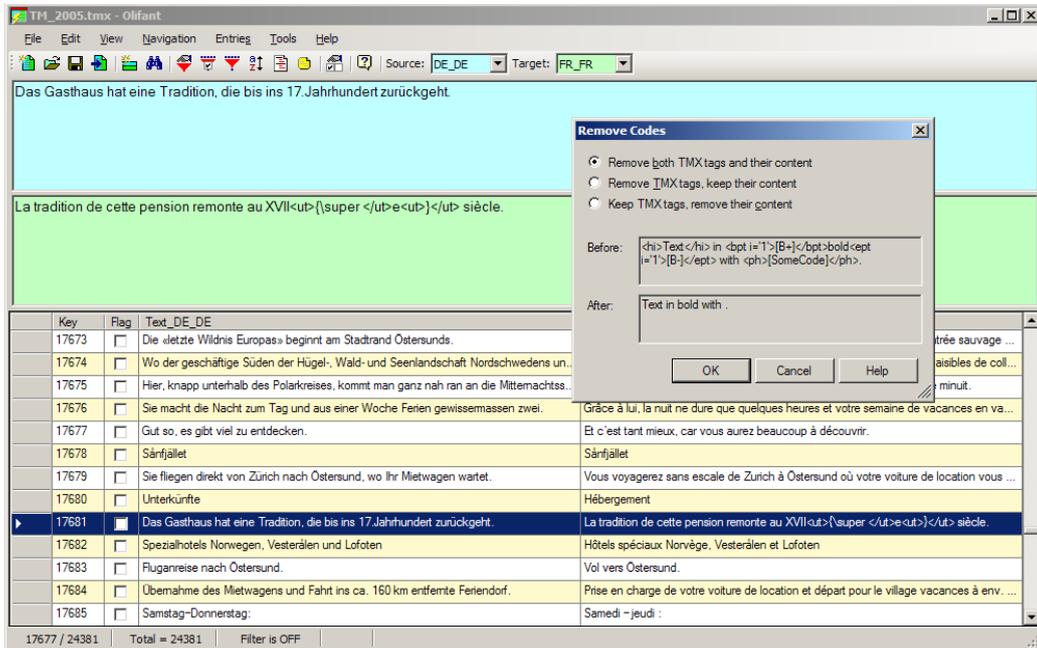


Abbildung 4.7: Hauptfenster von Olifant mit Dialogfenster zum Entfernen von TMX-Tags

kann²⁶.

4.2.3 Terminologiedatenbanken

Sind eine oder mehrere terminologische Datenbanken verfügbar, stellen sie wichtige Informationsquellen für den Übersetzer dar. Einige TM-Systeme verfügen über ein integriertes Terminologie-Management-Modul, das während des Übersetzungsvorgangs Termini, die sich im gerade bearbeiteten Segment befinden, in den verfügbaren terminologischen Datenbanken automatisch nachschlägt und dem Übersetzer Vorschläge unterbreitet. Umgekehrt bieten diese Module oft auch die Möglichkeit an, Termini, die sich im Ausgangstext befinden, in die Terminologiedatenbank mit deren Übersetzung sowie zusätzliche Informationen über eine einfache Tastenkombination zu speichern.

LISA-Standard: TBX Genauso wie für den Austausch von TM-Dateien hat die LISA einen Dateistandard für die Terminologie-Datenbanken anerkannt:

²⁶ Nach einer Reihe von Tests stellte der Verfasser dieser Arbeit fest, dass Olifant TMX-Dateien zwar ins Wordfast-TXT-Format exportieren kann, dass es sich bei besagtem Wordfast-Format allerdings um die veraltete Version 5 handelt. Damit Anaphraseus die über Olifant exportierten TM-Dateien lesen kann, ist es erforderlich, eine kleine Änderung in der ersten Zeile der Datei vorzunehmen: Es müssen fünf und nicht nur vier TAB-Zeichen nach dem letzten Attribut stehen.

»Term Base eXchange (TBX) is the open, XML-based standard for exchanging structured terminological data that has been approved as an international standard by LISA and ISO. It will soon be published by ISO as ISO 30042.«²⁷

Leider wird dieses Standardformat weder von OmegaT noch Anaphraseus unterstützt.

OmegaT Die Terminologie-Management-Funktion von OmegaT ist sehr einfach gehalten. Entsprechend einfach ist auch das von diesem TM-System verwendete Dateiformat: Es handelt sich um eine Reintextdatei, die folgende Voraussetzungen erfüllen muss, um vom System erkannt und verwendet zu werden:

- Die Datei muss in UTF-8 kodiert sein und die Endung `.utf8` haben.
- Die Termini müssen in zwei von TAB-Zeichen voneinander getrennten Spalten geordnet sein:
 - Die erste Spalte enthält die Termini in der Ausgangssprache.
 - Die zweite Spalte enthält die entsprechenden Übersetzungen.
 - Optional kann eine dritte Spalte für zusätzliche Informationen (Grammatik, Definition, Kontext, u.v.m.) verwendet werden.
- Diese Datei muss in den Projektunterordner `/glossary/` abgelegt werden. Dabei können sich gleichzeitig mehrere Glossardateien in diesem Ordner befinden: Sie werden alle vom TM-System berücksichtigt.

Abbildung 4.8 stellt ein solches OmegaT-kompatibles Glossar dar.

Druckknopf	Bouton poussoir
Druckluft	Air comprimé
Druckluftbremsen	Freins pneumatiques
Druckluftdüse	Buse de transport pneumatique
Druckluftelement	Composant pneumatique
Druckluftpresse	presse pneumatique
Druckluftpumpe	Pompes pneumatiques
Druckluftventil	Vanne à commande pneumatique
Druckluftwinde	Treuil à air comprimé
Druckreduzierventil	Détendeurs
Druckregelgerät	Appareil de réglage de la pression
Druckregler	Régulateur de pression
Druckschalter	Interrupteur à poussoir
Druckventil	Valve de pression
Druckübersetzer	Multiplicateur de pression

Abbildung 4.8: Glossardatei in OmegaT

Neben diesen formalen Kriterien ist ebenfalls zu beachten, dass die Termini in der Grundform (Verben im Infinitiv, Substantive im Singular) und ohne Artikel oder Personalpronomen

²⁷ Localization Industry Standards Association (LISA): *Term Base eXchange (TBX)*. URL: <http://www.lisa.org/Term-Base-eXchange.32.0.html> (besucht am 27.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZDKD7k>.

gespeichert werden müssen, damit sie vom System korrekt erkannt werden.²⁸

Anaphraseus Das Datenformat der von Anaphraseus erkannten Glossare entspricht genau demjenigen von OmegaT: Es handelt sich ebenfalls um eine *Tab-Separated Values (TSV)*-Datei, mit dem einzigen Unterschied, dass die Datei mit `.txt` enden muss. Zu Beginn der Übersetzung wird das Glossar über den Reiter `Glossary` im Projektfenster ausgewählt (siehe Abbildung 4.9).

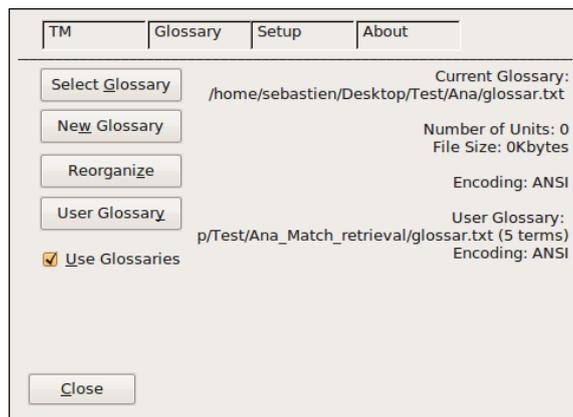


Abbildung 4.9: Dialogfenster zum Auswählen der Glossardatei in Anaphraseus

Zusammenfassung Die Tatsache, dass OmegaT und Anaphraseus das gleiche Dateiformat für ihre Glossare verwenden, erleichtert zwar den Austausch von terminologischen Daten zwischen beiden TM-Systemen, kann aber zu größeren Schwierigkeiten beim Austausch mit weiteren Systemen oder beim Importieren von fremden Terminologie-Datenbanken führen. Wünschenswert wäre die Verwendung des TBX-Standards, obwohl dies – zumindest bei OmegaT – nicht zu den dringendsten Verbesserungen gehört: So findet sich im OmegaT-Feature-Request-Tracker folgende Anfrage von Jean-Christophe Helary, die mit der Dringlichkeitsstufe »5«²⁹ versehen wurde, sowie ein anonymer Benutzerkommentar:

»TBX has been submitted to the ISO in February. Lisa is also working on a lite version (not finalized yet). Isn't it about time to offer some kind of import support for such formats? [...]

[Anonymer Kommentar vom 15. 05. 2009:] I really wish OmegaT had TBX support. I have terminology for many different clients plus general terminology and I want to be

28 Die Funktionsweise des Terminologie-Moduls von OmegaT wird in Kapitel 5.3.1 auf Seite 57 und dessen Leistungsfähigkeit in Kapitel 7.1.4 auf Seite 79 beschrieben.

29 Auf einer Skala zwischen 1 (nicht dringend) und 9 (sehr dringend).

able to use my TBX editor to manage my terms and then view the terms in OmegaT as easily as now with the tab-delimited glossaries.«³⁰

Freie bzw. Open Source Tools zum Konvertieren zwischen dem OmegaT-TSV- und dem TBX-Format sind nicht bekannt.

4.2.4 Integrierte Wörterbücher

Die von Lagoudaki befragten Übersetzer stufen Wörterbücher – insbesondere auf CD-ROM oder online – als wichtigste Nachschlagequellen (vgl. Lagoudaki 2006: 17; 2008: 188) ein. Es ist daher berechtigt zu fragen, inwiefern und wie TM-Systeme Wörterbücher integrieren.

OmegaT In OmegaT lassen sich GPL-lizenzierte Stardict-Wörterbücher, die online herunterzuladen und auf der eigenen Festplatte zu speichern sind, einbinden. Dazu genügt es, die entsprechenden Wörterbuchdateien in den Projektunterordner `/dictionary/` abzulegen. Auf der Stardict-Projektseite³¹ stehen zahlreiche ein- und zweisprachige Wörterbücher und Lexika zur Auswahl. Das Angebot beschränkt sich allerdings auf Werke, die für den professionellen Übersetzer – insbesondere im technischen Bereich – wenig interessant sind: Es sind ältere Auflagen, die zu wenige Stichwörter beinhalten oder zu allgemeinsprachlich sind.

Anaphraseus In Anaphraseus heißen Wörterbücher »Glossare«. Sie unterscheiden sich von den eigentlichen, benutzer- oder projektspezifischen Glossaren dennoch darin, dass man sie während des Übersetzungsvorgangs durch Hinzufügen/Löschen/Ändern von Stichwörtern nicht modifizieren kann. In Hinblick auf das Dateiformat müssen Wörterbücher dieselben formalen Kriterien erfüllen wie Glossare (siehe Seite 36).

Zusammenfassung In OmegaT und Anaphraseus sind »Wörterbücher« mehr oder minder einfache Wortlisten. Durch die beschränkte Auswahl an Werken sind sie außerdem nicht unbedingt sehr hilfreich. In dieser Hinsicht wäre eine Integration von Wörterbüchern aus den großen Verlagshäusern über eine eigens zu diesem Zweck zu entwickelnde Schnittstelle wünschenswert. So drückte eine Studienteilnehmerin ihre Erwartungen aus: » *I would very*

³⁰ Jean-Christophe Helary (2007): *SourceForge.net: OmegaT - multiplatform CAT tool: Detail: 1756560 - Supported glossary file formats (TBX, CSV, TSV etc)*. URL: http://sourceforge.net/tracker/index.php?func=detail&aid=1756560&group_id=68187&atid=520350 (besucht am 03. 09. 2009). Feature-Request-Tracker von OmegaT auf SourceForge.net: Feature-Request vom 19. 07. 2007. Archivierte URL verfügbar: <http://www.webcitation.org/5jsZ0uTQt>.

³¹ <http://stardict.sourceforge.net/index.php> (besucht am 5. 09. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsdiCWke>.

much like to be able to centralize the content of all the dictionaries I have purchased and of glossaries available free on internet, in order to query against them as I translate and have the possibility to insert translation of terms in my TM environment. <« (Lagoudaki 2008: 187). Von Seiten der Freie Software-Gemeinschaft ist jedoch eine derartige Neuerung aus rechtlichen Gründen nicht zu erwarten, da die von den Verlagshäusern zur Verfügung gestellten Softwares proprietär sind.

4.3 Vorübersetzen

Nachdem das entsprechende Übersetzungsprojekt angelegt, die Ausgangsdatei(en) im Bedarfsfall konvertiert und die aus früheren Projekten oder vom Auftraggeber zur Verfügung gestellte(n) TM-Datei(en) erfolgreich importiert wurden, ist es oft von großem Nutzen, das Ausgangsmaterial mit Hilfe der vorhandenen Translation Memory(ies) vom TM-System automatisch »übersetzen« zu lassen. Diese sogenannte »Vorübersetzen«-Funktion stellt sogar bei manchen TM-Systemen die einzige Möglichkeit dar, einen vollständigen Projektbericht – wie in Kapitel 4.4 auf Seite 40 beschrieben – zu erstellen (vgl. Massion 2005: 128).

Diese Funktion ist nicht mit maschineller Übersetzung, wie sie von Hutchins und Somers definiert wird, gleichzusetzen, obwohl das Programm automatisch zu »übersetzen« scheint. Eine Beschreibung dieser Funktion liefert Bowker:

»Most TM systems also allow for batch translation, sometimes referred to as pre-translation, which means that a user can run a complete source text through the system and, whenever it finds an exact match, it will automatically replace the new source-text segment with the translation that is stored in the TM. Segments for which no [exact] match is found must later be translated by either a human translator [...] or a machine-translation system [...]. In either case, the entire text must then be post-edited by a human translator to ensure that the replacements made by the system were correct.« (Bowker 2002: 112)

OmegaT Das Vorübersetzen in OmegaT ist eine Funktion, die zur Zeit noch recht umständlich ist: Sie ist nicht über die grafische Benutzeroberfläche ausführbar und wird in der Bedienungsanleitung nur spärlich beschrieben. Dies hängt wahrscheinlich damit zusammen, dass es sich bei dieser Funktionalität um eine kürzlich erfolgte Implementierung³² handelt. Die dazu notwendigen Schritte sind folgende:

³² Seit der Betaversion 2.0.1, die am 16. 02. 2009 freigegeben wurde.

1. Über die grafische Benutzeroberfläche wird ein Übersetzungsprojekt angelegt, das die zu übersetzenden Dateien enthält. Anschließend wird das Programm geschlossen.
2. Die in Frage kommende TM-Datei wird in `project_save.tmx` umbenannt und an Stelle der gleichnamigen, von OmegaT erstellten TM-Datei, in den Projektunterordner `/omegat/` abgelegt. Um bessere Ergebnisse zu erzielen, empfiehlt es sich, eine *level 1* TMX-Datei zu verwenden, da die von System zu System unterschiedlichen Formatierungszeichen eine Minderung der Trefferqualität zur Folge haben kann.³³
3. Abschließend wird OmegaT in der Kommandozeile über folgenden Befehl gestartet:

```
java -jar OmegaT.jar <project-dir> --mode=console-translate
```

Dabei ist »<project-dir>« durch den vollständigen Pfad zum Projekthauptordner zu ersetzen.

Bei diesem Vorgang ersetzt OmegaT alle Segmente des Ausgangsmaterials, die als *exact match* in der TM gefunden werden, und erzeugt alle Zieldateien automatisch.

Anaphraseus Im Gegensatz zu OmegaT ist es mit Anaphraseus recht einfach, den Ausgangstext vorübersetzen zu lassen. Nachdem die in Frage kommende TM, die in einem Wordfast-lesbaren Format vorliegen muss, geladen wurde, genügt ein Klick auf den Menüpunkt `Translate 100% matches`, um die Funktion zu starten. Das Vorübersetzen beginnt automatisch und hält beim ersten Segment an, das kein *exact match* ist. Diese Funktion kann allerdings, je nach Größe der geladenen TM, recht viele Computerressourcen beanspruchen und ist bei weitem nicht so schnell abgeschlossen wie bei OmegaT. Außerdem müssen – im Gegensatz zum OmegaT-Batchmodus, bei dem alle im Projektordner befindlichen Dateien behandelt werden – die Ausgangsdateien einzeln vorübersetzt werden.

Okapi Framework: Rainbow Obwohl dieses Programm bei der Erstellung von Übersetzungsprojekten (Konvertierung) über eine Vorübersetzen-Option verfügt, sind alle diesbezüglichen Tests des Verfassers fehlgeschlagen. Hierzu fehlt es in der Programmbeschreibung an näheren Informationen über das verwendete TM-Format »SimpleTM«. Außerdem wird der Benutzer darauf hingewiesen, dass es sich hierbei um ein provisorisches TM-Managementsystem handelt, das künftig durch ein leistungsfähigeres ersetzt werden soll³⁴.

³³ Die Bearbeitung und Pflege von TM-Dateien wird in Kapitel 6.3 auf Seite 71 behandelt.

³⁴ Vgl. *Okapi Framework - Rainbow*. URL: <http://okapi.opentag.com/help/applications/rainbow/index.html> (besucht am 13.09.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYpcVjx>.

4.4 Statistik

Die statistische Analyse des Ausgangsmaterials gehört zu den wichtigsten Funktionen von TM-Systemen. Dazu schreibt Lagoudaki (2008: 164):

»An integrated analysis function should be able to identify the translatable from the non-translatable parts of a text and report on:

- wordcount for translatable text (for each file separately and for the entire project in total)
- external matches (full and fuzzy) with the associated resources (for each file separately and for the entire project in total)
- internal matches (how many segments are repeated within the same document and/or within the project)
- number of images contained in the project which may contain translatable text (translators have reported that they often forget to translate images embedded in a document as their TM system skips them as untranslatable material)«

Massion erkennt ebenfalls die Wichtigkeit dieser Funktion und beschreibt sie wie folgt:

» [Mit der statistischen Analyse des zu übersetzenden Textes] soll der Arbeitsaufwand geschätzt, Angebote kalkuliert und die Leistung abgerechnet werden. Bei dieser Analyse wird zum einen der zu übersetzende Text mit dem Translation Memory verglichen und zum anderen wird geprüft, ob innerhalb eines Textes Segmente sich wiederholen.«
(Massion 2005: 137)

Für den Übersetzer ist es also wichtig, im Vorfeld zur Übersetzungsarbeit einzuschätzen, inwieweit die Translation Memory(ies), über die er verfügt, ihm dabei hilfreich sein wird (werden).

Des Weiteren setzt sich bei immer mehr Auftraggebern die Praxis durch, Übersetzungsleistungen nach gestaffelten Tarifen und auf der Grundlage eines Analyseberichts zu bezahlen, der die Segmente (oder Wörter) in verschiedene Trefferkategorien – als Ergebnis eines Datenabgleichs zwischen der mitgelieferten TM und dem zu übersetzenden Material – einordnet. So stellte ein Benutzer in der OmegaT-Mailing-Liste folgende Frage:

»What if the agency uses scales like these:

100% & Repetition (proofread needed) $0.10 \times$ base rate

95-99% $0.4 \times$ base rate

85-94% $0.66 \times$ base rate

Below 84% $1 \times$ base rate

Will it be possible to account the matches using OmegaT?³⁵

Diese Frage ist bezeichnend für eine Entwicklung, die auf die größere Verbreitung der TM-Systeme zurückzuführen ist: Wenn Übersetzer durch den Einsatz eines TM-Systems ihre Produktivität erhöhen können, so erwarten einige Auftraggeber, dass sich dieser Produktivitätsgewinn in einer Senkung der Übersetzungskosten niederschlägt.

OmegaT OmegaT bietet Analysefunktionen in drei Formen an: Nach jedem Laden des Übersetzungsprojektes wird ein Statistikfenster angezeigt, das zugleich zum Navigieren innerhalb des Projektes dient (Abbildung 4.10a). Des Weiteren wird eine Textdatei in einem Projektunterordner angelegt (/omegat/project_stats.txt) und laufend aktualisiert (Abbildung 4.10b). Eine dritte Statistikanzeige befindet sich im unteren Fensterbereich.

Filename	Number of Segments
Island_ODT/46+47_Auftakt Aktivferien.odt	29
Island_ODT/12+13_Hotels in Reykjavik.odt	103
Island_ODT/76+77_Trekking Illulissat+Eqi.odt	138
Island_ODT/50+51_Trekking+Reittour.odt	155
Island_ODT/04+05_Auftakt Kontiki als Partner.odt	32
Island_ODT/US2_Wir sind der Norden.odt	9
Island_ODT/21_Cruise.odt	59
Island_ODT/26_Mietwagenrundreisen.odt	77
Island_ODT/US4_Übersicht.odt	37
Island_ODT/28+29_Fosshotel.odt	79
Island_ODT/24+25_Auftakt Mietwagenrundreisen.odt	34
Total number of segments	4,038
Number of unique segments	3,010
Translated unique segments	2,013

You can find detailed statistic information in the file:
/home/sebastien/Desktop/Test/OmT/Pretranslate/omegat/project_stats.txt

(a) Statistik- und Navigationsfenster

```

Project Statistics

Total:
  Segments      4038
  Words      27948
  Characters (without spaces) 171868
  Characters (including spaces) 195667

Remaining:
  Segments      1203
  Words      7098
  Characters (without spaces) 40804
  Characters (including spaces) 46358

Unique:
  Segments      3010
  Words      23715
  Characters (without spaces) 144288
  Characters (including spaces) 164859

Unique Remaining:
  Segments      997
  Words      6176
  Characters (without spaces) 35922
  Characters (including spaces) 40833

Individual File Statistics:
File Name      Total Words      Remaining Words Total Characters (without
spaces) Remaining Characters (without spaces) Total Characters
(including spaces) Remaining Characters (including spaces)
Island_ODT/04+05_Auftakt Kontiki als Partner.odt      370      19
2326      101      2663      115
Island_ODT/06+07_Auftakt Island.odt      449      29      2708      239
3116      262
Island_ODT/08+09_Auftakt Reykjavik.odt      406      53      2583      375
2953      419
Island_ODT/10+11_Entdeckungsreise Island.odt      811      253      5189
1607      5910      1817

```

(b) Projektbericht (Auszug)

Abbildung 4.10: Statistische Analyse in OmegaT

Die Analyse-Funktion liefert Informationen zum gesamten Umfang des Ausgangstextes in verschiedenen Einheiten: Segmente, Wörter, Anschläge ohne/mit Leerzeichen³⁶. Allerdings weicht die Analysefunktion in OmegaT von den in Lagoudakis Studie geforderten

35 José Raeiro (2008): *Re: [OmT] Compatibility with Trados*. URL: <http://tech.groups.yahoo.com/group/OmegaT/message/11039> (besucht am 31. 08. 2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 02. 07. 2008. Archivierte URL verfügbar: <http://www.webcitation.org/5jsXw0Nc>.

