Open Source-MÜ-Systeme

Referat im Seminar «Maschinelle Übersetzung» (HS 2011)

Simon Hafner & Hernani Marques
01. Dezember 2011

Zusammenfassung

Wir beschäftigen uns in unserem Referat mit maschinellen Übersetzungssystemen (MÜ-Systemen), deren Quellcode als Open Source Software (OSS) frei verfügbar ist. Wir möchten thematisieren, weshalb OSS-MÜ-Systeme Vorteile gegenüber kommerziellen oder MÜ-Systemen ohne frei verfügbaren Quellcode (Closed Source Software) aufweisen. Den (praktischen) Fokus legen wir auf zwei Systeme: Apertium und Moses. Insbesondere im Bereich von Minderheitensprachen (z. B. Klingonisch) versprechen OSS-MÜ-Systeme dazu beizutragen diese im Web sichtbarer zu machen und somit das Verständnis über ihre Funktionsweise zu erhöhen und dazu beizutragen, dass diese Sprachen (in ihrer Nutzung) an Bedeutung gewinnen.

1 Einleitung

Wir legen zunächst die Problematik (Kap. 2) dar, die wir mit Closed Source-MÜ-Systemen sehen. In einem weiteren Schritt präsentieren wir MÜ-Systeme, die quelloffen sind, wobei wir uns auf nur zwei Systeme beschränken (Kap. 4), von denen eines regelbasiert, das andere (primär) statistisch-basiert ist. Wir führen diese Systeme praktisch vor. Gegenstand von Kap. 3 sind die Herausforderungen und (unsere) möglichen Lösungsansätze. Unsere Lektüreempfehlungen finden sich am Schluss (Kap. 5) dieses Handouts.

2 Problemstellung

Closed Source Software (CSS) ist Software, deren Quellcode unter Verschluss gehalten wird. Viele bekanntere MÜ-Systeme, seien sie webbasiert, wie Google Translate[1], oder für den stationären persönlichen oder unternehmensweiten Einsatz gedacht, wie MÜ-Software von Linguatec[2], sind im Quellcode nicht verfügbar.

Gerade bei kommerziellen Systemen werden in der Regel nur verbreitete Sprachpaare unterstützt. Das hat wirtschaftliche Gründe (Rentabilität). Hat man zum Ziel ein Übersetzungssystem für eine Minderheitensprache zu entwickeln, wie

1. `http://translate.google.com/` (letzter Zugriff: 2011-11-24)
2. `http://www.linguatec.de/products/` (letzter Zugriff: 2011-11-24)

beispielsweise Klingonisch, so steht man bei CSS-MÜ-Systemen üblicherweise vor verschlossenen Türen.

Das Problem der Transparenz stellt sich aber nicht nur hinsichtlich der Software an und für sich. Eine weitere Dimension stellen die linguistischen Ressourcen dar, wie bei regelbasierten Ansätzen die Transferregeln, das verwendete Lexikon und die dazugehörige Annotation. Es ist technisch möglich diese Ressourcen einem CSS-System zu entlocken, allerdings mögen dies die Lizenzbedingungen untersagen. Bei statistisch-basierten Systemen ist in aller Regel nicht transparent einsehbar, welches Korpus sich hinter dem System verbirgt. Dennoch: Sind bei einem CSS-System wenigstens die Ressourcen im Sinne von Open Source (frei) nutzbar oder ersetzbar, so ist der Spielraum bereits grösser (vgl. [Forcada2006] ab Kapitel 2.1).

3 Herausforderungen & Lösungsansätze

Eine Minderheitensprache zu implementieren, ist bei CSS schwierig, da zu jeder Sprache wenigstens ein paar Regeln gehören. Im Falle von Apertium kann man ein Kuhdorf-Dialekt von Katalanisch mit relativ wenig Aufwand hinzufügen. Dies würde vielleicht sogar mit einer etwas angepassten Wortliste erreicht, da die Syntax vermutlich sehr ähnlich wäre. Bei Sprachen, die stärker abweichen, wie z. B. Klingonisch[3], das stark agglutinierend ist, benötigen wir jedoch mindestens eine einfache Syntaxerkennung, um halbwegs brauchbare Ergebnisse zu erzielen. Diese in eine OSS zu implementieren, sollte sich wesentlich einfacher gestalten als bei einer CSS.