36 Letztere Einheit ist für die im deutschsprachigen Raum übliche Verrechnung in DIN-Zeilen (55 Anschläge mit Leerzeichen) hilfreich.

Funktionalitäten insofern ab, als sie Wiederholungen nur innerhalb des Gesamtprojektes berücksichtigt. Zudem werden Bilder nicht gezählt, wie aus Tabelle 4.1 zu entnehmen ist.

Analysefunktionen	Projektweit	Einzelne Dateien
Gesamtumfang Ausgangstext	ja	ja
»internal matches« (Wiederholungen)	ja	nein
»external matches« (TM-Matches)	ja ³⁷	ja ³⁷
Bilder	nein	nein

Tabelle 4.1: Analysefunktionen in OmegaT

Dass Bilder – mit übersetzbarem Inhalt – mitgezählt werden sollten, ist zwar eine wünschenswerte Verbesserung, die aber derzeit von keinem TM-System unterstützt wird. Des Weiteren sind statistische Werte über die in der/den TM möglicherweise vorhandenen 100%- bzw. *exact matches* nicht unmittelbar verfügbar: Dazu bedarf es eines ersten Durchlaufs durch die Vorübersetzen-Funktion.

Interessanterweise liefert OmegaT Informationen zum Projektfortschritt. Dies kann nützlich sein, um sich einen Überblick über die noch zu leistende Übersetzungsarbeit zu verschaffen (siehe Kapitel 5.2.3 auf Seite 56).

Anaphraseus Anaphraseus verfügt derzeit über keine nennenswerte Statistikfunktion, die über das einfache Zählen der Anschläge (mit und ohne Leerzeichen) des aktuellen Dokuments hinausgeht. Diese Daten werden vom Hauptprogramm Writer ermittelt und in einem Anaphraseus-Dialogfenster angezeigt. Wahlweise kann diese simple Statistikfunktion im Writer-Menü **Tools > Wordcount** abgerufen werden. Die Implementierung einer Prä- und Postprozessanalyse ist jedoch geplant (vgl. Medvedev 2008).

Zusammenfassung Wünschenswert wäre bei OmegaT, dass die statistischen Werte über die *exact matches* gleich ohne den Umweg über das Vorübersetzen zur Verfügung stehen.

Außerdem stellt man in den Analyseergebnissen, je nachdem, welches TM-System verwendet wird, erhebliche Unterschiede fest, wie der Test in Kapitel 7.2 auf Seite 80 zeigt.

³⁷ Die Ermittlung der *exact matches* erfolgt erst nach einem Durchlauf des Ausgangsmaterials durch die Vorübersetzen-Funktion im Batchmodus, wie in Kapitel 4.3 beschrieben. Somit ist dieser Wert nicht unmittelbar verfügbar.

4.5 Alignment

Verfügt der Übersetzer zu Beginn eines Projektes über keine TM, sondern über Textmaterial zum Thema des Projektes in zwei Sprachen (frühere Versionen der zu übersetzenden Dokumente, zweisprachige Textkorpora, Paralleltexte mit deren Übersetzungen, usw.) kann er dieses Material – vorausgesetzt, es liegt in maschinenlesbarer Form vor – mit Hilfe eines Computertools »alinieren« und daraus eine TM-Datei erstellen. Somers liefert eine Definition dieser Funktion und nennt drei Faktoren, die diesen Vorgang erschweren:

»Alignment involves matching up the source text and the translation segment by segment into translation pairs. [...] If the translation is straightforward, then so is the alignment. But three factors can make alignment more difficult than it at first seems: one is the difficulty of accurately recognizing where sentences begin and end; the second is the fact that – depending on the language pair – a single sentence in one language may not necessarily correspond to a single sentence in the other language; the third factor is that translators may more or less freely change the relative order of sentences in the translation.« (Somers 2003: 34)

Die erste Schwierigkeit hängt somit mit der in Kapitel 4.1.4 behandelten Segmentierungsproblematik zusammen: Beim Alignment soll das Tool im Stande sein, Segmentierungsregeln zu berücksichtigen. Sind zudem die Textstrukturen nicht identisch (zweite und dritte von Somers genannte Schwierigkeiten), muss es beim Alignment möglich sein, Anpassungen vorzunehmen.

Weder OmegaT noch Anaphraseus verfügen über eine Alignment-Funktion. Hierzu muss auf freie Anwendungen zurückgegriffen werden, die nicht Bestandteil dieser TM-Systeme sind.

bligner bligner ist ein einfaches Alignment-Skript, das von Didier Briel, dem jetzigen OmegaT-Release Manager, in Python/Perl geschrieben wurde. Die Bedienung ist relativ einfach und das Alignment erfolgt automatisch. Dennoch können nur einzelne Dateien aliniert werden und diese müssen strenge formale Kriterien erfüllen, damit das Skript reibungslos funktioniert:

- Original und Übersetzung müssen im Reintextformat vorliegen. Dabei ist die Kodierung (ASCII, UTF-8, usw.) irrelevant. Dies erfordert jedoch möglicherweise eine Konvertierung.
- Beide Texte müssen die genau gleiche Anzahl an Absätzen aufweisen, andernfalls bricht der Alignment-Vorgang ab und es wird keine TM-Datei erzeugt. Diese Vor-

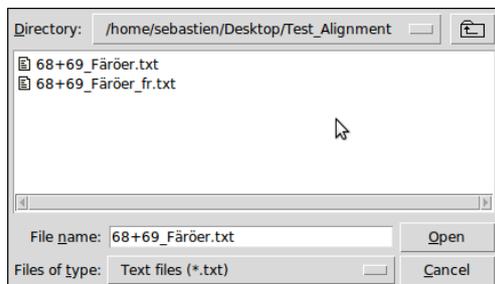
aussetzung kann unter Umständen aufwendige Korrekturarbeit an den Vorlagen bedeuten.

- Es empfiehlt sich, die mitgelieferten (rudimentären) Segmentierungsregeln gegebenenfalls zu ergänzen.

Des Weiteren verfügt dieses Skript über keine Konfigurationsdatei. Dies bedeutet, dass alle Änderungen (Ausgangs- und Zielsprache sowie Segmentierungsregeln) im Skript selbst vorgenommen werden müssen, was wenig computerverstärkte Übersetzer abschrecken mag.

Dem gegenüber stehen folgende Vorteile:

- Das Skript läßt Anpassungen wie eigene Segmentierungsregeln zu
- Die erzeugte TM-Datei ist im TMX-Format, was deren Einbindung in zahlreiche TM-Systeme erleichtert
- Das Skript bietet eine einfache grafische Benutzeroberfläche für das Auswählen der zu alinierenden Dateien
- Der Alignment-Vorgang erfolgt automatisch und erfordert – bei guter Vorbereitungsarbeit – keinen weiteren Eingriff von Seiten des Übersetzers.



(a) Grafische Benutzeroberfläche

```

36# Source language
37 lngsource = "DE-DE"
38# Target language
39 lngtarget = "FR-FR"
40# Admin language
41 lngadmin = "EN-US"
42
43# Segmentation
44# p = paragraph
45# s = sentence
46#segmenttype = "p"
47 segmenttype = "s"
48
49# Following expressions should not segment (in case of segment="s").
50# Each expressions is entered between "[" and "]", followed by a comma (",")
51# except for the last one.
52# ".", "?", "!", "(", " and ")" must be prefixed with a "\"
53# The following presets are the same as the default ("*.")
54# in OmegaT (version 1.6 RC7)
55 anosegment = [
56     ['\.\.\.', '\s+\P{Lu}'],
57     ['\.\.\.', '\s+[A-Z]'], # \P syntax doesn't seem to work in Python
58     ['etc\.', '\s+\P{Lu}'],
59     ['etc\.', '\s+[A-Z]'], # \P syntax doesn't seem to work in Python
60     ['Dr\.', '\s'],
61# You can modify the preset, or simply uncomment (deleting the #)
62# an abbreviation to add it to the list of expressions
63# ['U.S.', '\s'],
64# ['Tel\.', '\s'],
65# ['a.v.m\.', '\s'],
66# ['p.v.m\.', '\s'],
67# Some German abbreviations
68# ['ca\.', '\s'],
69# ['inkl\.', '\s'],
70# ['Min\.', '\s'],
71# ['bzgl\.', '\s'],
72# ['bzw\.', '\s'],

```

(b) Auszug aus dem zu konfigurierenden Skript

Abbildung 4.11: Benutzeroberfläche und Skript von bligner

TagAligner Eigentlich für Online-verfügbare, zweisprachige Textkorpora konzipiert, eignet sich TagAligner bedingt zum Alignment von üblichen Textdateien. Dazu ist es erforderlich,

dass die zu alinierenden Dateien in einem strukturierten Format³⁸ vorliegen. Entwickelt wurde TagAligner seit 2006 von der Transducens-Forschungsgruppe der Universität Alicante um Enrique Sánchez Villamil.

Es besteht bei diesem Tool die Möglichkeit, den Alignment-Algorithmus auszuwählen, wobei folgende Optionen zur Auswahl stehen:³⁹

- *alignment distance* (Church-Gale-Algorithmus)
- Vergleich der Textlängen
- Satz für Satz in der Reihenfolge (Standard)

Nach erfolgreichem Alignment wird eine TMX-Datei erstellt. TagAligner liefert zufriedenstellende Ergebnisse (siehe Tabelle 7.5 auf Seite 81) bei vergleichsweise geringerer Vorarbeit als mit bligner. Die mitgelieferten Konfigurations- bzw. Filterdateien ermöglichen feine Anpassungen der Segmentierung auf der Tag-Ebene. Allerdings sind Versuche, weitere Segmentierungsregeln vom Programm berücksichtigen zu lassen (damit beispielsweise kein neues Segment bei gängigen Abkürzungen entsteht), fehlgeschlagen oder die dafür benötigten Programmierkenntnisse reichten nicht aus.

Bitext2tmx Im Gegensatz zu beiden vorigen Programmen handelt es sich bei Bitext2tmx um eine Anwendung, die den Alignment-Prozess nicht automatisch erledigt. Sie verfügt über eine grafische Benutzeroberfläche in Tabellenform, wie sie von kommerziellen Anwendungen wie Trados WinAlign bekannt sind (siehe Abbildung 4.12 auf der nächsten Seite). Nachdem die zu alinierenden Dateien (im Reintextformat) importiert wurden, erfolgt der Alignment-Vorgang Satz für Satz. Dabei können Segmente zusammengefügt, getrennt oder gelöscht werden. Zum Schluss wird eine TMX-Datei erstellt. Leider sind auch bei dieser Anwendung Segmentierungsregeln nicht einstellbar, was zu unnötiger Korrekturarbeit führt. Entwickelt wurde Bitext2tmx von der Transducens-Forschungsgruppe der Universität Alicante um Susana Santos Antón.

Zusammenfassung Alignment-Tools sind sowohl für den Übersetzer als auch für den Sprachwissenschaftler, der im Bereich der Korpuslinguistik arbeitet, von Nutzen. So entstanden zu Forschungszwecken eine Reihe weiterer Tools, die über zahlreiche einstellbare Optionen verfügen. Eine detaillierte Darstellung all dieser Tools würde jedoch über den Rahmen dieser Arbeit hinausgehen.

³⁸ Für TagAligner kommen vordergründig HTML oder XHTML in Frage. Tests, bei denen versucht wurde, OpenOffice-Dateien zu alinieren, sind trotz des bereitgestellten Filters fehlgeschlagen.

³⁹ *Tagaligner: an aligner for parallel text*. URL: <http://tag-aligner.sourceforge.net/> (besucht am 03.09.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZ4qkHV>.

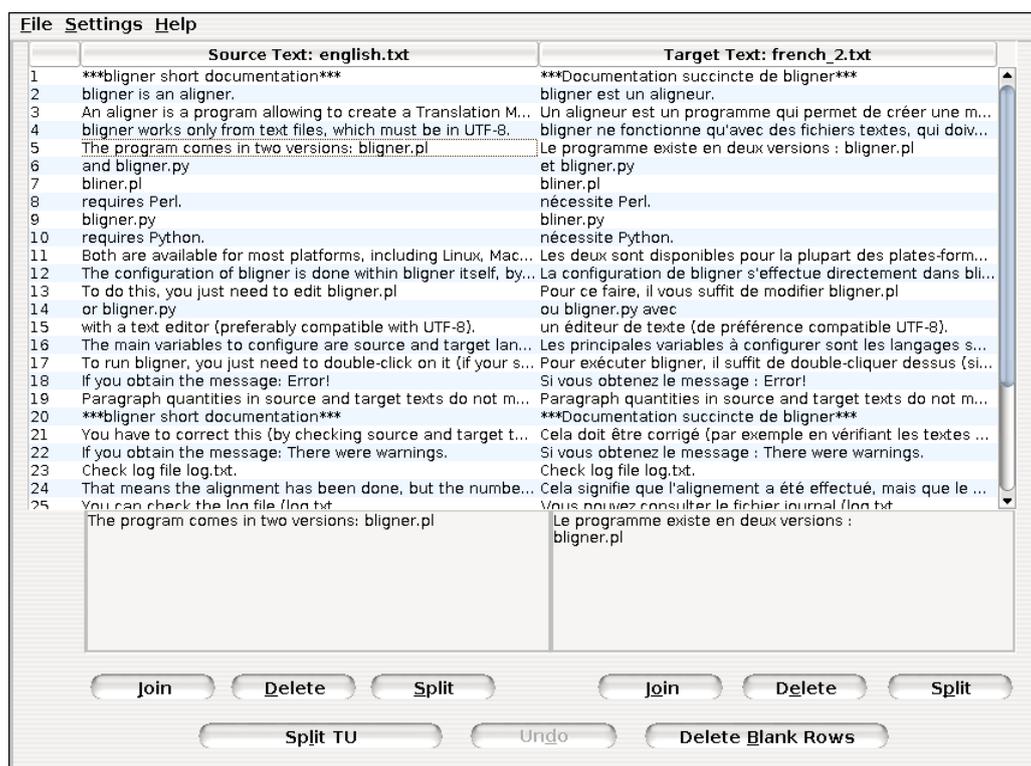


Abbildung 4.12: Hauptfenster von Bitext2tmx

Die in diesem Kapitel beschriebenen Tools ermöglichen es, mit größerem oder geringem Aufwand Originaltexte mit vorhandenen Übersetzungen zu alinieren. Im Rahmen ihrer Untersuchung stellt Lagoudaki fest, dass eine knappe Mehrheit der befragten Übersetzer dem Alignment-Modul ihrer TM-Systeme große Wichtigkeit beimisst. Dies hängt damit zusammen, dass Übersetzer ihre eigenen Übersetzungen als zuverlässige Informationsquellen betrachten (vgl. Lagoudaki 2008: 146). Außerdem bevorzugen die Studienteilnehmer einen automatischen Ablauf des Alignment-Prozesses:

»Based on the survey's findings, the expectation and general need of all surveyed TM users regarding the alignment facility is the automatic accurate alignment of pairs of texts (in any file format) at all levels (e.g. document, section, paragraph, sentence, phrase, word) or at any particular level specified by the user. Interestingly, alignment is found to be one of few TM functions for which users favour automation. This seems to derive from negative experiences with current TM systems which assist the alignment process by offering alignment ›suggestions‹ based on well-chosen parameters, but require substantial user intervention in order to verify and often repair the suggested aligned pairs.« (ebd.: 146)

Während bei Bitext2tmx Satz für Satz vorgegangen werden muss, erledigen bligner und

TagAligner die Alignment-Arbeit automatisch, wobei dies zu qualitativ unterschiedlichen Ergebnissen führt, wie der Test in Kapitel 7.3 auf Seite 80 zeigt.

Abschließend zu diesem Kapitel soll noch erwähnt werden, dass bligner optional das Alignment auf Absatz-Ebene anbietet, was dieses Programm bei einer Verwendung der erstellten TMX-Datei eher als Quelle für Kontextinformationen⁴⁰ denn als reine TM zusätzlich interessant macht.

⁴⁰ Siehe Konkordanzsuche in Kapitel 5.4 auf Seite 59.

5 Funktionale Merkmale: Während des Übersetzens

Nachdem das Übersetzungsprojekt erfolgreich angelegt wurde, beginnt für den Übersetzer die eigentliche Arbeit: das Übersetzen. Um sich dabei unterstützen zu lassen, erwartet er vom eingesetzten TM-System eine Reihe von Funktionen, die in diesem Kapitel beschrieben werden. Diese Erwartungen an Funktionalität sind nicht mit denjenigen an Benutzbarkeit oder Effizienz gleichzusetzen, die – als nichtfunktionale Merkmale – ab Seite 73 behandelt werden.

5.1 Editieren

In diesem Kapitel werden die Funktionen untersucht, die die Texteditoren der TM-Systeme auf der Eingabeseite anbieten.

5.1.1 Editortyp

Die Funktionen, die Übersetzer vom eingesetzten Editor erwarten, hängen eng mit der bevorzugten Arbeitsumgebung zusammen. Lagoudaki nennt dabei vier (vgl. Lagoudaki 2008: 135 ff.):

1. »standard text processing environment« (z. B. MS Word, OpenOffice.org Writer)
2. »dedicated text processing environment«
3. »translator-friendly word processor«
4. »native application of the source file« (DTP-Programm, CMS)

Diejenigen Übersetzer, die in ihrem gewohnten Textverarbeitungsprogramm arbeiten, werden Anaphraseus in seiner Eigenschaft als Plug-in von Writer bevorzugen. OmegaT hingegen bietet eine eigene Übersetzungsumgebung an und ist demnach in die zweite Kategorie einzuordnen.

Das von Lagoudaki erwähnte »translator-friendly word processor« ist kein Textverarbeitungsprogramm in engeren Sinne, sondern eine TM-Anwendung, die über ein Editierfenster verfügt, in die der Ausgangstext in Reintextformat kopiert und übersetzt wird. Anschließend

wird die produzierte Übersetzung in die ursprüngliche Anwendung kopiert. Ein Beispiel für diese dritte Kategorie ist die kommerzielle Anwendung T-Window for Clipboard, die mit Trados Translator's Workbench mitgeliefert wird. Schließlich arbeiten einige Übersetzer am liebsten in der ursprünglichen Anwendung des Ausgangstextes selbst (vierte Kategorie).

5.1.2 Auswahl zwischen Absatz- und Satzsegmentierung

Eine Kritik an den ersten beiden Kategorien von Übersetzungsumgebungen, zu denen Anaphraseus und OmegaT gehören, beschreibt Lagoudaki in der »rigiden« Arbeitsweise, in die sie den Übersetzer durch die vorgegebene Textsegmentierung zwingen:⁴¹

» [...] the first two models described above impose their own way of working through the text, with little flexibility given to the translator for customising the process. Most of these two types of editors segment the source text, forcing the translator to work on a segment-by-segment basis. The feelings of the surveyed TM users about this rigid process were mixed.« (ebd.: 138)

Ist die Segmentierung unumgänglich, so fordern die Studienteilnehmer, dass sie wenigstens die Textsegmentierungseinheit auswählen können:

»Overall, the surveyed TM users seem to prefer being asked how they want to work through the text: sentence by sentence, paragraph by paragraph, or seeing the whole text. This way, the system will adapt to the translation process that is most suitable for each translator, rather than imposing its own.« (ebd.: 139)

OmegaT Die Option zur Segmentierung des Textes auf Absatzebene ist im Projektfenster (Abbildung 4.1b auf Seite 23) beim Anlegen des Übersetzungsprojekts auszuwählen. Diese Option kann auch nachträglich geändert werden, wobei bereits übersetzte Segmente möglicherweise verloren gehen.

Anaphraseus Die globale Auswahl der Textsegmentierungseinheit ist in Anaphraseus nicht so einfach zu konfigurieren, das dies eine Änderung des Quellcodes erfordert: Innerhalb der Makro OFSetup muss der voreingestellte Wert der Variable `Delimiters` von `". !?: "` zu `" "` geändert werden. Als einzige Segmentbegrenzungszeichen innerhalb des Textes bleiben dann die Absatzmarken. Diese Vorgehensweise ist nicht besonders »benutzerfreundlich«: Es

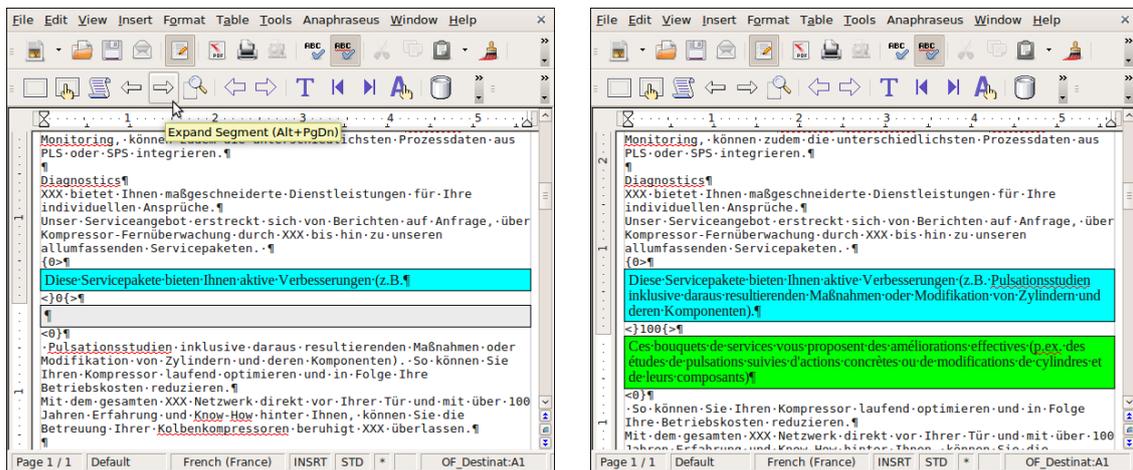
⁴¹ In dieser Arbeit soll die Sinnhaftigkeit der Textsegmentierung an sich und deren möglichen nachträglichen Auswirkungen auf die Übersetzungsarbeit nicht erörtert werden, da sämtliche TM-Systeme den Text vorerst segmentieren müssen, um Vorschläge unterbreiten zu können. Stattdessen wird diesbezüglich auf die in Lagoudaki (2008: 138 ff.) zitierte Literatur verwiesen.

wäre einfacher, wenn sich diese Option in einem Dialogfenster einstellen ließe. Will man sich diesen heiklen Eingriff ersparen, bleibt einem die Möglichkeit, die Segmente einzeln zu vergrößern. Dies stellt allerdings keine globale, für alle Texte gültige Einstellung dar.

5.1.3 Segmente verkleinern/vergrößern

Stößt man während des Übersetzens auf einen Satz, der auf Grund mangelnder oder falsch konfigurierter Regeln fehlerhaft segmentiert wurde, bieten die meisten TM-Systeme die Möglichkeit an, dies über eine Tastenkombination zu ändern.

Anaphraseus Hierzu genügt es, auf die Schaltfläche **Expand Segment** zu klicken (Abbildung 5.1a). Alternativ kann dieser Befehl über die Tastenkombination **[Alt] + [Bild ↓]** auf-



(a) Vorher: Falsche Segmentierung

(b) Nachher: Richtiggestellte Segmentierung

Abbildung 5.1: Segment vergrößern in Anaphraseus

gerufen werden. Über die Schaltfläche **Shrink Segment** (**[Alt] + [Bild ↑]**) kann ein Segment verkleinert werden.

OmegaT Falsch segmentierte Sätze lassen sich in OmegaT nicht so leicht vergrößern oder verkleinern wie in Anaphraseus. Diese im Benutzerforum oft gestellte Frage wird immer damit beantwortet, dass es dazu erforderlich sei, den Ausgangstext in der entsprechenden Anwendung zu öffnen und zu editieren: Ist ein Satz nach einer ungewöhnlichen Abkürzung segmentiert worden, soll im Originaltext das darauffolgende Leerzeichen beispielsweise durch ein geschütztes Leerzeichen ersetzt werden. Anschließend wird die Datei gespeichert, ge-

geschlossen und das Projekt neu geladen. OmegaT führt den Cursor automatisch an die Stelle, an der die Arbeit unterbrochen wurde.

Kommt diese Abkürzung jedoch öfters im Text vor, ist es auf jeden Fall ratsam, eine entsprechende Segmentierungsregel zu erstellen.

5.1.4 Kopieren/Einfügen aus dem Ausgangs- in den Zieltext sowie aus anderen Quellen

Oft müssen während des Übersetzens Stellen aus dem Ausgangstext in den Zieltext übernommen werden – Markennamen, komplizierte Formatierungszeichen, u.v.m. Manchmal müssen sogar ganze Segmente unübersetzt gelassen werden.

OmegaT Dies erfolgt reibungslos über die übliche Tastenkombination  +  (Kopieren) und  +  (Einfügen). Möchte man das gesamte Ausgangstextsegment übernehmen, genügt es, die Tasten  +  +  zu drücken oder den Menüpunkt `Edit > Replace with source` anzuklicken. Je nach Eigenschaften der Vorlage kann es sinnvoll sein, das automatische Editierverhalten von OmegaT einzustellen (siehe Kapitel 5.5.1).

Anaphraseus Die oben genannte Tastenkombination ermöglicht in Anaphraseus ebenfalls das Einfügen von Stellen aus dem Ausgangssegment oder aus externen Quellen. Zum Einfügen des gesamten Ausgangssegments steht die Schaltfläche `Insert Source` bzw. die Tastenkombination  + ⁴² zur Verfügung.

5.1.5 Einfügen von Sonderzeichen

Ist ein sprachenspezifisches oder sonstiges Zeichen über die Tastatur nicht verfügbar, bieten manche TM-Systeme Hilfestellung an.

Anaphraseus Über die programmeigene Zeichentabelle von OpenOffice.org Writer (Menü: `Insert > Special Character...`) lassen sich Sonderzeichen problemlos einfügen (Abbildung 5.2 auf der nächsten Seite).

⁴² Diese Tastenkombination mag irreführend sein, denn mit der Taste  wird eher der Begriff des Löschens und nicht des Einfügens in Verbindung gebracht.

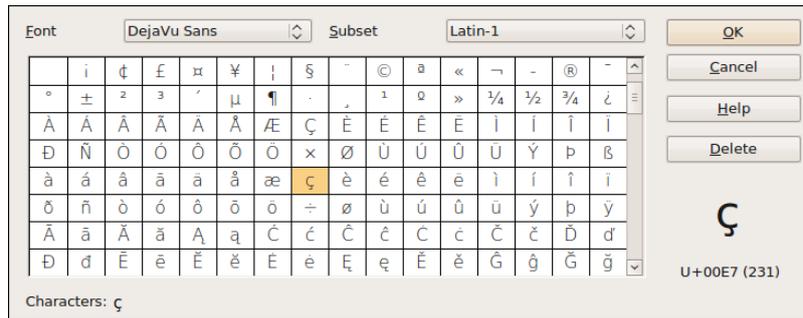


Abbildung 5.2: Zeichentabelle in Writer

OmegaT Es steht keine programminterne Zeichentabelle zur Verfügung. Hierzu ist es notwendig, auf die Zeichentabelle des jeweiligen Betriebssystems zurückzugreifen oder systemweite Tastenkürzel zu konfigurieren.

5.1.6 Formatierung

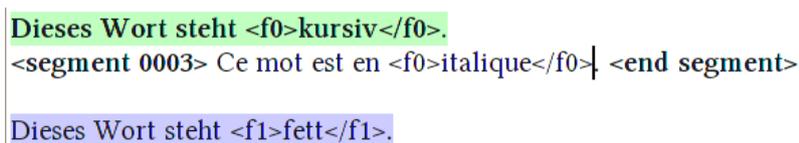
Die Formatierungs- und Layoutproblematik ist ein Teilaspekt des Übersetzungsvorgangs, der – wie Jürgen Schopp betont – nicht unterschätzt werden soll und immer näher ins Blickfeld der Forschung wie auch der Praxis gerückt ist, spätestens jedoch seitdem Computer Einzug in die Welt des Übersetzens gehalten haben. (vgl. Schopp 1999: 199). Abgesehen vom translatorischen Aspekt der Übertragung eines Layouts von einer Ausgangs- in eine Zielkultur und der damit verbundenen Problemstellungen, wird »das Translat heute meist schon am Computer erstellt und als ›druckfertige Übersetzung‹ vom Auftraggeber gefordert.« (ebd.: 199) Darüber hinaus zählen bei bestimmten Textsorten (Anleitungen) Formatierung und Layout zu den wesentlichen Bestandteilen einer Übersetzung, die es gilt, eins zu eins zu übertragen: »Im letzteren (und häufigeren) Fall [Übernahme der *corporate identity* des Auftraggebers] muß die Produktdokumentation in allen Sprachen gleich aussehen (z.B. hinsichtlich Papierformat, Layout, Typographie etc.), unabhängig davon, was in den jeweiligen Märkten üblich ist.« (Schmitt 1999a: 209) Lagoudaki stellt dabei fest:

»One of the major concerns among TM users who use either a MS Word-based editor or a dedicated text processing environment seems to be how well the system deals with complicated formatting and document layout. [...] For the users of these types of editors, the system's ability to preserve the original formatting of the text and the document's layout in perfect condition, as well as identifying text in complex formatting appears to be of paramount importance, as the lack of these features is causing them great frustration and delay in their work. Additionally, some also favour the possibility of cloning the formatting and layout of the source text to the target text, arguing that

a perfect cloning can save them time in recreating a complex formatting.« (Lagoudaki 2008: 143)

In diesem Kapitel werden die Funktionen der jeweiligen Editoren untersucht, die den Übersetzer beim Übernehmen bzw. Übertragen von Textformatierung unterstützen.⁴³

OmegaT Im Editorfenster werden sowohl die Ausgangstext- als auch die Zieltextsegmente in Reintext dargestellt. Besondere Formatierungen von Segmenten (beispielsweise Titel) oder Segmentteilen (fett, kursiv, andere Schriftart) sind somit im Editor ausgeschaltet und nicht als solche erkennbar. Stattdessen werden die betroffenen Stellen in – absichtlich kurz gehaltenen, jedoch kryptischen – Tags eingeschlossen, die der Übersetzer an der gleichen Stelle in den Zieltext übertragen muss, damit die Formatierungen korrekt übernommen werden (siehe Abbildung 5.3).⁴⁴



The image shows a snippet of XML code from the OmegaT editor. The first line is highlighted in green and reads: "Dieses Wort steht <f0>kursiv</f0>." Below it, the XML code is shown: "<segment 0003> Ce mot est en <f0>italique</f0>| <end segment>". The second line is highlighted in blue and reads: "Dieses Wort steht <f1>fett</f1>." The vertical bar in the XML code indicates the current cursor position.

Abbildung 5.3: Darstellung von Formatierungszeichen in OmegaT

An dieser Stelle soll angemerkt werden, dass es sich bei diesen Tags *nicht* um für den Erhalt der Dokumentintegrität notwendige, strukturelle Zeichen handelt: Diese werden vom programmeigenen Dateiformatfilter gleich beim Anlegen bzw. Neuladen des Projektes als solche automatisch erkannt und im Editor nicht angezeigt, da sie nicht zu übersetzen sind.

Anaphraseus Grundsätzlich stehen alle Funktionen von Writer zur Verfügung: Formatvorlagen ändern, Formatierungen (fett, kursiv, u.v.m.) anpassen und übertragen (Pinselsymbol in der Werkzeugleiste, ähnlich wie in MS Word).

Dennoch wurde in einer Reihe von Tests festgestellt, dass Anaphraseus beim Öffnen des zu bearbeitenden Segments *sämtliche Formatierungen zurücksetzt!* Dies kann als Mangel

⁴³ Da es sich um Leistungsmerkmale und nicht um Funktionsmerkmale handelt, werden Fragen zur richtigen Behandlung von speziellen Textabschnitten wie Tabellen, nummerierten Listen, Querverweisen usw. sowie zu möglichen Problemen, die korrupte Formatierungszeichen (sogenannte »Tags«) in strukturierten Dokumenten (XML, HTML, usw.) beim Erstellen der Zieldateien verursachen können, in Kapitel 7.5 auf Seite 82 behandelt.

⁴⁴ In dieser Hinsicht gestaltet sich die Bearbeitung von Dateien im Microsoft OOXML-Format auf Grund des vermehrten Vorkommens solcher Formatierungszeichen weitaus mühsamer als die von ODF-Dokumenten. Eine Konvertierung ist daher zu empfehlen.

angesehen werden, der – laut Dmitri Gabinski⁴⁵ – auf Einschränkungen der verwendeten Programmiersprache StarBasic zurückzuführen ist.

5.1.7 Undo-Funktion

Für den Fall, dass man ein Segment falsch übersetzt oder sich vertippt hat, ist es sinnvoll, über eine Funktion zu verfügen, die bestimmte Schritte rückgängig machen kann.

OmegaT Ist das Übersetzungssegment noch offen, können letzte Eingaben über die Tastaturkombination  +  oder den Menüpunkt Edit > Undo Last Action rückgängig gemacht werden. Wenn man bereits zum nächsten bzw. zu einem anderen Segment übergegangen ist, muss das betreffende Segment erneut editiert werden, wobei die Undo-Funktion in diesem Fall nicht mehr anwendbar ist. Die Funktion Forward bzw. Back in History (im Menü Go To) ist zum Nachverfolgen der zuletzt bearbeiteten Segmente – gerade wenn man quer durch den Text übersetzt – äußerst hilfreich. Sollte das nachträgliche Ändern von Segmenten auf diese Weise mit zu vielen Schritten verbunden sein, bietet sich notfalls die Alternative an, das Projekt in einem – händisch oder automatisch gespeicherten – früheren Stand zu laden (siehe Kapitel 8.5).

Anaphraseus Die Undo-Funktion in Writer ist über die oben erwähnte Tastenkombination bzw. den Menüpunkt Edit > Undo aufrufbar. Hierbei ist allerdings unbedingt darauf zu achten, dass nicht zu viele Schritte rückgängig gemacht werden, denn einige davon wurden von Anaphraseus-Makros automatisch ausgeführt. Wenn eine dieser Operationen versehentlich rückgängig gemacht wird, kann dies einen schwerwiegenden Fehler verursachen, der die gerade bearbeitete Datei unbrauchbar machen kann.⁴⁶

Der Undo-Funktion des Textverarbeitungsprogramms ist daher die Anaphraseus-Funktion Restore Source ( + ) vorzuziehen, da diese den Ausgangszustand des gerade bearbeiteten Segments auf sichere Art wieder herstellt.

⁴⁵ Dieser Punkt wurde in einer E-Mail-Korrespondenz zwischen dem *Release Manager* und dem Verfasser dieser Arbeit geklärt.

⁴⁶ Dies ist eine von Trados Translator's Workbenchs Anwendern, die ihr Programm in der Word-Umgebung ausführen, ebenfalls bekannte Fehlerquelle.

5.2 Anzeige

Dieses Kapitel widmet sich den Funktionen, die die Texteditoren auf der Ausgabeseite anbieten.

5.2.1 Formatierungen

Aus der Art und Weise, wie Textformatierungen im OmegaT-Editor angezeigt werden, können sich Probleme ergeben, auf die sowohl die translationswissenschaftliche Forschung als auch die Praxis aufmerksam wurden. So schreibt Lagoudaki:

»On the issue of source text visibility, using frequency of mention as a surrogate for importance, the survey showed that the visibility of the source document at all times (>the capacity to work side-by-side (Source-Target)<) during translation is imperative to any translation process. In fact, it appears that it is not only enough to be able to view the source text as a list of sentences, but it is also important to see the formatting of the text, as well as any non-textual elements, in other words, to be able to view the source text in its original form.« (Lagoudaki 2008: 139)

Da die Anzeige in OmegaT dahingehend nicht verändert werden kann, empfiehlt es sich, einen Teil des Computerbildschirms für die Anzeige des Originaldokuments samt Formatierungen und Bildern zu verwenden. Bei der Verwendung von Anaphraseus entfällt dieses Problem.

5.2.2 Zieltextvorschau

Eine weitere Funktion, die Studienteilnehmer von ihrem Editor erwarten, ist die Möglichkeit, sich den Zieltext jederzeit anzeigen zu lassen: »*Translators appeared to appreciate the possibility of previewing their translation in its original file format at any time by clicking on an icon, in order to ensure that they have not missed any non-editable text that also requires translation.*« (ebd.: 141)

Als »*dedicated text processing environment*« zeigt OmegaT Ausgangs- und Zieltext in einer vereinfachten Form an, die nicht dem WYSIWYG-Grundsatz entspricht. Wünscht man, sich eine Vorschau des übersetzten Texts samt Formatierungen und Bildern anzeigen zu lassen, ist es erforderlich, das/die Zieldokument(e) über den Menüpunkt **Project > Create Translated Documents** oder die Tastenkombination  +  zu erstellen. Dieser Vorgang kann jederzeit – die Übersetzung muss dazu nicht fertig sein – und beliebig oft wiederholt werden, wobei ältere Zieldokumente überschrieben werden. Die entsprechenden Datei-

en befinden sich im Projektunterordner /target/ und tragen den jeweils gleichen Dateinamen wie die Vorlage (einstellbar). Durch Öffnen in der entsprechenden Originalanwendung lassen sie sich problemlos betrachten.

Bei der Verwendung von Anaphraseus entfällt dieses Problem.

5.2.3 Anzeige des Übersetzungsfortschritts

Um sich Klarheit über den Stand der Übersetzung zu verschaffen, ist es hilfreich, auf statistische Werte zurückgreifen zu können.

OmegaT Wie in Kapitel 4.4 erläutert, verfügt OmegaT über zahlreiche Statistikfunktionen. Zusätzlich zu den dort erwähnten Werten werden in allen drei Anzeigetypen Kennzahlen über den Übersetzungsfortschritt bereitgestellt:

1. Statistik- und Navigationsfenster (Abbildung 4.10a auf Seite 41)
2. Projektbericht (Abbildung 4.10b auf Seite 41)
3. Minianzeige im unteren Hauptfensterbereich rechts⁴⁷ (Abbildung 5.4)

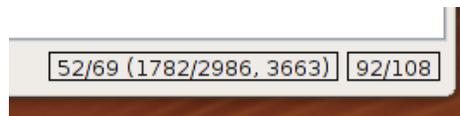


Abbildung 5.4: Statistische Hinweise im OmegaT-Hauptfenster

Wie aus Tabelle 5.1 zu entnehmen, liefern diese drei Anzeigetypen zahlreiche Informationen über den Stand der Übersetzung.

		Bezugsgröße		
		Gesamtprojekt	Datei	Segment
Anzeigetyp	Statistikfenster	Segmente	Segmente	–
	Projektbericht	Segmente Wörter Anschläge ⁴⁸	Wörter Anschläge ⁴⁸	–
	Minianzeige	Segmente	Segmente	Anschläge ⁴⁹

Tabelle 5.1: Verfügbare Kennzahlen über den Übersetzungsfortschritt in OmegaT

⁴⁷ Fährt man mit dem Mauszeiger über diesen Fensterbereich, erscheint eine Quickinfo mit einer Erklärung über die Bedeutung der angezeigten Zahlen.

⁴⁸ Mit und ohne Leerzeichen.

⁴⁹ Mit Leerzeichen.

Interessant ist auch die Information – in der Minianzeige – über die Länge des gerade bearbeiteten Segments (Quell- und Zielsegment). Dies ist besonders bei solchen Aufträgen nützlich, bei denen der Auftraggeber eine bestimmte Satzlänge vorgibt, die nicht überschritten werden darf.

5.3 Terminologie verwenden

Verfügt der Übersetzer über Glossare oder Wörterbücher und sind diese ins jeweilige TM-System erfolgreich eingebunden worden⁵⁰, stehen dem Übersetzer einige Funktionen zur Verfügung, die in diesem Kapitel untersucht werden:

5.3.1 Nachschlagen – Einfügen in den Zieltext

OmegaT Die Übersetzung erfolgt meist linear von einem Segment zum nächsten: Öffnet der Übersetzer ein neues Segment, schlägt das System automatisch in den vorhandenen terminologischen Ressourcen – Glossar(e), Wörterbuch(-bücher) – nach, ob es einen oder mehrere Wörter dort vorfindet und unterbreitet Listen von Vorschlägen, die nach dem Ähnlichkeitsgrad mit dem/den Ausgangswort(-wörtern) geordnet sind⁵¹ (Abbildung 5.5).

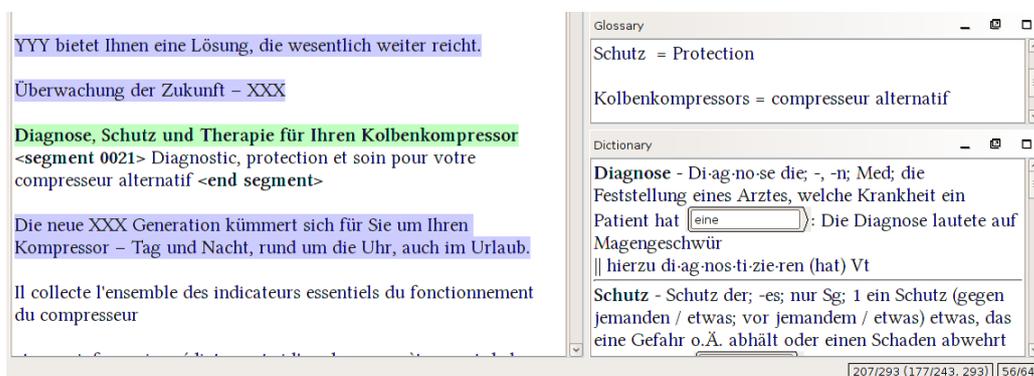


Abbildung 5.5: Terminologische Treffer (Glossar und Wörterbuch) in OmegaT

Um das Wort aus dem Glossar in den Zieltext einzufügen, verfügt OmegaT weder über Tastenkombination noch Menüpunkt. Hierzu ist es erforderlich, das gewünschte Wort bzw.

⁵⁰ Siehe Kapitel 4.2.3 auf Seite 34.

⁵¹ Wie leistungsfähig der programminterne Suchalgorithmus ist, wird in Kapitel 7.1.4 behandelt. An dieser Stelle soll nur darauf hingewiesen werden, dass sich die besten Suchergebnisse mit Termini in der Grundform erzielen lassen.

die gewünschte Wortfolge im Glossar- oder Wörterbuchfenster zu markieren und anschließend über die übliche Tastenkombination (siehe Seite 51) zu kopieren und an geeigneter Stelle in den Zieltext einzufügen.

Anaphraseus Wenn das System im geöffneten Segment auf ein im Glossar vorhandenes Wort stößt, wird dieses farblich gekennzeichnet, wobei die Signalwirkung auf Grund der vom System vorgegebenen Farbe (Tiefblau auf ohnehin blauem Hintergrund) eher gering ist (Abbildung 5.6). Außerdem ist es erforderlich, die Schaltfläche **Next Term** anzuklicken bzw. die Tastenkombination  +  zu drücken, damit das Wort auch tatsächlich ausgewählt und durch blaue Unterwellen gekennzeichnet wird. Zusätzlich erscheint im unteren Hauptfensterbereich links der vollständige Glossareintrag in der Form "Wort in der Ausgangssprache - Wort in der Zielsprache". Anschließend lässt sich das gewünschte

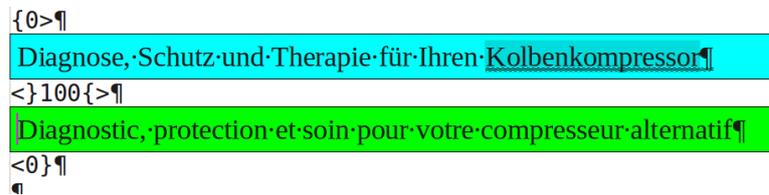


Abbildung 5.6: Terminologischer Treffer (Glossar) in Anaphraseus

Wort aus dem Glossar durch Anklicken der Schaltfläche **Paste Term** bzw. Drücken der Tastenkombination  +  in den Zieltext an der aktuellen Stelle des Cursors einfügen.

5.3.2 Speichern von neuen Begriffspaaren

Oft sind terminologische Ressourcen unvollständig und werden vom Übersetzer im Verlauf des Projektes auf der Grundlage seiner eigenen Rechercharbeit nach und nach ergänzt.

OmegaT Um neue Begriffspaare ins Glossar zu speichern, verfügt OmegaT über keine programmeigene Funktion. Hierzu muss die Glossardatei in der geeigneten Anwendung – beispielsweise in einem einfachen Texteditor – bearbeitet und anschließend gespeichert werden. Seit der Version 2.0.3 werden Glossardateien im Sekundentakt vom System überprüft. Dadurch entfällt das sonst nach Änderungen von Projektdateien erforderliche Neuladen des Projekts.

Anaphraseus Das System bietet zum Speichern von neuen Begriffspaaren während des Übersetzungsvorgangs eine integrierte Funktion an, die über die Schaltfläche **Add Term** bzw.

die Tastenkombination  +  aufgerufen wird. Es öffnen sich mehrere Dialogfenster hintereinander, die den Anwender zur Eingabe des zu speichernden neuen Begriffes in der Ausgangs- und Zielsprache sowie einer optionalen Beschreibung auffordern. Dieser Vorgang kann jederzeit abgebrochen werden.