Ein Problem besteht für Minderheitensprachen generell: Es liegt wenig Sprachmaterial vor, weil die Sprache in der (heutigen) Hauptressource des Webs wenig verbreitet ist. Gleichzeitig kann der OSS-Ansatz mit seiner offenen Austauschkultur dazu beitragen, dass dieses Wissen gemehrt wird, wenn die Entwicklung öffentlich erfolgt. Sind schliesslich MÜ-Systeme verfügbar, welche von Minderheitensprachen in weiter verbreite Sprachen, wie Deutsch oder Englisch, übersetzen können, so kann die Sichtbarkeit der Minderheitensprachen und damit auch die Verfügbarkeit linguistischen Sprachmaterials erhöht werden; denn: es wird Autoren möglich in der Minderheitensprache Texte zu verfassen, im Wissen darüber, dass eine Übersetzung für grössere Leserschaften (anderer Sprachen) möglich ist (vgl. [Forcada2006] ab Kapitel 5).

4 Zwei OSS-MÜ-Systeme: Apertium und Moses

Im Folgenden stellen wir zwei OSS-MÜ-Systeme vor: Apertium und Moses.

4.1 Apertium

4.1.1 Abriss

Apertium[4] ist ein RBMT-System, das 28 Sprachpaare (teilweise in bidirektionaler Übersetzungsrichtung) beherrscht. Es fällt auf, dass darunter Minderhei-

3. `http://www.coli.uni-saarland.de/fs-coli/procs/clauss.html` (letzter Zugriff: 2011-11-24)
4. `http://www.apertium.org/` (letzter Zugriff: 2011-11-24)

tensprachen zu finden sind, die (wirtschaftlich) wenig rentabel scheinen - u. a.: Katalanisch, Esperanto, Baskisch. Finanziell unterstützt wird das Projekt von der spanischen Zentralregierung. Die Software mit allen zugehörigen linguistischen Ressourcen untersteht der General Public License[5] (GPL), einer Lizenz, die von der Free Software Foundation stammt. Diese erlaubt es die Software in veränderter Form zu vertreiben - unter der wichtigen Bedingung, dass auch Einsicht in den Quellcode möglich ist. Einige linguistischen Daten sind mit Creative Commons-Lizenzen[6] belegt, welche zumindest eine Nutzung für nicht-kommerzielle Zwecke erlauben.

4.1.2 Technologien

Im Folgenden erfolgt eine Auswahl von drei Technologien, die Apertium einsetzt:

Shallow-Transfer Im einfachsten Fall sucht Apertium im Quellsprachtext nach Sequenzmustern von Wörtern, die in einem Katalog festgehalten sind; die Übersetzung erfolgt direkt (in einem Schritt) anhand eines Bilexikons in die Zielsprache. Dieses Verfahren, bekannt als "Shallow-Transfer", funktioniert nur bei syntaxähnlichen Sprachen (vgl. [Forcada2010] S.73ff.)

Advanced Transfer Seit das Sprachpaar Englisch-Katalanisch unterstützt wird, erfolgt die Übersetzung in einem (fortgeschrittenen) 3-Schritte-Prozess (vgl. ebd. S.77ff.):

1. Im *Chunking* werden Satzkonstituenten der Quellsprache erfasst, die Wort-für-Wort-Übersetzung betrieben und ferner morphosyntakische Anpassungen für die Zielsprache markiert.
2. Es erfolgt im *Interchunking* eine Neuordnung der Satzkonstituenten gemäss der Zielsprache.
3. Abschliessend werden mittels *Postchunking* Anpassungen der Wortformen (gemäss den Markierungen des 2. Schritts) vorgenommen und der Text in die Zielsprache geschrieben (validiert vom Sprachgenerator).