5.3.3 Listen von Stoppwörtern

Oft werden beim automatischen Nachschlagen in vorhandenen Wörterbüchern allzu gängige Begriffe als Treffer präsentiert, die von geringem Interesse für die Übersetzung sind, wodurch interessantere Begriffe in der Menge der angezeigten Stichwörter untergehen. Dies kann über die Definition einer Liste von sogenannten Stoppwörtern verhindert werden. Dieser Begriff, der aus dem Bereich der Information-Retrieval-Technologien stammt, wird von Arturo Trujillo wie folgt erklärt:

»Many of the most frequent words in a language are very poor discriminators of meaning. In English, using such function words as *the, of, a, an, in* and *on* in calculating similarity measures would result in almost any sentence matching any other to some extent. [...] These words are usually collected in a stoplist which simply enumerates everything that needs to be disregarded when calculating similarities.« (Trujillo 1999: 63)

In OmegaT steht hierzu die anfangs leere Datei `ignore.txt`, die sich im Projektunterordner `/dictionary/` befindet, zur Verfügung. In diese Datei werden die Wörter untereinander eingegeben, die beim Nachschlagen im Wörterbuch ignoriert werden sollen. Derartige sprachenbezogene Stopplisten können im Übrigen aus dem Internet heruntergeladen werden.

Über diese Funktion verfügt Anaphraseus nicht.

5.4 Konkordanzsuche

Der Begriff der Konkordanzsuche, der ebenfalls im Bereich der IR-Technologien – insbesondere in der Korpuslinguistik – entstanden ist, steht für eine *»nützliche Funktion, mit der während des Übersetzens nach Begriffen im Memory bzw. im Referenzmaterial gesucht werden kann.«* (Massion 2005: 281)

OmegaT Dieses System verfügt über eine sehr leistungsfähige Funktion, mit der alle im Projektordner befindlichen Ressourcen (TM, Ausgangstext, Glossar) durchsucht werden können. Die Funktion wird über die in vielen Anwendungen übliche Tastenkombination  +  bzw. den Menüpunkt `Edit > Search Project...` aufgerufen. Es stehen einfache

Suchoptionen, die mittels Kontrollkästchen ausgewählt werden können – exakte Suche, unscharfe Suche (Keyword search), Berücksichtigung von Groß- und Kleinschreibung, TM-Durchsuchung, Durchsuchung eines separaten Ordners, Anzeigoptionen (erstrangige bzw. alle Treffer) –, wie in Abbildung 5.7 dargestellt, sowie eine Expertensuche, die über die Eingabe von regulären Ausdrücken gesteuert wird (Tabelle 9 auf Seite 103 im Anhang).

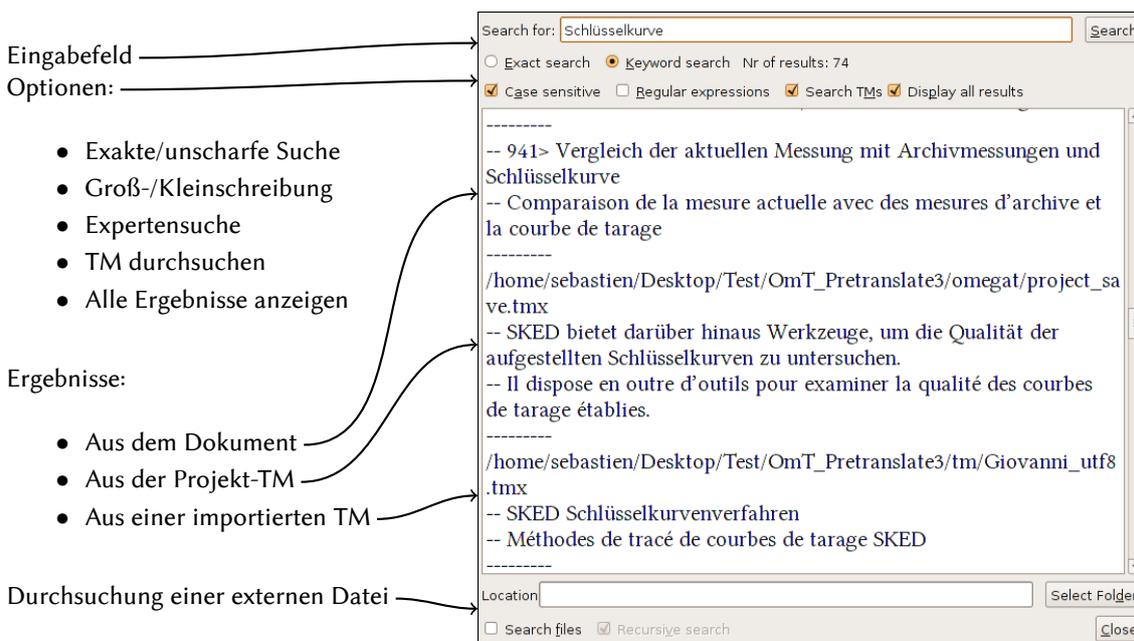


Abbildung 5.7: Konkordanzsuche in OmegaT

Ohne auf die Experten-Option zurückgreifen zu müssen, lassen sich Suchanfragen auch mit den in anderen Anwendungen üblichen Platzhaltern * (ersetzt beliebig viele Zeichen) und ? (ersetzt genau ein Zeichen) ausführen. So kann beispielsweise die Eingabe von »A*end« »Abbildend«, »Absteigend« und viele weitere Suchergebnisse liefern, die von »A?end« jedoch nur »Abend«.

Bei Treffern, die in einem Dokument des Projektes gefunden wurden, besteht außerdem die Möglichkeit, mit einem Doppelklick auf den jeweiligen Treffer direkt zum entsprechenden Segment im Text zu navigieren.

Schließlich soll angemerkt werden, dass die Suche sprachenunabhängig erfolgt. Somit kann ein Wort in der Ausgangs- oder in der Zielsprache eingegeben werden, ohne dass es einer vorherigen Einstellung bedarf.

Anaphraseus Das TM-System verfügt über keine Konkordanzsuchfunktion. Diese Funktionalität dürfte jedoch demnächst implementiert werden (vgl. Medvedev 2008).

5.5 Sonstige Produktivitätssteigerung

5.5.1 Automatische Propagation

OmegaT Sobald die erste Okkurrenz eines im Ausgangstext mehrmals und in der gleichen Form vorkommenden Segments⁵² vom Benutzer übersetzt wurde, werden alle weiteren Okkurrenzen dieses Segments im gesamten Text durchgängig und automatisch ersetzt. Diese auch bei den kommerziellen Anwendungen Déjà Vu, Transit und SDLX verfügbare Funktionalität (vgl. den Eintrag »AutoPropagate« in Massion 2005: 279) kann sich als zeitsparend erweisen, ist allerdings nicht einstellbar.

Nicht mit automatischer Propagation gleichzusetzten, jedoch als durchaus nützliche Funktion zu betrachten, ist die Möglichkeit, das Editierverhalten von OmegaT zu konfigurieren: Beim Öffnen eines neuen Segments kann das Editierfeld wahlweise leer gelassen, mit dem unveränderten Ausgangssegment oder automatisch mit dem besten TM-Treffer gefüllt werden, wobei die Mindesttrefferquote einstellbar ist (Abbildung 5.8). Außerdem kann letztere

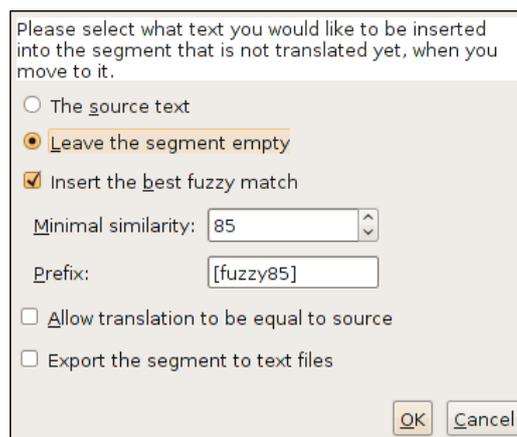


Abbildung 5.8: Einstellung des Editierverhaltens von OmegaT

Option dazu verwendet werden, einen halbautomatischen Vorübersetzungsvorgang durchzuführen: Hierzu wird der Trefferschwelienwert auf 100% gesetzt und von einem nicht übersetzten Segment zum nächsten durch Drücken der Tastenkombination **Strg** + **U** navigiert: Dadurch werden alle *100% matches*⁵³ automatisch in den Zieltext eingefügt.

Anaphraseus Das System verfügt über diese Funktion nicht.

⁵² Sogenannte »Wiederholung«. Zur Klärung dieses Begriffes, siehe Kapitel 7.1.2 auf Seite 74.

⁵³ Diese *100% matches* sind allerdings keine *exact matches*, wie der Test auf Seite auf Seite 76 zeigt.

5.5.2 Automatisches Ersetzen von Maß- und Währungseinheiten, Datum- und Zeitangaben

Maß- und Währungseinheiten sowie Datum- und Zeitangaben im Ausgangstext stellen für den Übersetzer nicht selten eine Herausforderung dar. Von einer Sprache zur anderen, sogar von einer Region innerhalb einer Sprache (*locale*) zur anderen können Bezeichnungen und Schreibweisen abweichen (siehe Tabelle 5.2).

Sprache	Locale	Währung	Datum	Uhrzeit
Deutsch	Deutschland	12. 050, 25 €	16. 03. 2009	22. 15 Uhr
	Österreich	€ 12. 050, 25	16. 03. 2009	22. 15 Uhr
Französisch	Frankreich	12 050, 25 €	16/03/2009	22 h 15
Englisch	England	£12, 050. 25	16/03/2009	10: 15 PM
	USA	\$12, 050. 25	03/16/2009	10: 15 PM

Tabelle 5.2: Unterschiedliche Einheiten und Schreibweisen je nach Sprache und *locale*.

Zusätzlich dazu ist es je nach Texttyp notwendig, kulturabhängige Einheiten – man denke an das angloamerikanische Maßsystem – umzurechnen (vgl. Schmitt 1999b: 298 f.).

Im Gegensatz zu einigen kommerziellen TM-Systemen (vgl. Massion 2005: 291) verfügen weder OmegaT noch Anaphraseus über eine Funktion zum automatischen erkennen und Ersetzen von Zahlenvariablen.

5.5.3 Rechtschreibprüfung

Laut Lagoudaki gehört die Rechtschreibprüfung zu den wichtigsten Komponenten eines TM-Systems:

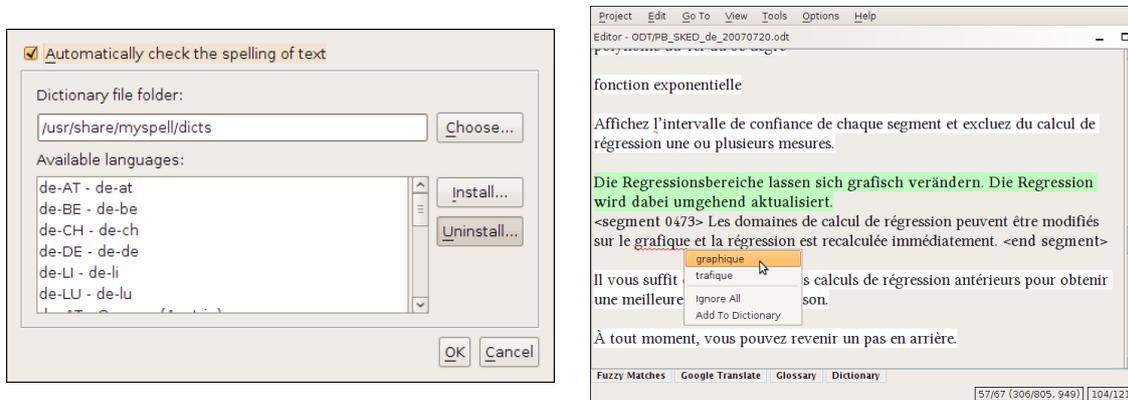
»Regarding the type of errors that a TM system should be able to detect and correct automatically, spelling mistakes were the ones most frequently mentioned by translators, reviewers and proofreaders. More specifically, these TM users expect the system to be able to run context-sensitive language checks with the possibility of learning from the corrections made by the users.« (Lagoudaki 2008: 168)

Dabei wird zwischen Rechtschreibprüfung während der Übersetzung und Rechtschreibprüfung nach der Übersetzung unterschieden.⁵⁴

Sowohl OmegaT als auch Anaphraseus unterstützen die Rechtschreibprüfung während der Eingabe und greifen dabei auf die Ressourcen von Hunspell zurück.

⁵⁴ Der Vorgang der Rechtschreibprüfung nach der Übersetzung gehört zur Qualitätsprüfung und wird in Kapitel 6.1.1 auf Seite 67 behandelt.

OmegaT Um diese Funktion nutzen zu können, müssen die MySpell/Hunspell-Wörterbücher aus dem Internet heruntergeladen⁵⁵ und konfiguriert werden. Anschließend wird die Funktion über den Menüpunkt Options > Text Checking... aktiviert (Abbildung 5.9a).



(a) Aktivierung der Rechtschreibprüfung

(b) Rechtschreibfehler mit Korrekturauswahl

Abbildung 5.9: Rechtschreibprüfung während der Übersetzung in OmegaT

Wurde ein Rechtschreib- oder Tippfehler während der Eingabe erkannt, wird das betreffende Wort rot unterwellt. Bei einem Rechtsklick auf die fehlerhafte Stelle werden Vorschläge aus dem Rechtschreibwörterbuch unterbreitet (Abbildung 5.9b). Zur Auswahl stehen – wie bei gängigen Rechtschreibprüfungsprogrammen – folgende Optionen:

1. Einen der Rechtschreibvorschläge annehmen: Das fehlerhafte Wort wird durch den ausgewählten Vorschlag im Text ersetzt.
2. Jede weitere Okkurrenz dieses Wortes ignorieren: Ein Eintrag erfolgt in die Projektdatei `ignored_words.txt` im Projektunterordner `/omegat/`.
3. Dieses Wort »ins Wörterbuch« übernehmen: Ein Eintrag erfolgt in die Projektdatei `learned_words.txt` im Projektunterordner `/omegat/`.

Bei der Auswahl von Option 2 oder 3 werden die entsprechenden Wörter nicht mehr auf die Rechtschreibung überprüft. Änderungen werden in externe, projektbezogene Dateien gespeichert. Dies hat einen doppelten Vorteil: Erstens werden die ursprünglichen Wörterbücher durch falsche Eingaben nicht beschädigt und zweitens ermöglichen diese Wörtersammlungen den Aufbau weiterer projekt-, kunden-, sprachen- und/oder bereichsspezifischer Glossare.

Es soll an dieser Stelle angemerkt werden, dass MySpell/Hunspell-Wörterbücher *locales* unterscheiden. Somit muss dies bei der Auswahl der Projektsprachen berücksichtigt werden:

⁵⁵ Bei einer Installation von OpenOffice.org werden die Wörterbücher mitinstalliert.

Wird beim Anlegen des Übersetzungsprojekts beispielsweise nur der Sprachcode "DE" eingegeben, kann keine Rechtschreibprüfung erfolgen, weil die verfügbaren Myspell/Hunspell-Ressourcen die Eingabe des entsprechenden *locale* erfordern (siehe Abbildung 5.9a).

Anaphraseus Sofern die projektbezogene TM hinsichtlich Ausgangs- und Zielsprache richtig konfiguriert und die Rechtschreibfunktion von Writer aktiviert wurde, erfolgt die Rechtschreibprüfung während der Eingabe des Übersetzungstextes. Da Writer ebenfalls auf die MySpell/Hunspell-Technologie zurückgreift, stehen ähnliche Funktionen wie bei OmegaT zur Verfügung: Bei einem Rechtsklick auf das fehlerhafte Wort erscheint ein Kontextdialog mit verschiedenen Optionen (Vorschlag annehmen, ignorieren, ins Wörterbuch übernehmen). Auch hier sind benutzerdefinierte Wörterbücher, in die »neue Einträge« abgelegt werden, konfigurierbar.

5.5.4 MT-Modul

Seit der am 9.9.2009 veröffentlichten Betaversion 2.0.4 verfügt OmegaT über ein zusätzliches Modul, das es ermöglicht, das online-Maschinenübersetzungssystem Google Translate automatisch abzufragen. Bei jedem geöffneten Segment erscheint in einem Nebenfenster ein MT-Vorschlag zusätzlich zu den bereits beschriebenen Informationsquellen (TM, Glossare, Wörterbücher).⁵⁶

5.6 Zusammenfassung

Hinsichtlich der unmittelbaren Unterstützung des Benutzers während des Übersetzungsvorgangs bieten sowohl OmegaT als auch Anaphraseus relativ umfangreiche Funktionen an. Beide untersuchten TM-Systeme erfüllen die von Lagoudaki in ihrer Definition⁵⁷ geforder-

⁵⁶ Da diese Funktionalität recht neu ist, konnte sie vom Verfasser dieser Arbeit nicht ausführlich getestet werden. Sie wird dennoch der Vollständigkeit halber hier erwähnt. Ebenso wenig sollen an dieser Stelle der potenzielle Nutzen und die möglichen Risiken – insbesondere hinsichtlich der vertraulichen Behandlung von Daten –, die mit der Verwendung dieser Technologie einhergehen, erörtert werden. Es soll nur darauf hingewiesen werden, dass durch diese Neuerung ein weiterer Schritt in Richtung eines TM-Systems des »dritten Typs« – wie sie Melby beschreibt (siehe Seite 5) – gesetzt wurde. Allerdings handelt es sich hierbei nicht um »reine« maschinelle Übersetzung, da es dem Benutzer nach wie vor frei steht, den Übersetzungsvorschlag anzunehmen, abzulehnen oder zu verändern – im Gegensatz zur MT, bei der der Computer weitgehend die Kontrolle übernimmt. Vielmehr kann dieses neue Modul als zusätzliches CAT-Tool betrachtet werden, bei dem die Technologie der EBMT (*Example Based Machine Translation*, vgl. Hutchins 1998: 14; Hutchins 2005: 15; Lagoudaki 2008: 45 f.; Bowker 2002: 138; Somers 2003: 44 f.) zum Einsatz kommt, selbst wenn es sich bei Google Translate um EBMT handelt, die auf statistischen Algorithmen basiert.

⁵⁷ Siehe Seite 11.

ten Funktionsmerkmale zu einem großen Teil.

Sind Funktionen dennoch nicht verfügbar, befinden sie sich entweder in der Entwicklungsphase oder können über das Feature-Request-System der jeweiligen Anwendung angefordert werden. Betroffen sind bei Anaphraseus die Konkordanzsuche und bei OmegaT die Glossarfunktion, die nach eigener Aussage eines Teammitglieds »unterentwickelt« ist: »*The glossary function is notoriously underdeveloped in OmegaT, this will be the object of further developments in the future eventually.*«⁵⁸

Bei Anaphraseus kann die Tatsache, dass beim Öffnen eines Segments die Formatierung desselben verlorengeht, gewissermaßen als Mangel eingestuft werden.

Als hervorragend hingegen kann die Konkordanzsuche in OmegaT bewertet werden, da sie sowohl einfache als auch Expertensuchanfragen ermöglicht. Des Weiteren ist die jüngste Einbindung eines MT-Moduls, das Übersetzungsvorschläge aus den enormen Ressourcen von Google schöpft, vielversprechend, muss sich jedoch auf längere Sicht erst bewähren.

58 Aus: Jean-Christophe Helary (2009): *Re: [OmT] Problems in importing glossary and TM from Trados into OmegaT (English-Chinese)*. URL: <http://tech.groups.yahoo.com/group/OmegaT/message/13024> (besucht am 09. 09. 2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 08. 02. 2009. Archivierte URL verfügbar: <http://www.webcitation.org/5jsY3K0Du>

6 Funktionale Merkmale: Nach dem Übersetzen

Im Anschluss an die Übersetzung des Ausgangsmaterials, bieten die meisten TM-Systeme Funktionen zur nachträglichen Bearbeitung und Qualitätssicherung an, die in diesem Kapitel beschrieben werden.

6.1 Qualitätsprüfung

Die Messung und Sicherung der Qualität des Translats gehört zu den Kernbereichen der translationswissenschaftlichen Forschung. Zahlreiche Projekte lieferten Ansätze zu einer Typologie von Übersetzungsfehlern (z.B. das MeLLANGE-Projekt). Im Bereich des Qualitätsmanagements führten Forschungsergebnisse zur Verfassung der europäischen Norm EN 15038 *Übersetzungs-Dienstleistungen – Dienstleistungsanforderungen*.

Trotz dieser Versuche, Fehlerkategorien nach objektiven Kriterien zu schaffen, sind Computer nicht im Stande, alle Fehlertypen zu analysieren. So schreibt Massion (Massion 2005: 149):

»Es ist nicht so einfach, die Qualität einer Übersetzung zu bewerten, denn die Sprache ist nicht in jeder Hinsicht genau und messbar. Faktoren wie Stil oder Verständnis des Textes können nur von einem fachkundigen Übersetzer bzw. von einem sprachkundigen Fachmann beurteilt werden. Es gibt jedoch eine ganze Reihe von Merkmalen, die mit Hilfe einer Software geprüft werden können. Das sind beispielsweise:

- die korrekte Rechtschreibung des Textes
- die Einhaltung der Terminologie
- die Übereinstimmung der Formatierung bzw. der Tags in beiden Sprachversionen
- die Korrektheit von Zahlen
- die Vollständigkeit der Übersetzung«

Lagoudaki (2008: 169) führt eine ausführlichere Liste von Qualitätsmerkmalen an. Aus Gründen der Einfachheit wird in diesem Kapitel dennoch Massions Systematik verwendet, da sie die relevantesten Probleme zusammenfasst.

6.1.1 Rechtschreibprüfung

Weder OmegaT noch Anaphraseus bieten Funktionen zur Überprüfung der fertigen Übersetzung auf Rechtschreibfehler an. Hierzu ist es erforderlich, die Funktionen zur Prüfung während der Eingabe (Segment für Segment) zu benutzen oder auf die Funktionen zurückzugreifen, die die dateiformatbezogene ursprüngliche Anwendung jeweils anbietet. Da die meisten Dateien, die übersetzt werden, Office-Dokumente sind, können die umfangreichen Rechtschreib- und Grammatikprüfungsfunktionen von OpenOffice.org in Anspruch genommen werden⁵⁹.

6.1.2 Terminologieprüfung

Ist auf Grund der Textsorte⁶⁰ oder des Kundenauftrags⁶¹ die Verwendung einer bestimmten vorgegebenen Terminologie zwingend erforderlich, halten TM-Systeme in vielen Fällen Funktionen bereit, die die durchgängige Verwendung dieser Terminologie – man spricht in diesem Zusammenhang von *terminologischer Konsistenz* – überprüfen.

Dennoch verfügen die beiden untersuchten TM-Systeme über keine derartige Funktion. Wie bereits erwähnt⁶², ist sich das OmegaT-Entwicklungsteam der mangelnden Funktionen im Zusammenhang mit der Terminologieverwaltung – auch was die nachträgliche Prüfung auf Konsistenz anbelangt – durchaus bewusst. Im Falle von Anaphraseus scheint das Implementieren einer derartigen Funktion nicht geplant zu sein.

6.1.3 Tagschutz und -prüfung

Werden bestimmte Tags ausgelassen oder falsch gesetzt, kann dies zu fehlerhaften Formatierungen des Zieldokumentes führen. Bei manchen TM-Systemen verhindert ein fehlendes Formatierungszeichen sogar die Erstellung des Zieldokumentes. Um dem entgegenzuwirken, bieten diese Systeme Funktionen zur nachträglichen Prüfung der Formatierungszeichen.

OmegaT In diesem System wird die nachträgliche Prüfung der Formatierungstags über den Menüpunkt Tools > Validate Tags bzw. die Tastenkombination  +  aufgerufen.

⁵⁹ Die Rechtschreib- und Grammatikprüfung von Writer soll in dieser Arbeit nicht beschrieben werden.

⁶⁰ Fachsprachliche Texte (aus dem technischen, wissenschaftlichen, juristischen oder medizinischen Bereich) erfordern, dass zu einem Fachbegriff jeweils nur eine Benennung verwendet wird, um Zweideutigkeiten auszuschließen.

⁶¹ Je nach Auftrag kann eine unternehmensspezifische Terminologie, die zur *corporate identity* gehört, vorgegeben sein.

⁶² Siehe Seite 65.

fen. Alle Ausgangs- und Zielsegmente werden automatisch abgeglichen und Segmentpaare, bei denen Formatierungszeichen ausgelassen oder anders gesetzt wurden, werden in einem Dialogfenster angezeigt, das gleichzeitig das Navigieren zu den betreffenden Segmenten ermöglicht (Abbildung 6.1).



Abbildung 6.1: Tagprüfung in OmegaT

Anaphraseus Dieses System bietet diese Funktion nicht an.

6.1.4 Zahlen

Wie bereits in Kapitel 5.5.2 erwähnt, erkennt keines der beiden untersuchten TM-Systeme Zahlenvariablen als besonders zu behandelnde Textelemente. Demnach verfügen sie auch über keine Funktion zur Prüfung deren richtiger Übertragung aus dem Ausgangs- in den Zieltext.

6.1.5 Vollständigkeit der Übersetzung

Manche Systeme erkennen, ob Segmente bei der Übersetzung ausgelassen oder unvollständig übersetzt wurden. Eine diesbezügliche nachträgliche Prüfung ist weder bei OmegaT noch Anaphraseus gegeben. Nichtsdestoweniger bietet zumindest OmegaT die Möglichkeit, nicht übersetzte Segmente über Einstellungen der grafischen Benutzeroberfläche (Menü *View*) farblich zu kennzeichnen: Optional können dadurch übersetzte Segmente mit gelber, nicht übersetzte Segmente mit grauer Farbe untermalt werden, was deren visuelle Unterscheidung am Bildschirm erleichtert.

6.2 Lieferung der Zieldateien

Sobald die Übersetzungs- und die Qualitätsprüfungsphase abgeschlossen sind, werden die fertiggestellten Dateien – meistens im gleichen Dateiformat – an den Auftraggeber gesendet. Dennoch erwarten einige Auftraggeber – vor allem Übersetzungsdienstleister –, dass die

Dateien in einem zweisprachigen Format (Trados TagEditor TTX, »unbereinigte« Worddokumente oder XLIFF) zwecks interner Weiterverwendung geliefert werden.

6.2.1 Einsprachige Dateiformate

OmegaT Die Zieldateien werden über den Menüpunkt **Project > Create Translated Documents** oder die Tastenkombination  +  erstellt und sind anschließend im Projektunterordner `/target/` zu finden.

Anaphraseus Bei diesem System verläuft der Übersetzungsvorgang meist linear von einem Segment zum nächsten. Wurde ein Segment übersetzt, wird es vom System mit einer besonderen Formatierung versehen, die es zwar in den Hintergrund treten lässt, aber nicht löscht. Somit verfügt der Übersetzer am Ende des Vorgangs über eine zweisprachige Datei mit »versteckten« Ausgangssegmenten und normal formatierten Zielsegmenten. Dieses Verfahren entspricht denjenigen – in der MS-Word-Umgebung – von Trados Translator's Workbench und Wordfast.

Um die versteckten Ausgangssegmente zu löschen, muss die Datei »bereinigt« werden, indem die Funktion **Clean-Up** über die entsprechende Schaltfläche oder die Tastenkombination  +  aufgerufen wird.

Okapi Framework: Rainbow In Kapitel 4.2.1 wurde die Möglichkeit beschrieben, Dateien, die in einem von OmegaT nicht unmittelbar unterstützten Format vorliegen, von Rainbow – der grafischen Benutzeroberfläche zahlreicher Filter- und Konvertierungsanwendungen aus der Reihe des Okapi Framework – konvertieren zu lassen. Dabei konnte eine gesamte OmegaT-Projektstruktur samt konvertierter Ausgangsdateien angelegt werden.

Zur Rückkonvertierung wird folgendermaßen vorgegangen:

1. Die Zieldateien werden in OmegaT wie oben beschrieben erstellt.
2. In Rainbow wird die Adresse des lokal gespeicherten Projektordners als Wurzeladresse gesetzt und die bei der Erstellung des Projektordners automatisch angelegte Datei `manifest.xml` zur Bearbeitungsliste hinzugefügt (Abbildung 6.2a auf der nächsten Seite)
3. Über den Menüpunkt **Utilities > Translation Package Post-Processing...** wird die Rückkonvertierungsfunktion aufgerufen und über die Schaltfläche **Execute** des erscheinenden Dialogfensters gestartet (Abbildung 6.2b auf der nächsten Seite).
4. Die fertig konvertierten Dateien stehen anschließend im neu angelegten Unterordner `/done/` zur Verfügung.

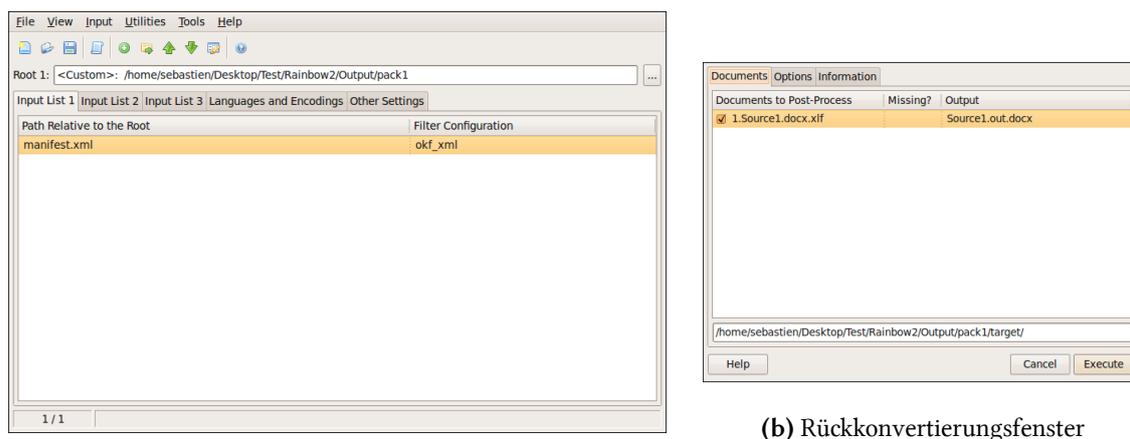


Abbildung 6.2: Rückkonvertierung von Dateien ins ursprüngliche Format mit Okapi Rainbow

6.2.2 Zweisprachige Dateiformate

Toxic Wenn der Übersetzer zweisprachige Trados TagEditor TTX-Dateien von seinem Auftraggeber erhält, können diese Dateien mit Hilfe des auf Seite 31 erwähnten Toxic-Programms in ein von OmegaT lesbares Zwischenformat (TOX) konvertiert werden. Nun kann mit demselben Programm die Rückkonvertierung aus dem TOX-Format ins TTX-Format erfolgen.

Anaphraseus Falls der Auftraggeber die Lieferung von »unbereinigten« MS-Word-Dateien wünscht, genügt es, in Anaphraseus den oben beschriebenen Schritt der »Bereinigung« zu überspringen und die OpenOffice.org ODT-Datei mit Writer ins MS-Word DOC-Format abzuspeichern.

Auch wenn er OmegaT benutzt hat, kann der Übersetzer »unbereinigte« Dateien ebenfalls mit Hilfe von Anaphraseus erstellen. Hierzu muss folgendermaßen vorgegangen werden:

1. Die Zieldateien werden in OmegaT wie oben beschrieben erstellt und anschließend in Writer geöffnet.
2. Eine der im Verlauf des Übersetzungsprojekts von OmegaT erstellten TM-Dateien – vorzugsweise OmT-level1.tmx – wird in Anaphraseus entweder unmittelbar oder nach einer Konvertierung durch Olifant⁶³ importiert.

⁶³ Siehe Kapitel 4.2.2 auf Seite 33.

3. Nun genügt es, die Vorübersetzen-Funktion von Anaphraseus zu starten⁶⁴. Gegebenenfalls müssen einige auf Grund von unterschiedlichen Formatierungen abweichende Segmente händisch bearbeitet werden.
4. Zum Schluss erhält der Übersetzer eine zweisprachige Datei, die er »unbereinigt« lässt und mit Writer ins MS-Word DOC-Format abspeichert.

Okapi Framework: Rainbow Sofern man über Rainbow ein Projekt von Dateien im standardisierten XLIFF-Format (»XLIFF Generic«) angelegt und dieses Projekt mit OmegaT übersetzt hat, genügt es, die Zieldateien in OmegaT wie oben beschrieben zu erstellen. Die Rückkonvertierung mit Rainbow ist dazu nicht notwendig, weil man dadurch zum ursprünglichen einsprachigen Format zurückkehren würde.

6.3 Pflege des Datenbestands

Nach Abschluss eines Übersetzungsprojektes ist es sinnvoll, die vorhandenen Ressourcen (TM, Glossare) auf Fehler zu überprüfen und mögliche Mängel zu beseitigen, damit sie bei weiteren Projekten reibungslos eingesetzt werden können. Dennoch verfügen die beiden untersuchten TM-Systeme über keine oder sehr eingeschränkte Funktionen: Die Dateien müssen meist händisch in einem gewöhnlichen Texteditor bearbeitet werden (mit Hilfe der gängigen Suchen/Ersetzen-Funktion). In Bezug auf die Pflege von Translation Memories bietet sich als Alternative die Verwendung vom bereits erwähnten Programm Olifant.

Okapi Framework: Olifant Es folgt ein Auszug der Operationen, die sich mit diesem Programm an TM-Dateien durchführen lassen:⁶⁵

- Import/Export in/aus verschiedenen TM-Formaten.
- TU nach zahlreichen Kriterien markieren (AS- bzw. ZS-Segment ist leer; AS- und ZS-Segment sind gleich; TU sind mehrfach vorhanden u.v.m.).
- Globales Suchen/Ersetzen.
- Ausgangs- mit Zielsprache vertauschen.
- TM- und TU-Attribute verändern/ergänzen.
- Exakte und unscharfe Treffer im Vergleich zu einem vorgegebenen Segment.

⁶⁴ Siehe Kapitel 4.3 auf Seite 39.

⁶⁵ Eine ausführliche Beschreibung kann auf der Projektseite im Internet nachgelesen werden: *Olifant Help - Welcome*. URL: <http://okapi.sourceforge.net/Release/Olifant/Help/index.html> (besucht am 13.09.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYx9ipC>.

- TU spalten/zusammenfügen.
- TU-Inhalt bearbeiten.

6.4 Zusammenfassung

Hinsichtlich der Qualitätsprüfung bieten weder OmegaT noch Anaphraseus zufriedenstellende Funktionen an. Gerade in Bezug auf die Terminologieprüfung weisen beide Systeme noch Entwicklungslücken auf. Lediglich die Tagprüfung in OmegaT ist ausgereift und brauchbar.

Bei der Erstellung der Zieldateien hingegen bieten beide Systeme – in Kombination mit externen Tools wie dem Okapi Framework und auch durch ihre programminternen Funktionen – vielseitige Möglichkeiten an, die auf die meisten Auftragsituationen abgestimmt sind.

Bezüglich der Pflege des Datenbestands sind OmegaT und Anaphraseus unzureichend ausgestattet. Dazu ist die Heranziehung eines externen Tools wie Okapi Olifant unerlässlich.

7 Nichtfunktionale Merkmale: Effizienz

Nachdem im zweiten Teil dieser Arbeit der Frage nachgegangen wurde, über *welche* Funktionen die freien TM-Systeme verfügen, wird in diesem Kapitel untersucht, *wie effizient* diese Funktionen sind. Im Rahmen ihrer Studie stellte Lagoudaki fest, dass die Teilnehmer der Leistungsfähigkeit der eingesetzten TM-Systeme große Wichtigkeit beimessen und dass sie ihnen sogar gegenüber dem gesamten gebotenen Funktionsumfang vorrangig erscheint:

»Efficiency in TM systems refers to how well they perform their functions and at what cost in terms of computer resources. Our survey showed that each group of TM users had different priorities in terms of efficiency. [...] The majority of our respondents, who were translators, attached great importance to the efficiency with which the system leveraged existing linguistic assets. In fact, the majority of TM users seemed to be interested more in the performance of the tool in terms of matching, than in the variety of features that a TM software package may offer. Efficient match recall and precision appeared to be their primary concern.« (Lagoudaki 2008: 179 f.)

7.1 Suchalgorithmus

7.1.1 Ähnlichkeitsmessung

Edit Distance Bei den meisten TM-Systemen bildet die sogenannte *Edit Distance*⁶⁶ die Grundlage der eingesetzten Suchtechnologien. Es handelt sich um einen Algorithmus, mit dem sich Zeichenketten vergleichen lassen und der eine Zahl ausgibt, die die Summe aller Editierschritte – Einfügen, Löschen, Ersetzen – darstellt, die minimal notwendig sind, um Einheit für Einheit von einer Zeichenkette zur anderen zu kommen. Dabei erhöht jeder Editierschritt die Gesamtsumme um einen Punkt (vgl. Trujillo 1999: 66). Diese Distanz $lev(A, B)$ zwischen den Zeichenketten A und B wird normalisiert, indem sie durch die Länge der längsten Zeichenkette dividiert wird. Der **Ähnlichkeitswert** (»*similarity*«) wird schließlich durch folgende Formel errechnet (vgl. ebd.: 63):

$$100 \times \left(1 - \frac{lev(A, B)}{\max(|A|, |B|)} \right)$$

⁶⁶ In der Literatur findet man ebenfalls die Benennung *Levenshtein Distance*, nach ihrem Erfinder, dem russischen Wissenschaftler Levenshtein.

Beispiel 1 (Zeichenketten = Wörter)

0. Wecker
1. Wacker → Ersetzen (+1)
2. Wafker → Ersetzen (+1)
3. Waffer → Ersetzen (+1)
4. Waffel → Ersetzen (+1)
5. Waffeln → Einfügen (+1)

Dies ergibt eine **Distanz** von 5 und einen **Ähnlichkeitswert** von $100 \times (1 - \frac{5}{6}) = 16\%$.

Beispiel 2 (Zeichenketten = Sätze)

0. Das Leben ist schön.
1. Das Essen ist schön. → Ersetzen (+1)
2. Das Essen ist gut. → Ersetzen (+1)

Dies ergibt eine **Distanz** von 2 und einen **Ähnlichkeitswert** von $100 \times (1 - \frac{2}{4}) = 50\%$.

Würfelkoeffizient Beim sogenannten *Dice Coefficient* werden zwei Zeichenketten so miteinander verglichen, dass man die Zahl der gemeinsamen Elemente durch die Summe aller Elemente dividiert. Die Formel lautet (vgl. Trujillo 1999: 62):

$$\frac{2|A \cap B|}{|A| + |B|}$$

Das Ergebnis dieser Berechnung ist eine Zahl zwischen 0 und 1.⁶⁷

7.1.2 Trefferquoten

An Hand einer Reihe einfacher Tests wurde das Verhalten des in OmegaT bzw. Anaphra-seus eingesetzten Suchalgorithmus untersucht. Hierbei ist es erforderlich, die verschiedenen Trefferkategorien durch klare Definitionen auseinanderzuhalten:

repetition Segment, das mehrmals im Ausgangstext mit jeweils identischer Form und identischem Inhalt vorkommt.

exact match Segment, das unter einer identischen Form und mit einem identischen Inhalt in der TM vorhanden ist. Je nach System wird es auch *perfect match* genannt.

100% match Segment, das – in der Regel – mit einem identischen Inhalt in der TM vorhanden ist.

fuzzy match Segment, das unter einer abweichenden Form und/oder mit einem abweichenden Inhalt in der TM vorhanden ist. Die Ähnlichkeit wird bei den meisten TM-Systemen in Prozenten ausgedrückt: Je höher die Trefferquote, umso »ähnlicher« sind sich AS-Segment und TU.

⁶⁷ Den Prozentsatz erhält man, indem man diese Zahl mit 100 multipliziert.

Diese Begriffe mögen noch so genau definiert werden, der nächste Test zeigt, dass die verschiedenen TM-Systeme unterschiedliche Ergebnisse liefern, was darauf zurückzuführen ist, dass jeweils mehr oder weniger komplexe Suchtechnologien eingesetzt werden.

OmegaT Im Test auf der nächsten Seite wurden die Ergebnisse unterschiedlicher *tokenizer*-Konfigurationen zusätzlich angeführt. Beim *tokenizer* handelt es sich um ein von Alex Bulochik entwickeltes OmegaT-Plug-in, das den Suchalgorithmus mit sprachenabhängigen »linguistischen« Informationen speist: Jedes Wort wird in seine Bestandteile zerlegt – Präfix-Stamm-Suffix – und gegebenenfalls mit internen Worttabellen abgeglichen. Dieser als *stemming* bekannte Vorgang ermöglicht es, morphologisch ähnliche Wörter miteinander zu verbinden (vgl. ebd.: 64). Das OmegaT-*tokenizer* greift dabei auf die leistungsfähigen, mehrfach eingesetzten Lucene-Programmbibliotheken zurück. Um korrekt zu funktionieren, muss das *tokenizer* der Ausgangssprache entsprechend konfiguriert werden. Zur Auswahl stehen derzeit zehn Sprachen mit der Lucene- und fünfzehn Sprachen mit der Lucene-Snowball-Bibliothek.

Der Test zeigt, dass OmegaT die Ausgangssegmente 1 bis 7 als *fuzzy matches* betrachtet. Mit einer Trefferquote von 100% gilt Ausgangssegment 8, bei dem – abweichend zur in der TM vorhandenen TU – ein Buchstabe großgesetzt wurde, als *100% match*, obwohl beide Segmente eigentlich nicht identisch sind. Dies muss bei einem entsprechend eingestellten Editierverhalten (siehe Kapitel 5.5.1) unbedingt beachtet werden! Als einziger *exact match*, der beim Vorübersetzen tatsächlich durch die TM-Entsprechung ersetzt wird, gilt für OmegaT das Ausgangssegment 9. Die etwas irreführende Bezeichnung »*100% match*« veranlasste die Entwickler zu folgender Aussage:

»I have under the eyes a 100/100/100% match where the capitalization isn't the same between the source document and the match.

So, in OmegaT, 100% doesn't mean ›exactly the same‹.

Note that the situation hasn't changed since 1.6. I remember commenting we would need a kind of ›101%‹, or lower the current 100% to 99%, to be able to tell the difference.«⁶⁸

68 Aus: Didier Briel (2009): RE: [OmT] This 100% match isn't a 100% match... URL: <http://tech.groups.yahoo.com/group/OmegaT/message/14820> (besucht am 15.09.2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 14. 07. 2009. Archivierte URL verfügbar: <http://www.webcitation.org/5jsYY5BN7>.

Test 1

Bei diesem Test wurden verschiedene Ausgangstextsegmente mit einem in der TM vorhandenen Segment verglichen:

- | | | |
|---|----------------|---|
| 0 | TM-Segment | Das Auto des Ministers fuhr 200 m weiter und stoppte. |
| 1 | Weibliche Form | Das Auto der <u>Ministerin</u> fuhr 200 m weiter und stoppte. |
| 2 | Plural | Das Auto <u>der Minister</u> fuhr 200 m weiter und stoppte. |
| 3 | Formatierung | Das Auto des <u>Ministers</u> fuhr 200 m weiter und stoppte. |
| 4 | Gegenwart | Das Auto des Ministers <u>fährt</u> 200 m weiter und <u>stoppt</u> . |
| 5 | Perfekt | Das Auto des Ministers <u>ist</u> 200 m <u>weitergefahren</u> und <u>hat gestoppt</u> . |
| 6 | Satzzeichen | Das Auto des Ministers fuhr 200 m weiter und stoppte_ |
| 7 | Zahl | Das Auto des Ministers fuhr <u>180</u> m weiter und stoppte. |
| 8 | Großschreibung | Das Auto des Ministers fuhr 200 <u>M</u> weiter und stoppte. |
| 9 | Identisch | Das Auto des Ministers fuhr 200 m weiter und stoppte. |

Dies führte zu folgenden Ergebnissen:

Programm	OmegaT mit »tokenizer«			Anaphraseus		Trados 2007	
	Lucene	Snowball	vorübersetzt?		vorübersetzt?		vorübersetzt?
AS 1	71/77/90%	77/77/90%	nein	75%	nein	92%	nein
AS 2	85/77/90%	88/77/90%	nein	92%	nein	85%	nein
AS 3	100/100/90%	100/100/90%	nein	100% (grün)	ja	100%	ja
AS 4	85/77/90%	88/77/90%	nein	90%	nein	87%	nein
AS 5	42/60/77%	60/60/77%	nein	90%	nein	70%	nein
AS 6	100/100/95%	100/100/95%	nein	100% (grau)	ja	99%	nein
AS 7	100/100/95%	100/100/95%	nein	94%	nein	100%	ja
AS 8	100/100/100%	100/100/100%	nein	100% (grün)	ja	83%	nein
AS 9	100/100/100%	100/100/100%	ja	100% (grün)	ja	100%	ja

Tabelle 7.1: Vergleich der Trefferquoten bei OmegaT, Anaphraseus und Trados

Anaphraseus Dieses System betrachtet die Segmente 3, 6, 8 und 9 als *100% matches*, wobei die grafische Benutzeroberfläche die drei Segmente 3, 8 und 9 in einem grünen und Segment 6 in einem farblosen Kasten darstellt. Dies erschwert die Unterscheidung zwischen *100% matches* und *exact matches*, da man sich die Frage stellen muss, ob Segment 6 (fehlender Punkt am Satzende) doch nicht als *exact matches* gilt. Dies scheint vom Vorübersetzen-Test widerlegt zu werden, denn alle vier genannten Segmente werden – unabhängig von der ausgegebenen farblichen Unterscheidung – automatisch übersetzt. Dass Anaphraseus das Segment 3 nicht als *fuzzy match* betrachtet, hängt mit der auf Seite 53 bereits erwähnten programmiersprachenseitigen Einschränkung zusammen, da StarBasic Formatierungen

nicht beachtet. Segmente 6 und 8 letztendlich als *exact matches* gelten zu lassen, stellt eine eindeutige Quelle für Übersetzungsfehler dar.