XML Die linguistischen Daten werden im XML-Format gehalten (vgl. ebd. S.114ff.), aus Gründen der Interoperabilität. So sind diese auch für andere MÜ-Systeme (theoretisch) verwendbar. Bei CSS-Systemen wird im Gegenzug oftmals (bewusst) auf proprietäre, nicht-standardisierte Formate gesetzt.

4.2 Moses

4.2.1 Abriss

Moses[7] ist primär ein SBMT-System. Da es unter der LGPL[8] lizenziert ist, kann es als (kostenfreie) Basis für kommerzielle Produkte verwendet werden. Es ist z. B. denkbar eigene Korpora einzubinden und ein GUI hinzuzufügen, um in der Folge das Ganze als benutzerfreundliches Paket zu verkaufen. Der interessante Teil ist zudem, dass es die Möglichkeit bietet, eine Minderheitensprache zu implementieren. Es wird bereits eine betrachtliche Anzahl von Sprachpaaren unterstützt, die eine umfassende Anwendung ermöglichen.

5. `http://www.gnu.org/copyleft/gpl.html` (letzter Zugriff: 2011-11-24)
6. `http://creativecommons.org/licenses/` (letzter Zugriff: 2011-11-04)
7. `http://www.statmt.org/moses/` (letzter Zugriff: 2011-11-24)
8. `http://www.gnu.org/licenses/lgpl.html` (letzter Zugriff: 2011-11-24)

4.2.2 Technologien

Beam search Moses verwendet den "Beam search"-Algorithmus, um im Rahmen seiner SBMT-Ansätzen effizient zu einem Ergebnis zu kommen (vgl. [Norvig1992]). Dieser Algorithmus ist eine Mischung aus den Algorithmen "A*" und "Breadth-first search" mit reduziertem Speicherverbrauch - im Vergleich zu "Best-first".

Phrase-based Viele Wordfolgen werden direkt aus einem Korpus übernommen, wenn es ein Beispiel gibt; das sind Ansätze aus der EBMT (Example-Based Machine Translation).

Factored Moses unterstützt mehrdimensionale Daten für einzelne Tokens, was es möglich macht, manuell Regeln zu implementieren, um dem System bei stark flektierenden bzw. agglutinierenden Sprachen auf die Sprünge zu helfen. Dies ist bei z.B. Quechua / Kingonisch extrem hilfreich.

5 Lektüreempfehlungen

Als wichtigste Lektüre empfehlen wir den interessanten Text "Open-source machine translation: an opportunity for minor languages", welcher als Leitartikel für unseren Beitrag gelten kann. Der Artikel ist im PDF verfügbar (vgl. [Forcada2006]). Wir empfehlen ferner die Webseiten von Apertium und Moses zu durchstöbern, um Einblick in die Welt von OSS-MÜ-Systemen zu erhalten:

- `http://www.apertium.org/` (Mit Web-Übersetzer)
- `http://www.statmt.org/moses`

Wir wünschen viel Spass!

Literatur

[Norvig1992] Norvig, Peter: *Paradigms of artificial intelligence programming: case studies in common LISP*. S.195-196. San Fransisco: Morgan.

[Forcada2010] Forcada Mikel L. et al.: *Documentation of the Open-Source Shallow-Transfer Machine Translation Platform Apertium*.
URL: `http://xixona.dlsi.ua.es/~fran/apertium2-documentation.pdf` (letzter Zugriff: 2011-11-24)

[Forcada2006] Forcada, Mikel L.: *Open-source machine translation: an opportunity for minor languages*. In: Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages).
URL: `http://www.dlsi.ua.es/~mlf/docum/forcada06p2.pdf` (letzter Zugriff: 2011-11-24)