7.1.3 Precision – Recall

In der Forschungsliteratur werden zwei weitere Kriterien zur Bewertung der Effizienz von Suchalgorithmen angeführt:

» **Precision:** This is the ratio of relevant items retrieved over all the items retrieved. For example, if a system retrieves five sentences from TM, but only one is actually useful to the translator, whatever that may be, the precision is 20%.

Recall: This is the ratio of relevant items retrieved over all relevant items in TM. If two relevant sentences are retrieved, but there are four relevant sentences in TM, then recall is 50%.

In some cases increasing one of these measures means decreasing the other. For example, adding ordering information to the search might improve precision, but it might also decrease recall. Similarly, using a thesaurus can improve recall but it may decrease precision.« (Trujillo 1999: 63)

In Zusammenhang mit der in den meisten TM-Systemen einstellbaren Schwelle (Trefferquote in Prozenten), unter der bestimmte *fuzzy matches* nicht angezeigt werden sollen⁶⁹, führt Bowker eine weitere Unterscheidung zwischen »*silence*« und »*noise*« an:

»If the sensitivity threshold is set too high (e.g., a minimum of 95 percent similarity), there is a risk that the TM system will produce ›silence‹: potentially useful partial matches will not be retrieved. However, if the sensitivity threshold is set too low (e.g., a minimum of 10 percent similarity), there is a risk that the TM system will produce ›noise‹: the suggested translations that are retrieved will be too different from the new source-text segment and therefore will not be helpful.« (Bowker 2002: 99)

Um Aussagen über die potenzielle Nützlichkeit der vom jeweiligen TM-System unterbreiteten Übersetzungsvorschläge treffen zu können, wurde folgender von Bowker (vgl. 2002: 107) inspirierter Test durchgeführt:

⁶⁹ Dieser Schwellenwert ist in Anaphraseus einstellbar und liegt standardmäßig bei 75%. Bei OmegaT hingegen lässt sich dieser Wert nicht verändern. Dafür werden in OmegaT jeweils nur die fünf besten Treffer angezeigt.

Test 2

Bei diesem Test wurde ein neues Segment (NS) mit zwei in einer TM gespeicherten Übersetzungseinheiten (TU 1 und 2) verglichen.

Neues Segment Die Wörter auf diesem Band klingen sehr wichtig.

TU 1 Die Mörder auf diesem Band klingen sehr richtig.

TU 2 Das Wort_ auf diesem Band klingt sehr wichtig.

Dies führte zu folgenden Ergebnissen:

	Edit Distance	Similarity	Dice-Koeff. (Wörter)	OmegaT				Anaphraseus	
				Lucene		Snowball			
				Trefferquote	Rang	Trefferquote	Rang	Trefferquote	Rang
NS/TU1	3	94%	75%	66/75/87%	2.	75/75/87%	1.	68%	2.
NS/TU2	7	86%	63%	100/62/81%	1.	75/62/81%	2.	75%	1.

Tabelle 7.2: Messung der *Precision* bei OmegaT und Anaphraseus in Bezug auf die TM

TU 1 weist gegenüber TU 2 eine größere formelle Ähnlichkeit zum neuen Segment auf, was alle drei angegebenen Ähnlichkeitswerte⁷⁰ belegen. Dennoch ist TU 2 sinngemäß am nächsten zum neuen Segment – da es sich um den gleichen Satz im Plural handelt – und somit auch für den Übersetzer »nützlicher«. Mit dem auf die deutsche Sprache eingestellten *tokenizer* betrachtet OmegaT die Token »Die«, »Wörter« und »klingen« als abweichende Formen der jeweiligen in der TM gespeicherten Wörter. Auf dieser Grundlage errechnet es die erstere Zahl, die für die Reihung der Übersetzungsvorschläge entscheidend ist: Demzufolge wurde die »nützlichere« TU 2 als erster Vorschlag angegeben, was eine Erhöhung des *Precision*-Wertes des Systems bedeutet.⁷¹

Bei diesem Test legt Anaphraseus ebenfalls eine gute *Precision* an den Tag, da es die »richtige« TU vorschlägt.⁷²

⁷⁰ Zur Erklärung der drei Werte *Edit Distance*, *Similarity* und *Dice*-Koeffizient, siehe Kapitel 7.1.1 auf Seite 73.

⁷¹ Ein hier nicht beschriebener Test mit einer älteren Version von OmegaT führte zum gegensätzlichen Ergebnis. Der vorliegende Test zeigt auch, dass die Snowball-Programmbibliothek ein schlechteres Ergebnis erzielt als die andere. Dies hängt jedoch vom jeweiligen Ausgangstext ab.

⁷² Anaphraseus wendet ein einfaches *stemming*-Verfahren an: Alle im neuen Segment befindlichen Wörter werden – ab einer Wortlänge von fünf Buchstaben – um den letzten Buchstaben gekürzt. Werden diese gekürzten Wörter in der TU gefunden, wird ein Zähler um die Länge dieser Wörter erhöht. Die von Anaphraseus errechnete Trefferquote ergibt sich anschließend aus dem Verhältnis zwischen der Gesamtlänge der gefundenen Wörter und der Länge der TU. In einer privaten E-Mail-Korrespondenz bestätigte der für die Entwicklung von Anaphraseus Hauptverantwortliche Oleg Tsygany diese aus dem Betrachten des Quellcodes gewonnene Erkenntnis.

7.1.4 Glossare

Die zur Durchsuchung der TM verwendeten Algorithmen werden bei den untersuchten Systemen auch auf die Glossare angewandt. Der folgende Test zeigt mit welcher Effizienz:

Test 3

	Token im Text	Glossareinträge	OmegaT		Anaphraseus
			Lucene	Snowball	
Deutsch					
Partizip II	gesprungen	springen	nein	nein	nein
Gegenwart	springst	springen	nein	ja	nein
Substantivierung	Voraussetzung	voraussetzen	nein	ja	nein
Kompositum	Terminologearbeit	Terminologie	nein	nein	nein
	Terminologearbeit	Arbeit	nein	nein	nein
Pluralform	Wörter	Wort	ja	ja	nein
	Morde	Mord	ja	ja	nein
	Niederungen	Niederung	ja	ja	nein
Genitiv	Dorfes	Dorf	ja	ja	nein
Dativ Plural	Göttern	Gott	ja	ja	nein
Französisch					
Partizip II	sauté	sauter	nein	ja	nein
	couru	courir	nein	nein	nein
Pluralform	mots	mot	ja	ja	nein
	travaux	travail	nein	nein	nein
	bas-fonds	bas-fond	ja	ja	nein
Adverb	anticonstitutionnellement	anticonstitutionnel	ja	ja	nein
»Fallen« (für die	soit	être	nein	nein	nein
	soit	soie	nein	nein	nein
Messung der Precision)	joignons	joindre	nein	nein	nein
	joignons	oignon	nein	nein	nein

Tabelle 7.3: Messung des *Recall* bei OmegaT und Anaphraseus in Bezug auf die Glossare

Der Test zeigt, dass Anaphraseus Token, die von der im Glossar gespeicherten Grundform abweichen, nicht erkennt. Hingegen zeigt OmegaT – in Kombination mit dem *tokenizer*⁷³ – in beiden Sprachen insbesondere bei den Pluralformen relativ gute *Recall*-Werte, wobei die Snowball-Programmibibliothek besser abschneidet⁷⁴. Gleichzeitig wurde versucht, die *Precision* an Hand von »Fallen« zu messen: OmegaT hat das »falsche« Wort zwar nicht erkannt, aber das »richtige« auch nicht.

73 In einem hier nicht angeführten Test, bei dem auf die Verwendung des *tokenizer* verzichtet wurde, fand OmegaT auch keine Glossareinträge wieder.

74 Im Gegensatz zum vorherigen Test der TM-Suchgenauigkeit.

7.2 Analyse

Bei dem auf dieser Seite präsentierten Test der Analysefunktionen von OmegaT wurde die von Massion (2005: 137) und Seewald-Heeg (2005: 15) in Zusammenhang mit kommerziellen Anwendungen bereits gemachte Feststellung bestätigt, nach der es je nach verwendetem System zu erheblichen Ergebnisunterschieden kommt.

Test 4

Bei diesem Test wurde ein Projekt mit 44 MS-Word 2003 DOC-Dateien (44 OpenOffice.org ODT-Dateien für das OmegaT-Projekt, nach Konvertierung aus dem DOC-Format) angelegt und anschließend von Trados Translator's Workbench und OmegaT analysiert. In beiden Fällen wurde die gleiche *level 1*-TMX-Datei verwendet.

Programm	Gesamt		Wiederholungen		exact matches	
	Wörter	Anschläge ^a	Wörter	Anteil	Wörter	Anteil
OmegaT	27 948	171 868	4233	15%	20 850	75%
Trados	27 242	167 080	326	1%	22 627	83%

Tabelle 7.4: Analysevergleich OmegaT-Trados

^a Anschläge ohne Leerzeichen

Eine Erklärung für diese Diskrepanz zwischen den Analyseergebnissen liegt darin, dass die TM-Systeme jeweils andere Segmentierungsregeln anwenden, sowie darin, dass die Begriffe »*repetition*«, »*100% match*« und »*exact match*« von der jeweiligen Anwendung anders ausgelegt werden. Selbst wenn dies in Bezug auf eine grobe Einschätzung der anstehenden Übersetzungsarbeit nicht zwangsläufig hinderlich ist, kann dies bei der Erstellung eines Kostenvoranschlags oder einer Rechnung zu größeren Abweichungen führen, die unbedingt mit dem Auftraggeber vorabzuklären sind. Des Weiteren erfolgt keine Einteilung der Segmente nach Trefferquoten.

7.3 Alignment

Der nächste Test zeigt, dass die besten Ergebnisse im automatischen Alignment durch die Verwendung von bligner erzielt werden. Wie in Kapitel 4.5 beschrieben, erfordert bligner allerdings eine genauere Arbeit an den Vorlagen (Absatzstruktur) und am Skript selbst (Segmentierungsregeln, Sprachen) als es mit TagAligner der Fall ist, bei dem es genügt, die Vorlagen ins XHTML-Format (über das Zwischenformat Reintext) zu konvertieren. Je nach ge-

wünschter TM-Qualität und zur Verfügung stehender Zeit, kann sich der Übersetzer für die eine oder andere Lösung entscheiden:

Test 5

Bei diesem Test wurden zwei kleinere MS-Word DOC-Dateien (deutsches Original mit 1062 und französische Übersetzung mit 1314 Wörtern) aus einem Ferienkatalog konvertiert und aliniert. In diesen Texten kommen alle drei von Somers beschriebenen Schwierigkeiten vor: Sie enthalten Abkürzungen, Originalsätze wurden stellenweise durch mehrere Sätze übersetzt und die Satzreihenfolge der Übersetzung weicht geringfügig von derjenigen des Originals ab. Für den Test mit bligner wurden die Dateien vom DOC- ins Reintextformat konvertiert und bei TagAligner wurden die gerade erstellten Reintextdateien zusätzlich ins XHTML-Format exportiert, Beides mit Hilfe von OpenOffice.org Writer.

Programm	Falsch		Unvollständig		Richtig		Gesamt	
	TU	Anteil	TU	Anteil	TU	Anteil	TU	Anteil
TagAligner (Standardalgorithmus)	37	16%	22	10%	170	74%	229	100%
Bligner (ohne Segmentierung)	41	21%	0	0%	156	79%	197	100%
Bligner (mit Segmentierung)	14	9%	0	0%	149	91%	163	100%

Tabelle 7.5: Alignment-Ergebnisse TagAligner/Bligner

7.4 Zeit- und Ressourcenbeanspruchung

Die Effizienz eines TM-Systems – und einer Software im Allgemeinen – wird nicht nur an Hand dessen gemessen, wie präzise es die Aufgaben erledigt: Es soll auch *schnell* arbeiten und dabei so wenig Rechnerzeit wie möglich beanspruchen.

Eine hier nicht angeführte Testreihe zeigte, dass sowohl OmegaT als auch Anaphraseus gleichermaßen schnell und ressourcensparsam sind, wobei letzteres TM-System ab einer bestimmten TM- bzw. Glossargröße (etwa mit über 100 000 TU bzw. Einträgen) Zeichen einer gewissen Verlangsamung an den Tag legte. Des Weiteren ist OmegaT im Stande, umfangreiche Dateien zu verarbeiten, ohne dabei abzustürzen⁷⁵. Bei Anaphraseus ist allerdings die Importfunktion einer TMX-Translation Memory nicht sehr leistungsstark und nimmt mitunter recht viel Zeit in Anspruch.

⁷⁵ Bei einem »Stresstest« konnte OmegaT eine Tabellenkalkulation-Datei von 100 KB mit 1000 x 18 Zellen in anderthalb Minuten Zeit laden. Das Laden der gleichen Datei von Trados TagEditor (Version 2007 unter Windows XP) wurde nach zehnminütiger Wartezeit abgebrochen. Ähnliche Ergebnisse zeigten Tests mit Textdokumenten und Präsentationen.

7.5 Formatierungen

OmegaT wurde auf seine Fähigkeit getestet, nachstehende Dokument- und Textformatierungen korrekt zu erkennen und ins Zieldokument zu übertragen⁷⁶:

- Fett, kursiv, unterstrichen
- Formatvorlagen (wie Titel usw.)
- Tabellen
- Bildüber- und unterschritten
- Textverweise (Querverweise, Fuß- und Endnoten, Indexeinträge)
- Automatische Felder (wie Datum usw.)

Alle Formatierungen wurden fehlerfrei in den Zieltext übernommen, sofern die entsprechenden Tags vom Übersetzer beachtet wurden.

Wurde ein Formatierungstag falsch gesetzt, hindert dies die Erstellung des Zieldokuments zwar nicht, aber die Datei wird vom Textverarbeitungsprogramm als fehlerhaft (korrupt) erkannt und nicht geöffnet. Um dies zu verhindern, empfiehlt sich daher die Verwendung der Funktion zur Prüfung der Tags⁷⁷.

⁷⁶ Aus den auf Seite 53 angegebenen Gründen ist Anaphraseus noch nicht im Stande, Formatierungen zu behandeln; daher wurde es keinem entsprechenden Test unterzogen. In einer privaten E-Mail-Korrespondenz bestätigte Oleg Tsygany gegenüber dem Verfasser dieser Arbeit, dass die Formatierungsproblematik zu den höchsten Prioritäten des Projektes gehöre.

⁷⁷ Siehe Seite 67.

8 Nichtfunktionale Merkmale: Benutzbarkeit

Bei der Evaluierung eines TM-Systems nach dessen Benutzbarkeit soll versucht werden, auf folgende einfache Frage zu antworten: *Wie benutzerfreundlich* ist das System? Hierbei spielen allerdings subjektive Einrückte eine nicht zu unterschätzende Rolle.

8.1 Anpassbarkeit

OmegaT Das äußere Erscheinen der grafischen Benutzeroberfläche von OmegaT wird weitgehend von der Java-Umgebung bestimmt, in der das Programm läuft. Trotzdem lassen sich folgende Merkmale vom Benutzer einstellen:

- Verwendete Schriftarten und -größe: Sind für das Editorfenster und teilweise auch für die Dialogfenster einstellbar.
- Farben: Die im Editor zur visuellen Hervorhebung von übersetzten und nichtübersetzten Segmenten verwendeten Farben lassen sich nicht einstellen, höchstens deaktivieren.
- Sprachen: Die Software ist in 22 lokalisierten Versionen verfügbar⁷⁸.
- Tastenkombinationen: Bis auf die Taste zum Navigieren von einem Segment zum nächsten⁷⁹ lassen sich die voreingestellten Tastenkombinationen nicht ohne größeren Aufwand anpassen⁸⁰.

Anaphraseus Dadurch, dass Anaphraseus lediglich ein Plug-in von Writer ist, ist es an die grafischen Konventionen des Hauptprogramms gebunden. Daher lassen sich – bis auf

⁷⁸ Darunter auch Deutsch. Die Herausgabe einer Stableversion hängt sogar davon ab, ob die GUI, die Hilfedateien und die Reintext-Benutzeranleitung in allen Sprachen vollständig lokalisiert wurden. Dadurch kann eine Betaversion, die bis auf die Dokumentation alle Code-Verbesserungen berücksichtigt, trotzdem durchaus lauffähig sein. Sofern man eine andere GUI-Sprache als die betriebssystemweit geltende wünscht, erfolgt dies über die Bearbeitung des Startskripts.

⁷⁹ Die Eingabetaste ist voreingestellt. Diese kann aber durch die Tabulatortaste ersetzt werden.

⁸⁰ Die dazu benötigten Schritte, die eine gewisse Anpassung des Quellcodes erfordern, können auf folgender Seite nachgelesen werden: <http://tech.groups.yahoo.com/group/OmegaT/message/13117> (Archivierte URL: <http://www.webcitation.org/5juMGpM7q>). Eine wünschenswerte Verbesserung wäre die Vereinfachung dieser Schritte z.B. durch die Einbindung einer zusätzlichen Funktion im Menü.

die Aktivierung der -Taste zum Navigieren von einem Segment zum nächsten – nach jetzigem Entwicklungsstand keine benutzerspezifischen Anpassungen vornehmen⁸¹

8.2 Erlernbarkeit

8.2.1 Dokumentation

Ein wichtiger Bestandteil jeder Software stellt die beigelegte Produktdokumentation dar, die als Bedienungsanleitung, als über das Menü abrufbare Hilfedatei oder unter anderen Formen verfügbar sein sollte, um die Bedienung und Funktionsweisen der Software erlernen zu können. Über die Wichtigkeit der Dokumentation auch bei freier Software schreibt Stallman: »*Dokumentation ist ein wesentlicher Teil jedes Softwarepakets; wenn ein wichtiges freies Softwarepaket nur ohne freie Anleitung vorliegt, ist das eine gravierende Lücke. Bis heute bleiben uns leider viele solcher Lücken.*« (Stallman 2000)

Das OmegaT-Team zählt engagierte *Technical Writer* und Lokalisierer, die die Produktdokumentation laufend ergänzen und übersetzen⁸², zu seinen Mitgliedern. Sollte der Benutzer dennoch Schwierigkeiten haben, die er an Hand der Programmhilfe nicht lösen kann, kann er sich an den Online-Support wenden (siehe unten).

Bei Anaphraseus hingegen ist die Dokumentation so gut wie nicht vorhanden. Dies ist ein Mangel, der hoffentlich bald behoben wird.

8.2.2 Einfachheit

Ein Kriterium zur Bewertung der Erlernbarkeit einer Software ist selbstverständlich auch deren Einfachheit. In diesem Zusammenhang erwähnt Lagoudaki, dass die Erwartungen unterschiedlicher Benutzerkategorien sich widersprechen können, wenn es um die Ausgewogenheit zwischen Einfachheit auf der einen und Funktionsumfang auf der anderen Seite geht:

»The real conflict lies in the need for enhanced and rich functionality on one hand and the need for simplicity and ›pared-down‹ functions on the other. In other words, when the developer attempts to please the first user by adding additional functionality, the satisfaction of the second user's need for simplicity of is likely to decrease. Based on our

81 In einer privaten E-Mail-Korrespondenz sendete der Programmentwickler dem Verfasser dieser Arbeit eine Testversion, die in zehn Sprachen lokalisiert ist und mit der die Sprachenauswahl unabhängig vom Hauptprogramm Writer vorgenommen werden kann. Des Weiteren verfügt diese neue Version über die Möglichkeit – ähnlich wie bei Wordfast –, die Werkzeugliste zusammenzuklappen.

82 Siehe Fußnote 78 auf der vorherigen Seite.

investigation of user needs, conflicting needs usually result from differences in the use of context by different user groups (e.g. translators, terminologists, project managers), from a variation in levels of experience both in translation and in using the system, and finally from differences in the users' level of computer usage competence.« (Lagoudaki 2008: 183)

Sowohl bei OmegaT als auch bei Anaphraseus ist die Software bewusst so einfach gehalten wie möglich. Dieser Grundsatz hindert die jeweiligen Entwickler dennoch nicht, bestimmte Anforderungen an Leistungsmerkmale, die Benutzer über das *Feature Request System* stellen, im Laufe der Zeit und je nach geschätzter Priorität ins Programm zu integrieren.

Des Weiteren ist Einfachheit auch dadurch bedingt, ob die Software gewisse ungeschriebene Normen im Aufbau der Menüs berücksichtigt oder nicht. In dieser Hinsicht halten sich beide untersuchten TM-Systeme an diese Standards. Außerdem sei noch in dieser Stelle angemerkt, dass zwischen dem Starten von OmegaT oder Anaphraseus und dem tatsächlichen Beginn des Übersetzungsvorgangs nur weniger Schritte liegen.⁸³

Etwas weniger benutzerfreundlich ist bei OmegaT allerdings das Ausführen der Vorübersetzen-Funktion, weil es dazu notwendig ist, das Programm über Kommandozeile zu starten, was wenig computerversierte Benutzer abschrecken mag. Außerdem lässt sich die Größe von Übersetzungssegmenten aus dem Stehgreif nicht ohne den Umweg über die Veränderung des Ausgangstextes oder die Anpassung der Segmentierungsregeln bewältigen.

8.2.3 Support

Das Produktsupport gehört bei den meisten kommerziellen Systemen zur Selbstverständlichkeit und ist nicht selten einer der wichtigsten Gründe bei der Wahl zum Kauf. Bei den freien TM-Systemen, die *per definitionem* kostenlos sind, stellt sich eine ähnliche Frage. Diesbezüglich schreibt Hüttenegger:

»Die [Unterstützung durch den Hersteller] ist bei Open Source in anderer Form und in sehr unterschiedlicher Qualität gegeben (wie ja auch bei kommerzieller Software). In vielen Fällen gibt es sehr aktive ›Communities‹ und bemühte Entwickler zu einem Produkt. In anderen Fällen gibt es vielleicht nur sehr wenige Personen mit dem notwendigen Know-how und es wird versucht, die Kosten der Entwicklung mit kostenpflichtigem Support abzudecken [...] .« (Hüttenegger 2006: 5)

Bei dem jungen Produkt Anaphraseus beschränkt sich das Team derzeit auf drei Personen, wodurch die Supportmöglichkeiten sich in Grenzen halten.

⁸³ In OmegaT wird ein neues Projekt nach nur drei Dialogfenstern angelegt. Im Vergleich dazu sind bei Déjà Vu 11 und bei Transit gar 14 Konfigurationsschritte notwendig (vgl. Massion 2005: 40).

Das bereits etwas länger existierende OmegaT hingegen erfreut sich einer regen Benutzer- und Entwicklergemeinschaft, die unablässig auf Fragen in der Mailing Liste antwortet, wie die in Abbildung 8.1 dargestellte Nachrichtenstatistik zeigt⁸⁴.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2009	324	298	352	155	396	517	232	270	264			
2008	161	270	319	369	220	322	115	390	280	296	269	288
2007	178	275	188	110	185	183	192	233	43	142	68	185
2006	341	242	298	220	176	289	241	209	237	320	566	401
2005	223	69	103	290	239	405	187	140	170	210	349	247
2004			258	134	107	104	42	90	100	181	113	83

Abbildung 8.1: Anzahl der Nachrichten auf der OmegaT-Mailing Liste

An dieser Stelle soll betont werden, dass es sich bei beiden untersuchten freien TM-Systemen um Softwares handelt, die weitgehend von Übersetzern selbst entwickelt werden; Anwender, die früher nie einen Blick hinter die »Kulissen« des Quellcodes gewagt hätten, beginnen nun selbst zu programmieren. Die Tatsache, dass Benutzer von TM-Systemen oft gleichzeitig gute bis herausragende Computerkenntnisse besitzen, wird von Lagoudakis Studie bestätigt: Von den 874 Studienteilnehmern (von denen 90% Übersetzer waren) benutzten 82,5% mindestens ein TM-System; 64% sagten über »gute« und 30% über »sehr gute« Computerkenntnisse (Kompetenz) zu verfügen (vgl. Lagoudaki 2006: 11, 14 und 18).

8.3 Erweiterbarkeit

Gerade die Fähigkeit, problemlos erweiterbar zu sein, ohne dass man dabei auf den Eigentümer des nichtverfügbaren Codes angewiesen ist, zeichnet freie Softwares aus. Diese Grundeigenschaft ist in den von Stallman formulierten Freiheiten 1 und 3 sowie in der *General Public Licence* als deren Rechtsgrundlage verankert⁸⁵.

Ein Beispiel für die Erweiterbarkeit von OmegaT liefert das *tokenizer*-Plug-in: Es wurde am Rande des Hauptprojektes entwickelt und jedem Anwender steht es frei, dieses zu benutzen oder nicht.

⁸⁴ Als Beispiel kann eine Anfrage zur Verwendung von toxic angeführt werden, die am 27. August 2009 in der OmegaT-Liste gestellt wurde und innerhalb von 24 Stunden nicht weniger als 46 Antworten von verschiedenen Entwicklern und Benutzern erhielt!

⁸⁵ Siehe Kapitel 2.2.2 und 2.2.3 ab Seite 13.

8.4 Übertragbarkeit

Die Übertragbarkeit bezeichnet die Fähigkeit einer Software, in jeder beliebigen Rechnerumgebung – das heißt unabhängig von der gegebenen Hardware- und Betriebssystemkonfiguration – gleichermaßen zu arbeiten.

OmegaT funktioniert in einer Java-Laufzeitumgebung. Diese wird entweder mit dem Programm geliefert oder der Anwender greift auf die auf seinem Computer installierte Java-Laufzeitumgebung zurück. Java selbst läuft unter allen Betriebssystemen und auf jedem Computer. Daher gilt OmegaT als »plattformunabhängig«: Es kann sowohl unter MS Windows als auch GNU/Linux wie Mac OS X verwendet werden.⁸⁶ Dies gilt ebenfalls für Anaphraseus, da es im plattformunabhängigen Programm OpenOffice.org Writer eingebunden wird.

8.5 Zuverlässigkeit

Unter der »Zuverlässigkeit« eines Softwareproduktes versteht man dessen Fähigkeit, ohne Absturz und Fehlermeldungen zu arbeiten. Des Weiteren wird darunter die Fähigkeit verstanden, bei auftretenden Fehlern die bereits produzierten Dateien des Benutzers im gleichen Zustand wiederherzustellen.

OmegaT Das System ist in vieler Hinsicht gegen Abstürze abgesichert, da die Projekthauptdatei (die Projekt-TM) in regelmäßigen Abständen unter einer laufenden Versionsnummer gespeichert wird.⁸⁷ Dies ermöglicht gegebenenfalls eine Wiederherstellung der Übersetzungsarbeit in einen früheren Stand. Der Verfasser dieser Arbeit, der das Programm seit mehreren Monaten testet, stellte keinen Programmabsturz fest.

Anaphraseus Dieses im Hinblick auf seine Funktionen viel versprechende Programm ist derzeit noch in der Betaversion verfügbar, wodurch Absturz und Fehler nicht auszuschließen sind. Aus diesem Grund ist es für eine Produktionsumgebung noch nicht zu empfehlen. Nichtsdestoweniger beschränkten sich die vom Verfasser dieser Arbeit beobachteten Fehler auf einige wenige Meldungen. Darüber hinaus werden alle Dokumente von Writer durch die Funktion zur Dokumentenwiederherstellung weitgehend geschützt.

⁸⁶ Die Teilnehmer an Ladoudakis Studie gaben an, zu 92% ausschließlich MS Windows als Betriebssystem zu verwenden. 4% verwendeten ausschließlich Mac OS X und nur 2% GNU/Linux. Weitere 2% gaben an, mehr als nur ein Betriebssystem zu verwenden (vgl. Lagoudaki 2006: 16). Diese Zahlen beweisen einmal mehr, wie »übermächtig« proprietäre Betriebssysteme – und sonstige kommerzielle Software – sind.

⁸⁷ Wenn das Projekt händisch gespeichert oder das Programm beendet wird. Sonst alle 30 Minuten.

9 Zusammenfassung

Bilanz

Unsere Untersuchung freier TM-Systeme kommt zu dem Ergebnis, dass OmegaT und Anaphraseus über umfangreiche Funktionen⁸⁸ verfügen, die sie für den professionellen Einsatz durchaus brauchbar machen. Sie zeichnen sich außerdem durch Vielseitigkeit aus, da sie unter allen Betriebssystemen laufen können und zahlreiche Dateiformate unterstützen.

Letzteres Merkmal erlaubt es dem Übersetzer, sich weitgehend von proprietären Formaten und Anwendungen zu lösen. Wie aus Lagoudakis Studie hervorgeht, benutzen 72% der Teilnehmer TM-Systeme aus freiem Wunsch, während die restlichen 28% dies auf Verlangen ihres Auftraggebers tun. Demnach erhält ein Übersetzer einen Auftrag nicht selten deswegen, weil er dieses oder jenes System besitzt und weniger, weil er die übersetzerische Fachkompetenz aufweist.⁸⁹ Dies ist besonders für kompetente Übersetzer aus Ländern der sogenannten »Dritten Welt« relevant, die sich unter Umständen den Kauf eines teureren TM-Systems nicht leisten können und – bei vergleichbarer Leistungsfähigkeit – auf freie Systeme zurückgreifen.⁹⁰

Ausblick

Es ist in nächster Zukunft zu erwarten, dass freie TM-Systeme immer mehr auf online verfügbare Ressourcen (TM und MT) zurückgreifen werden oder sich gar vom Internet aus starten lassen.⁹¹ Eine weitere Tendenz geht in die Entwicklung von serverbasierten TM-Systemen

88 Die Tabelle auf Seite 105 im Anhang stellt die Funktionalitäten beider TM-Systeme zusammenfassend dar.

89 Diese allgemeine Beobachtung sollte dennoch mit Zahlen belegt werden, die derzeit allerdings fehlen.

90 So kämpfen zahlreiche Nichtregierungsorganisationen gegen den technologischen Rückstand (»*global information poverty*«) in diesen Ländern und stellen ihre Arbeit im Rahmen von Workshops auf Fachmessen wie die *Action week for Global Information Sharing* (AGIS) dar, die vom 21. bis 23. September 2009 in Limerick (Irland) unter der Schirmherrschaft der dortigen Universität stattfand, und bei der unter anderem OmegaT und Anaphraseus präsentiert wurden. Vgl. <http://agis09.org/index.htm> (besucht am 21. 09. 2009).

91 Hierzu hat das OmegaT-Team eine Java Webstart-Version entwickelt. Siehe <http://omegat.sourceforge.net/webstart.html> (besucht am 23. 09. 2009).

wie OpenTMS.⁹²

Auf Grund der Themenabgrenzung dieser Arbeit wurde weitgehend kein kritischer Vergleich mit kommerziellen Anwendungen gezogen. Aus demselben Grund wurde genausowenig die Verwendung beider freier Systeme auf der Grundlage einer umfassenden empirischen Studie untersucht. Immerhin zeigt Lagoudakis Erhebung (vgl. 2006: 21,27-28), dass 10% der Teilnehmer OmegaT kennen und 7% (Platz 8) es einsetzen.⁹³

Beides eröffnet weitere Forschungsmöglichkeiten, die spannende Ergebnisse liefern dürften.

⁹² Siehe die OpenTMS-Projektseite: <http://www.opentms.de/> (besucht am 21. 09. 2009).

⁹³ OmegaT wird sogar von 15% (Platz 2) der Studienteilnehmer, die nicht MS Windows als Betriebssystem haben, verwendet.

Anhang

Bibliographie

Gedruckte Quellen

- Arthern, Peter (1979): »Machine translation and computerized terminology systems: a translator's viewpoint«. In: *Translating and the computer. Proceedings of a Seminar, London, 14th November, 1978*. Hrsg. von Barbara M Snell. Amsterdam: North-Holland, S. 77–108.
- Barnet, Silvia u. a. (2005): *Der Brockhaus – Computer und Informationstechnologie*. Mannheim: Bibliographisches Institut und F. A. Brockhaus.
- Bowker, Lynne (2002): *Computer-Aided Translation Technology: A Practical Introduction*. 2. Aufl. University of Ottawa Press.
- Dudenredaktion, Hrsg. (2007): *Duden – Das große Fremwörterbuch*. 4. Aufl. Mannheim: Bibliographisches Institut und F. A. Brockhaus.
- Esselink, Bert (2003): »Localisation and translation«. In: *Computers and Translation - A translator's guide*. Hrsg. von Harold Somers. Amsterdam: Benjamins, S. 67–86.
- Höge, Monika (2002): »Towards a Framework for the Evaluation of Translators' Aids«. Diss. Helsinki. URL: http://ethesis.helsinki.fi/julkaisut/hum/yleis/vk/hoge/towar_dsa.pdf (besucht am 19.06.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZyU7Ur>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.
- Hutchins, John (1998): »The origins of the translator's workstation«. In: *Machine Translation* 13(4), S. 287–307. URL: <http://www.hutchinsweb.me.uk/MTJ-1998.pdf> (besucht am 13.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsa4YUCc>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.
- (2005): »Current commercial machine translation systems and computer-based translation tools: system types and their uses«. In: *International Journal of Translation* 17.1-2, S. 5–38. URL: <http://www.hutchinsweb.me.uk/IJT-2005.pdf> (besucht am 13.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsaC4VW9>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.
- Hutchins, John und Harold Somers (1992): *An Introduction to Machine Translation*. London: Academic Press.
- Hüttenegger, Georg (2006): *Open Source Knowledge Management*. Heidelberg: Springer.

-
- Kay, Martin (1980): »The proper place of men and machines in language translation«. In: *Machine Translation*, S. 3–23. Nachdruck (1997) vom Forschungsbericht CSL-80-11, Xerox PARC, Palo Alto, Kalifornien. Zitiert nach Hutchins (1998).
- King, Margaret (1998): »Evaluation Design: The EAGLES Framework«. In: *Evaluation of the Linguistic Performance of Machine Translation Systems. Proceedings of the Konvens '98 in Bonn*. Hrsg. von Rita Nübel und Uta Seewald-Heeg. St. Augustin: Gardez, S. 151–169.
- Krollmann, Friedrich (1971): »Linguistic data banks and the technical translator«. In: *Meta* 16, S. 117–124. Zitiert in Hutchins (1998).
- Lagoudaki, Elina (2006): *Translation memory systems: Enlightening users' perspective; key findings of the TM survey 2006 carried out during July and August 2006*. London. URL: <http://www3.imperial.ac.uk/portal/pls/portal-live/docs/1/7307707.PDF> (besucht am 16.06.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsaMDqiR>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.
- (2008): »Expanding the Possibilities of Translation Memory Systems. From the Translator's Wishlist to the Developer's Design«. Diss. Imperial College London. Die Seitenzahlen beziehen sich auf das von der Autorin dem Verfasser dieser Arbeit zur Verfügung gestellte PDF-Dokument.
- Lippmann, Erhard (1971): »An approach to computer-aided translation«. In: *IEEE Transactions on Engineering Writing and Speech* 14(1), S. 10–33. Zitiert nach Hutchins (1998).
- Massion, François (2005): *Translation Memory Systeme im Vergleich*. Reutlingen: doculine.
- Melby, Alan (1981): »A bilingual concordance system and its use in linguistic studies«. In: *The Eighth LACUS Forum 1981*. Hrsg. von W. Gutwinski und G. Jolly. Columbia: Hornbeam, S. 541–549. Zitiert nach Hutchins (1998).
- (1982): »Multi-level translation aids in a distributed system«. In: *Coling 82. Proceedings of the Ninth International Conference on Computational Linguistics, Prague, July 5-10, 1982*. Hrsg. von Jan Horecký. Amsterdam: North-Holland, S. 215–220. Zitiert in Hutchins (1998).
- (1992): »The translator workstation«. In: *Computers in translation: a practical appraisal*. Hrsg. von John Newton. London: Routledge, S. 147–165.
- Quah, Chiew Kin (2006): *Translation and Technology*. New York: Palgrave Macmillan.
- Schmitt, Peter A. (1999a): »Anleitungen / Benutzerhinweise«. In: *Handbuch Translation*. Hrsg. von Mary Snell-Hornby u. a. 2. Aufl. Tübingen: Stauffenburg, S. 209–213.
- (1999b): »Maßeinheiten«. In: *Handbuch Translation*. Hrsg. von Mary Snell-Hornby u. a. 2. Aufl. Tübingen: Stauffenburg, S. 298–300.
- Schmitz, Klaus-Dirk (2000): »Softwarelokalisierung – Eine Übersicht«. In: *Softwarelokalisierung*. Hrsg. von Klaus-Dirk Schmitz und Kirsten Wahle. Tübingen: Stauffenburg, S. 1–10.
- Schopp, Jürgen F. (1999): »Typographie und Layout«. In: *Handbuch Translation*. Hrsg. von Mary Snell-Hornby u. a. 2. Aufl. Tübingen: Stauffenburg, S. 199–204.

- Seewald-Heeg, Uta (2005): »Der Einsatz von Translation-Memory-Systemen am Übersetzer-arbeitsplatz«. In: *MDÜ (Mitteilungen für Dolmetscher und Übersetzer)* 6/2005, S. 25–30. URL: http://www.metatexis.org/reviews/TM-Vergleich_Version_300805.pdf (besucht am 31.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsc1KkGA>. Die Seitenzahlen beziehen sich auf die online verfügbare Version.
- Somers, Harold (2003): »Translation memory systems«. In: *Computers and Translation: a translator's guide*. Hrsg. von Harold Somers. Amsterdam: Benjamins, S. 31–47.
- Trujillo, Arturo (1999): *Translation Engines: Techniques for Machine Translation*. London: Springer.

Digitale Quellen

- Briel, Didier (2009): *RE: [OmT] This 100% match isn't a 100% match...* URL: <http://tech.groups.yahoo.com/group/OmegaT/message/14820> (besucht am 15.09.2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 14.07.2009. Archivierte URL verfügbar: <http://www.webcitation.org/5jsYY5BN7>.
- EAGLES Report*. URL: <http://www.issco.unige.ch/en/research/projects/ewg95/node1.html> (besucht am 25.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsX7TBvF>.
- Free Software Foundation: *Categories of Free and Non-Free Software - GNU Project*. URL: <http://www.gnu.org/philosophy/categories.html> (besucht am 24.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXlnCGe>.
- FSFE - Free Software Foundation Europe*. URL: <http://fsfe.org/index.de.html> (besucht am 23.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXexyHb>.
- Helary, Jean-Christophe (2007): *SourceForge.net: OmegaT - multiplatform CAT tool: Detail: 1756560 - Supported glossary file formats (TBX, CSV, TSV etc)*. URL: http://sourceforge.net/tracker/index.php?func=detail&aid=1756560&group_id=68187&atid=520350 (besucht am 03.09.2009). Feature-Request-Tracker von OmegaT auf SourceForge.net: Feature-Request vom 19.07.2007. Archivierte URL verfügbar: <http://www.webcitation.org/5jsZ0uTQt>.
- (2009): *Re: [OmT] Problems in importing glossary and TM from Trados into OmegaT (English-Chinese)*. URL: <http://tech.groups.yahoo.com/group/OmegaT/message/13024> (besucht am 09.09.2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 08.02.2009. Archivierte URL verfügbar: <http://www.webcitation.org/5jsY3K0Du>.
- Helary, Jean-Christophe u. a.: *OmegaT 2 User Manual*. Hilfedatei zum Programm.

-
- IDC: *Open Source Software Market Accelerated by Economy and Increased Acceptance From Enterprise Buyers*, IDC Finds. URL: <http://www.idc.com/getdoc.jsp?containerId=prUS21947509> (besucht am 24. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXKcCfX>.
- (LISA), Localization Industry Standards Association: *Term Base eXchange (TBX)*. URL: <http://www.lisa.org/Term-Base-eXchange.32.0.html> (besucht am 27. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZDKD7k>.
- *Translation Memory eXchange (TMX)*. URL: <http://www.lisa.org/Translation-Memory-e.34.0.html> (besucht am 27. 07. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXZU85X>.
- Medvedev, Sergei (2008): *Anaphraseus an OpenOffice.org Extension for Computer-Assisted Translation (CAT)*. Juli 2008. URL: http://downloads.sourceforge.net/project/anaphraseus/Presentation%20of%20Anaphraseus/20080713/Anaphraseus_Presentation.odp (besucht am 13. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsWsyLUW>.
- Okapi Framework - Introduction*. URL: <http://okapi.sourceforge.net/> (besucht am 23. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYg8HP0>.
- Okapi Framework - Rainbow*. URL: <http://okapi.opentag.com/help/applications/rainbow/index.html> (besucht am 13. 09. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYpcVjx>.
- Olifant Help - Welcome*. URL: <http://okapi.sourceforge.net/Release/Olifant/Help/index.html> (besucht am 13. 09. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsYx9ipC>.
- Open Source Initiative (OSI): *History of the OSI*. URL: <http://www.opensource.org/history> (besucht am 24. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZYQqtV>.
- Prior, Marc (2009): *TOXIC utility (readme.txt)*. URL: http://www.omegat.org/resources/toxic_scripts.zip (besucht am 24. 08. 2009).
- Raeiro, José (2008): *Re: [OmT] Compatibility with Trados*. URL: <http://tech.groups.yahoo.com/group/OmegaT/message/11039> (besucht am 31. 08. 2009). Mailing Liste OmegaT <OmegaT@yahoogroups.com>: Nachricht vom 02. 07. 2008. Archivierte URL verfügbar: <http://www.webcitation.org/5jsXwv0Nc>.
- Stallman, Richard M. (1996a): *Free Software Definition*. URL: <http://www.fsf.org/licensing/essays/free-sw.html> (besucht am 31. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsceS0Jk>.
- (1996b): *Was ist das Copyleft?* URL: <http://www.gnu.org/copyleft/copyleft.de.html> (besucht am 24. 08. 2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXjFs7v>.

- Stallman, Richard M. (1999a): *15 Jahre Freie Software*. URL: <http://www.gnu.org/philosophy/15-years-of-free-software.de.html> (besucht am 23.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZd8zje>.
- (1999b): *Das GNU Projekt*. URL: <http://www.gnu.org/gnu/thegnuproject.de.html> (besucht am 23.08.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsVybImp>.
- (2000): *Freie Software und freie Handbücher*. URL: <http://www.gnu.org/philosophy/free-doc.de.html> (besucht am 18.09.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsnmEft9>.
- Tagaligner: an aligner for parallel text*. URL: <http://tag-aligner.sourceforge.net/> (besucht am 03.09.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsZ4qkHV>.
- Welsh, Susan und Marc Prior (2009): *OmegaT for CAT Beginners*. URL: <http://www.omegat.org/en/tutorial/OmegaT%20for%20Beginners.pdf> (besucht am 07.04.2009). Archivierte URL verfügbar: <http://www.webcitation.org/5jsXTqPay>.

Glossar

Dateiformate

ASCII	American Standard Code for Information Interchange: 7-Bit-Zeichenkodierung mit 128 Zeichen.
CSV	Comma Separated Values
DOCX	Office Open XML Document (siehe OOXML)
HTML	HyperText Markup Language
IDML	Adobe InDesign Markup Language
MIF	Adobe FrameMaker Interchange Format
ODF	OpenDocument Format (OpenOffice.org u.a.)
ODG	OpenDocument Graphics (siehe ODF)
ODP	OpenDocument Presentation (siehe ODF)
ODS	OpenDocument Spreadsheet (siehe ODF)
ODT	OpenDocument Text (siehe ODF)
OOXML	Office Open XML (Microsoft)
PDF	Portable Document Format
PPTX	Office Open XML Presentation (siehe OOXML)
RTF	Rich Text Format
SRX	Segmentation Rules eXchange
TBX	TermBase eXchange
TMX	Translation Memory eXchange
TSV	Tab Separated Values
TTX	Zweisprachige Datei von Trados TagEditor
TXML	Zweisprachige Datei von Wordfast Professional
UTF-16	Ältere Unicode-Zeichenkodierung

UTF-8	8-bit Unicode Transformation Format: Zeichenkodierung mit über 1 Mio. Zeichen
XHTML	Extensible HyperText Markup Language
XLIFF	XML Localization Interchange File Format
XLSX	Office Open XML Workbook (siehe OOXML)
XML	Extensible Markup Language

Akronyme

ALPAC	<i>Automatic Language Processing Advisory Committee</i>
AS	Ausgangssprache
CAT	<i>Computer-Aided Translation</i> : Computerunterstützte Übersetzung
CMS	<i>Content Management System</i>
DTP	<i>Desktop Publishing</i>
EDV	Elektronische Datenverarbeitung
EG	Europäische Gemeinschaft
FAHQT	<i>Fully Automatic High Quality Translation</i> : Vollautomatisierte hochqualitative Übersetzung
FOSS	<i>Free and Open Source Software</i>
FSF	<i>Free Software Foundation</i>
GPL	<i>GNU General Public Licence</i>
GUI	<i>Graphical User Interface</i> : Grafische Benutzeroberfläche
HAMT	<i>Human-Aided Machine Translation</i> : Humangestützte maschinelle Übersetzung
HT	<i>Human Translation</i> : Humanübersetzung
IR	<i>Information Retrieval</i> : siehe dort
LGPL	<i>GNU Less General Public Licence</i>
LSP	<i>Language Service Provider</i>
MAHT	<i>Machine-Aided Human Translation</i> : Maschinengestützte Humanübersetzung
MIT	<i>Massachusetts Institute of Technology</i>
MT	<i>Machine Translation</i> : Maschinelle Übersetzung

OCR	<i>Optical Character Recognition</i> : Optische Texterkennung
OSI	<i>Open Source Initiative</i>
TM	<i>Translation Memory(ies)</i>
TMS	<i>Translation Memory System(e)</i>
TT	<i>Translation Technologies</i>
TU	<i>Translation Unit</i> : Übersetzungseinheit
TW	<i>Translator's Workstation</i> : Übersetzerarbeitsplatz (auch: <i>Translator's Workbench</i>)
WYSIWYG	<i>What You See Is What You Get</i> : siehe dort
ZS	Zielsprache

Fachausdrücke

Alignment Bezeichnet im Bereich der Translation Technologien das (automatische) Erstellen von Segmentpaaren aus einem Originaltext und dessen Übersetzung.

Alphaversion »Vorläufige Version eines Software-Produkts (z. B. eines Anwendungsprogramms), das weitgehend fertig gestellt ist und in diesem Stadium einigen Tests (Alphatest) unterzogen werden kann. Das nächste Entwicklungsstadium wird als Beta-version bezeichnet.« (*Brockhaus – Computer und IT*)

Batchverarbeitung (auch: Stapelverarbeitung) Bezeichnet in der Datenverarbeitung das »Abarbeiten mehrerer Programme in aufeinander folgenden Schritten [...] . Das Gegenteil der Stapelverarbeitung ist die Dialogverarbeitung.« (*Brockhaus – Computer und IT*)

Betaversion Spätes Stadium im Entwicklungsprozess einer Software oder Softwarekomponente, bei dem Funktionstests beinahe abgeschlossen und Programmfehler weitgehend behoben wurden.

Feature-Request Engl. für Leistungsmerkmalanforderung. Bezeichnet Anforderungen von Softwarebenutzern und/oder -entwicklern in Bezug auf Verbesserungen oder zusätzliche Funktionen der betroffenen Software. Diese Anfragen werden oft in einem Tracker (siehe dort) gesammelt.

Hacker »a) Computerbegeisterter, der versucht, Programmierprobleme durch Ausprobieren zu lösen; b) Computerbenutzer, der versucht, über Datenfernverbindungen widerrechtlich in fremde Datenbanken einzudringen.« (*Duden – Das große Fremdwörterbuch*)

Hardcopy Dokumentvorlage, die nicht maschinenlesbar ist (z.B. Drucksachen).

Hunspell Freies (GPL) Rechtschreibprüfungsprogramm, das ursprünglich für die ungarische Sprache entwickelt wurde und inzwischen zahlreiche Sprachen unterstützt. Es wird u.a. in OpenOffice.org, Mozilla Firefox (Webbrowser) und OmegaT eingesetzt.

Information Retrieval »Informationsrückgewinnung [...] , das Heraussuchen bestimmter Informationen aus einer umfangreichen Daten- oder Wissensbasis.« (*Brockhaus – Computer und IT*)

Java Programmiersprache.

Kernel Bezeichnet den »zentrale [n] Teil eines Betriebssystems, in dem alle »lebenswichtigen« Funktionen zusammengefasst sind.« (*Brockhaus – Computer und IT*)

Kommandozeile »Eine Zeichenkette, die in einer Maschinsprache geschrieben ist und zur Ausführung an den Befehls-Interpreter übergeben wird. In einer befehlsorientierten Benutzerschnittstelle ist die Kommandozeile der Ort auf dem Bildschirm oder Terminal, wo die über die Tastatur eingegebenen Befehle angezeigt werden. Auch grafisch orientierte Benutzerschnittstellen (etwa die Unix-Shell, aber auch Commander unter Windows) bieten die Möglichkeit, anstelle des Anklickens eines Symbols einen Befehl im Klartext einzugeben. Die einzelnen Befehle können auch in einer Stapeldatei abgelegt sein, sodass sie nacheinander abgearbeitet werden.« (*Brockhaus – Computer und IT*)

Laufzeitumgebung bezeichnet »Eine Menge von Programmbibliotheken und anderen Dateien, z. B. Initialisierungsdateien, die ein Programmmodul während seiner Ausführung braucht.« (*Brockhaus – Computer und IT*)

Linux (GNU/Linux) ist »ein Unix-ähnliches [...] Betriebssystem für Personal Computer und Workstations. Es wurde seit 1991 von dem Finnen Linus Torvalds aus Helsinki entwickelt und zeichnet sich durch eine große Verarbeitungsgeschwindigkeit und Stabilität aus. Die Besonderheiten von Linux liegen darin, dass es kostenlos ist bzw. nur zum Preis des Vertriebsmediums verkauft werden darf und dass der Quellcode von Anfang an frei zur Verfügung gestellt wurde.« (*Brockhaus – Computer und IT*)

locale Bezeichnet einen kulturellspezifischen Raum (meistens einen Staat) innerhalb eines gegebenen Sprachraums. So ist beispielsweise das österreichische Deutsch (ISO-Abkürzung: DE-AT) ein *locale* des deutschen Sprachraums. Aus diesem Wort entstand die sogenannte (Software-) Lokalisierung (siehe dort).

Lokalisierung »Den Prozess der Anpassung eines (Software-) Produktes an regionale Märkte, d. h. an unterschiedliche Sprach- und Kulturräume, bezeichnet man als (Software-) Lokalisierung. Ziel der Softwarelokalisierung ist es, technisch, sprachlich und kulturell angemessene Programmversionen für die jeweiligen Märkte zu erstellen [...] .« (Schmitz 2000: 3)

Match Bezeichnet im Bereich der Translationstechnologien einen Treffer in den TM-Ressourcen. Je nach Ähnlichkeitsgrad mit dem Originalsegment werden Matches mit einer Trefferquote (meist in Prozenten) versehen.

Mono »Projekt der Firma Ximian, das darauf abzielt, .NET auf Linux zu portieren und dabei zu einer Open Source Software zu machen.« (*Brockhaus – Computer und IT*) Ximian wurde 2003 von Novell gekauft.

.NET »Von Microsoft entwickeltes Konzept für eine Software-Infrastruktur (ein Framework), das verteilte Rechner und Anwendungen kommunizieren lässt sowie integriert.« (*Brockhaus – Computer und IT*)

Perl Programmiersprache.

Plug-in »Kleines Softwareprogramm, das in eine größere Anwendung integriert werden kann, um dessen Funktionalität zu erweitern [...] .« (*Duden – Das große Fremwörterbuch*)

Portieren Unter portieren versteht man »ein Betriebssystem, eine Anwendung oder eine andere Computerkomponente so [zu] modifizieren, dass es/sie in einer anderen Umgebung in gleicher Weise arbeitet.« (*Brockhaus – Computer und IT*)

Programmbibliothek bezeichnet »allgemein eine Sammlung von Programmen oder Dateien, bei der Programmierung eine Sammlung von Unterprogrammen, Routinen oder Prozeduren, die sich in einer Datei befindet. [...] Gewöhnlich werden immer alle Routinen eines Anwendungsbereichs in einer Bibliotheksdatei zusammengefasst, z. B. alle Routinen zur Ein- und Ausgabe von Daten in einer Ein-/Ausgabedatei. Weitere Beispiele sind Grafikbibliotheken, Laufzeitbibliotheken oder dynamische Bibliotheken (DLL). Es gibt fertige Bibliotheken bzw. Bibliotheksdateien für alle gängigen Programmiersprachen. [...] « (*Brockhaus – Computer und IT*)

Python Programmiersprache

Quickinfo »Kontextbezogene Hilfe«. (*Brockhaus – Computer und IT*)

Regulärer Ausdruck Zeichenkette, die nach bestimmten syntaktischen Regeln (einer Programmiersprache) geschrieben wird, um Informationen zu filtern, zu beschreiben oder zu suchen. Die Verwendung von Platzhaltern bei der Eingabe eines Suchbegriffes ist ein einfacher regulärer Ausdruck.

Segmentierung Bezeichnet im Bereich der Translationstechnologien die Zerteilung eines Textes in kleinere Einheiten (Segmente), meistens Sätze.

Skript Kleines, meist einfaches Programm mit wenigen Befehlen (beispielsweise zur Automatisierung von repetitiven Vorgängen).

Stable-Version Version einer Software, die (meistens) fehlerfrei ist und für die Verwendung in einer Produktionsumgebung geeignet ist.

StarBasic Programmiersprache.

Tcl/tk Programmiersprache.

Tag Bezeichnet ein »Auszeichnungsbefehl in Auszeichnungssprachen. In SGML oder HTML besteht ein solches Tag aus dem in spitze Klammern eingeschlossenen Befehlswort, das die Art der beschriebenen Formatierung und Strukturierung angibt (z. B. bestimmt das Tag die Schriftart oder das Tag <body> den Textkörper usw.)« (*Brockhaus – Computer und IT*)

Time-Sharing Bezeichnet »die Aufteilung von Rechenzeit auf mehrere gleichzeitig aktive Prozesse. [...] [Es] spielt hauptsächlich bei Mehrbenutzersystemen eine Rolle. Früher waren solche Mehrbenutzersysteme im wesentlichen Großrechner, heute sind es häufig Server. Leistungsfähige Mehrbenutzersysteme vermögen bis zu mehrere Hundert von Benutzern ausgelöste Prozesse zu bedienen, ohne dass es zu spürbaren Wartezeiten kommt.« (*Brockhaus – Computer und IT*)

Tracker Tracker sind Softwares, die im Bereich der Softwareentwicklung und -wartung eingesetzt werden, um Fälle oder Anfragen zu bearbeiten und zu verwalten. Sie verfügen oft über eine Online-Benutzeroberfläche und dienen dazu, die Kommunikation zwischen Entwicklern und/oder Benutzern zu erleichtern.

Undo »Funktion bei Anwendungsprogrammen und Betriebssystemen, mit der die letzte Aktion (bzw. die letzten Aktionen) rückgängig gemacht wird. Während früher oft nur die jeweils letzte Aktion (Texteingabe, Formatierung, Bewegen eines Zeichnungsobjekts usw.) aufgehoben werden konnte, sind heute bei Anwendungen Undo Buffer (dt. »Rückgängigpufferspeicher«) für 128 Befehle und mehr Stand der Technik.« (*Brockhaus – Computer und IT*)

Unix »Betriebssystem für Rechenanlagen, das heute in verschiedenen Varianten auf Geräten vom PC bis zum Supercomputer eingesetzt wird.« (*Brockhaus – Computer und IT*)
Entstanden in den 1970er Jahren.

What You See Is What You Get (Abgekürzt WYSIWYG) »Fähigkeit von Programmen, Dokumente auf dem Bildschirm genauso darzustellen, wie sie im Ausdrucke erscheinen. Dies ist heute Standard für Grafik-, Layout- und anspruchsvollere Textprogramme (DTP).« (*Brockhaus – Computer und IT*)

Liste der in OmegaT verwendbaren regulären Ausdrücke

The construct...	...matches the following:
Flags	
(?i)	Enables case-insensitive matching (by default, the pattern is case-sensitive).
Characters	
x	The character x , except the following...
\uhhhh	The character with hexadecimal value 0x hhhh
\t	The tab character (' \u0009')
\n	The newline (line feed) character (' \u000A')
\r	The carriage-return character (' \u000D')
\f	The form-feed character (' \u000C')
\a	The alert (bell) character (' \u0007')
\e	The escape character (' \u001B')
\c x	The control character corresponding to x
\0 n	The character with octal value 0 n ($0 \leq n \leq 7$)
\0 nn	The character with octal value 0 nn ($0 \leq n \leq 7$)
\0 mnn	The character with octal value 0 mnn ($0 \leq m \leq 3, 0 \leq n \leq 7$)
\x hh	The character with hexadecimal value 0x hh
Quotation	
\	Nothing, but quotes the following character. This is required if you would like to enter of the meta charactes !\$() * + . < > ? [\] ^ { } to match as themselves.
\\	For example, this is the backslash character
\Q	Nothing, but quotes all characters until \E
\E	Nothing, but ends quoting started by \Q
Classes for Unicode blocks and categories	
\p{InGreek}	A character in the Greek block (simple block)
\p{Lu}	An uppercase letter (simple category)
\p{Sc}	A currency symbol
\P{InGreek}	Any character except one in the Greek block (negation)

The construct...	...matches the following:
<code>[\p{L}&&[^\p{Lu}]]</code>	Any letter except an uppercase letter (subtraction)
Character classes	
<code>[abc]</code>	a , b , or c (simple class)
<code>[^abc]</code>	Any character except a , b , or c (negation)
<code>[a- zA- Z]</code>	a through z or A through Z , inclusive (range)
Predefined character classes	
<code>.</code>	Any character (except for line terminators)
<code>\d</code>	A digit: [0- 9]
<code>\D</code>	A non-digit: [^0- 9]
<code>\s</code>	A whitespace character: [\t\n\x0B\f\r]
<code>\S</code>	A non-whitespace character: [^\s]
<code>\w</code>	A word character: [a- zA- Z_0- 9]
<code>\W</code>	A non-word character: [^\w]
Boundary matchers	
<code>^</code>	The beginning of a line
<code>\$</code>	The end of a line
<code>\b</code>	A word boundary
<code>\B</code>	A non-word boundary
Greedy quantifiers	
These will match as much as they can. For example, <code>a+</code> will match <code>aaa</code> in <code>aaabbb</code>	
<code>X ?</code>	X , once or not at all
<code>X *</code>	X , zero or more times
<code>X +</code>	X , one or more times
Reluctant (non-greedy) quantifiers	
These will match as little as they can. For example, <code>a+?</code> will match the first <code>a</code> in <code>aaabbb</code>	
<code>X ??</code>	X , once or not at all
<code>X *?</code>	X , zero or more times
<code>X +?</code>	X , one or more times
Logical operators	
<code>XY</code>	X followed by Y
<code>X Y</code>	Either X or Y
<code>(XY)</code>	XY as a single group

Aus: Jean-Christophe Helary u. a.: *OmegaT 2 User Manual*. Hilfedatei zum Programm.

Vergleich der Funktionen von OmegaT und Anaphraseus

	OmegaT	Anaphraseus
Vor dem Übersetzen		
Projekt anlegen	ja	nein
Projekt ändern		
Sprachen	ja	–
Ausgangsdateien	ja	–
Sprachkodierung		
ISO 639-1	ja	ja
ISO 639-1 + ISO 3166	ja	ja
ISO 639-3	ja	ja
Dateikodierung		
ASCII	ja	ja
ANSI	ja	ja
ISO 8859	ja	ja
UTF-16	ja	ja
UTF-8	ja	ja
Satzsegmentierung	ja	ja
anpassbar	ja	nein
SRX	nein	nein
Ausgangsdateienformate ⁹⁴		
Reintext	ja	ja
RTF	ja	ja
OpenDocument	ja	ja ⁹⁵
OOXML	ja	ja
MS Office vor 2007	ja ⁹⁶	ja ⁹⁷

⁹⁴ Nicht angeführt sind hier software- und websitelokalisierungsspezifische Formate.

⁹⁵ Nur ODT.

⁹⁶ Nach Konvertierung.

⁹⁷ Nur DOC.

	OmegaT	Anaphraseus
PDF	nein	nein
TTX	ja ⁹⁶	nein
TXML	ja ⁹⁶	nein
DTP	ja ⁹⁶	nein
XLIFF	ja ⁹⁶	nein
TM-Dateienformate		
TMX	ja	ja ⁹⁶
Wordfast-TXT	nein	ja
Terminologie-Dateienformate		
TBX	nein	nein
TSV	ja	ja
Wörterbücher	ja	ja
Vorübersetzen	ja ⁹⁸	ja
Statistische Analyse		
Gesamtumfang	ja	ja
TM-Matches	teilw. ⁹⁹	nein
Wiederholungen	ja	nein
Bilder	nein	nein
Alignment	nein	nein
Übersetzen		
Editortyp	dediziert	in Writer
Absatzsegmentierung	ja	nein
Segmentgröße verändern	nein ¹⁰⁰	ja
Kopieren/Einfügen	ja	ja
Einfügen von Sonderzeichen	nein ¹⁰¹	nein ¹⁰²
Übernahme der Formatierung	ja	nein
Undo	ja	ja
Zieltextvorschau	nein ¹⁰³	–

98 Über Batchmodus.

99 Und erst nach einem Vorübersetzen-Durchlauf.

100 Nur über die Bearbeitung der Ausgangsdatei oder Anpassung der Segmentierungsregeln.

101 Nur über die systemweite Zeichentabelle.

102 Nur über die systemweite Zeichentabelle oder die von Writer.

103 Nur nach Erstellung der Zieldateien.

	OmegaT	Anaphraseus
Anzeige des Übersetzungsfortschritts	ja	nein
Konkordanzsuche	ja	nein
Automatische Propagation	teilw.	nein
Ersetzen von Einheiten	nein	nein
Rechtschreibprüfung	ja	ja
MT-Modul	ja	nein
Nach dem Übersetzen		
Rechtschreibprüfung	nein ¹⁰³	ja ¹⁰⁴
Terminologieprüfung	nein	nein
Tagprüfung	ja	nein
Vollständigkeitsprüfung	teilw.	nein
Pflege des Datenbestands	nein	nein

104 Funktion von Writer.

Verzeichnis der erwähnten freien Anwendungen

Stand: 5. September 2009.

Name	Typ	Projektseite
Anaphraseus	TM-System	http://anaphraseus.sourceforge.net/
Bitext2tmx	Alignment-Programm	http://sourceforge.net/projects/bitext2tmx/
bligner	Alignment-Skript	http://www.omegat.org/en/resources.html
GOOCR	Texterkennungsprogramm	http://jocr.sourceforge.net/
mso2ooo	Konvertierungsskript	http://leapon.net/en/mso2ooo-batch-convert- microsoft-office-documents-openoffice-documents
Okapi Framework	Konvertierungsprogramme	http://okapi.opentag.com/ (Mono+Java) http://okapi.sourceforge.net/ (.NET)
OmegaT	TM-System	http://www.omegat.org/en/omegat.html
Pdftotext	Konvertierungsskript	http://www.foolabs.com/xpdf/
TagAligner	Alignment-Skript	http://tag-aligner.sourceforge.net/
Toxic	Konvertierungsprogramm	http://www.omegat.org/en/resources.html

Lebenslauf

Name: Sébastien Guillardau
Geburtsdatum: 23. Dezember 1977
Staatsbürgerschaft: Frankreich
E-Mail: s.guillardau@gmx.at

Ausbildung

Sep. 2004 – Feb. 2005: Erasmus-Austauschsemester in Las Palmas de Gran Canaria (E)
Seit Oktober 2002: Übersetzer- und Dolmetscherstudium an der Universität Wien
Okt. 2001 – Jul. 2002: Übersetzer- und Dolmetscherstudium an der Fachhochschule Köln (D)
Okt. 1999 – Jun. 2001: Germanistik- und Romanistikstudium (Lehramt) an der Universität Bonn (D, ohne Abschluss)
Okt. 1996 – Jun. 2000: *Maitrise »LLCE allemand«*: Germanistikstudium an der Universität Caen (F), Magisterabschluss
Sep. 1995 – Mai 1996: *»Classes préparatoires aux Hautes Études Commerciales«*: Vorbereitungslehrgang für BWL-Hochschulen in Caen
Juni 1995: Matura
Sep. 1988 – Jun. 1995: Besuch des Gymnasiums in l'Aigle (F); Schwerpunkt: Physik und Mathematik

Berufliche Erfahrung

Seit Mai 2003: Übersetzer und Dolmetscher (freier Mitarbeiter) bei der VAMED Engineering GmbH, Wien
Seit 2004: Freiberuflicher Übersetzer für zahlreiche Agenturen im In- und Ausland
Apr. 2005 – Jul. 2008: Übersetzer und Projektassistent (freier Mitarbeiter) bei der M27 WEDO Unternehmensberatungsgesellschaft mbH, Wien
Aug. 2003 – Sep. 2003: Übersetzer und Projektassistent für SAP R/3 bei der CNT Management Consulting GmbH, Wien

Sprachkenntnisse

Französisch: Muttersprache
Deutsch: Sehr gut in Wort und Schrift
Spanisch: Sehr gut in Wort und Schrift
Englisch: Gut in Wort und Schrift

Abstract

DEUTSCH In den vergangenen zwanzig Jahren hat das Interesse an Translation Memory Systemen sowohl vonseiten der professionellen Übersetzer als auch der translationswissenschaftlichen Forscher stetig zugenommen. Trotzdem wurden bis dato keine Studien durchgeführt, die sich ausschließlich einem besonderen Typus von TM-System widmen: der freien Software. Nach einer kurzen historischen Darstellung und einer Definition sowohl von TM-Systemen als auch von freier Software zieht diese Arbeit einen Vergleich zwischen den zwei freien TM-Systemen OmegaT und Anaphraseus nach funktionalen und nichtfunktionalen Kriterien. Aus diesem Vergleich geht hervor, dass beide Programme, die von einer Gemeinschaft von freien Programmierern und Übersetzern entwickelt werden, in einen professionellen Kontext einsetzbar sind. Als freie Anwendungen tragen sie außerdem zur Verbreitung einer Technologie bei, von der weltweit all diejenigen, die im Übersetzungsbereich – und zwar ungeachtet ihres aktuellen Wohnsitzes – tätig sind, profitieren können.

ENGLISH In the past twenty years the interest in translation memory systems has grown among practitioners and researchers of translation studies alike. Nevertheless, until now, no researches have been carried out exclusively on a particular type of TM systems: free softwares. After giving a brief history and definition of both TM systems and free softwares, this paper compares the two free TM systems OmegaT and Anaphraseus with regard to functional and non-functional aspects. The comparison results show that both softwares, developed by communities of free programmers and translators, are suitable for use in a professional environment. As free softwares they contribute furthermore to spread out a technology from which all those dealing with translation can benefit, regardless of where they live.

This work has been written in \LaTeX using the free fonts LINUX LIBERTINE and Linux Biolinum.

Copyright © 2009 Sébastien Guillard

Permission is granted to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found at:
<http://www.gnu.org/copyleft/fdl.html>.