[*Translating and the Computer 22*: Proceedings of the Twenty-second international conference…16-17 November 2000. (London: Aslib, 2000)]

TQPro:
Quality Tools for the Translation Process

Gr. Thurmair
Sail Labs

Introduction

This paper describes the developments of tools for machine translation as will take place in the TQPro project. These tools support the translation workflow by providing support for the different steps of it: Preparation of the translation, translation proper (with a focus on machine translation), and revision. The tools to be developed communicate with each other using standard interfaces for lexicon exchange, memories, and text handling.

1 The TQPro project

This project is one of the research projects funded by the European Commission in the context of the Fifth Framework Program, in the section of Human Language Technology. Partners are Sail Labs (coordinator), SAP, Lotus Development Ireland, Logoscript (Barcelona), CAT (Milan), and CST Copenhagen. It started in April 2000 and has a duration of 30 months.

Baseline of the project is the current situation of professional translation, which is characterised by increasing demands (integration into localisation workflow) and increasing time constraints (simultaneous shipment). This situation requires special attention to quality aspects of the workflow.

The main objective of TQPro is therefore to improve the quality of professional translation, by developing software support for each phase of the workflow. TQPro tools support different types of workflow, and provide support for preparation, translation and revision phases, both for conventional and machine translation. Tools to be developed are: verification of terminology, extraction of lexicon entries from corpora for fast resource build-up, pattern matchers and intelligent search and replace, and others. But also, TQPro will increase the quality of the translation tools themselves (translation memories, machine translation). Partners expect a significant increase in productivity without quality loss.

In more detail, TQPro has the following objectives:

- Provide quality control tools. TQPro tools will help to verify the quality of input and output texts and terminology used.
- Improve the quality of machine translation systems. Making MT a quality tool for professionals requires a significant quality increase. TQPro will provide better linguistic capabilities and customising tools for vertical application MT.
- Support quality in the workflow, by providing models for better workflow integration, and by productivity tools for all phases.
- Speed up the production of resources for human and machine translation: Verify the quality of memories, by indicating mis-aligned segments; and maintain terminology in multi-language workflows, including human and machine translation terminology.
- Develop standards for text and terminology interchange. This is to support multi-vendor environments within professional translation agencies.

2 The TQPro system architecture

TQPro is intended to be a toolbox of functions which do certain tasks in a translation environment. As a result, each component should be independent, have its own user interface and linguistic resources, and will be usable in a stand-alone mode.

In addition, it should also be demonstrated that the components can be integrated into a complete translation environment. TQPro uses the Lotus DTO (Domino Translation) Environment for this purpose; it provides standard functionality like task management, file upload and download, status

querying, and other useful functions. It allows for distributed services and includes a scheduler which calls these services and monitors them. There will also be a common TQPro Graphical User Interface, for demonstration purposes.

The overall architecture is described in fig. 1:

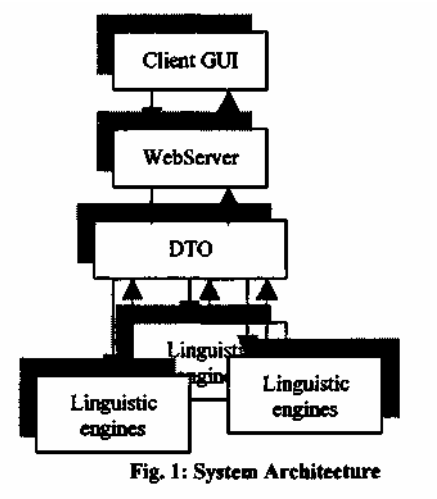

Fig. 1: System Architecture

The client will be browser-based, or implemented in Java directly. DTO, which will be provided by Lotus as a background software, provides a front-end interface to the web server which will be used by the TQPro tools. Most of the TQPro components will be seen as linguistic engines; they will be interfaced with the DTO backend API and linked to the overall system.

3 Standards

The linguistic engines just mentioned are independent components; they can be grouped in a modular way to support the different workflow requirements. In order to be usable in a broader environment, they use standards for communication and data exchange. TQPro is active in the standardisation in the following areas:

- **Text Handling**. The text is a common point of reference for all translation tool. A common way of text representation is a basic requirement for multi-vendor translation systems
- **Translation Memory** exchange. The representation of bilingual text is the basis for a series of tools in the context of quality checks. TQPro has decided to use the TMX standard here, developed by the LISA OSCAR group, (http://www.lisa.org/tmx/).
- **Lexicon Exchange**. Lexical resources are of utmost importance for any kind of translation task, so exchange of lexicon information needs to be easily possible.
- **MT-API**. In large translation environments like iTranslator (www.itranslatoronline.com) or Alis (www.alis.com), there are always MT engines of different vendors involved, in order to have a significant coverage in language pairs. This fact requires common and standardised APIs to link new MT engines into such environments.

TQPro is active in three of these fields: OText for text Handling, OLIF for lexicon exchange, and MT-API for linking MT-engines. It continues activities which have been started by the OTELO project (www.otelo.lu).

3.1 OText: Text Handling

This standard defines a lean format for text handling. It follows the *shadow file* approach, i.e. it splits the text parts from the formatting parts of a document (which are collected in a shadow file), and merges the two files after translation. The text part is an XML file, and it contains a small set of markups.

- **Structural markups** are paragraphs, comments, and protected segments (which should not be touched by tools).
- **Inline markups** are formats, references (like footnote references), and special characters. In the formatting policy, OText follows the OpenTag approach rather than the TMX proposals.

- **Delimiting markups** are text units (created by some sentence segmenting tool), literals (like names, abbreviations, constants), and phrases (like index terms, or special terms in general, e.g. marked up by some term quality checking tool).
- There are also **variants** for text units and phrases. Variants could be either translations, or corrections as produced by quality checking tools).

All markups have attributes, like the **types** of paragraphs, text units, literals; the **language** of these segments, a quality **score,** and others.

OText is a lean format, covering only the most important parts of text formatting. It has been designed to be a subset of more elaborate standards like OpenTag (www.opentag.com) or EPTAS (as used in the Euramis project).

The current version of OText is V3.2, and it is available on the TQPro web side (www.tqpro.de).

3.2 OLIF: Lexicon Exchange

Lexicon Exchange is a cornerstone in a multilingual translation environment. The development of OLIF (Open Lexicon Interchange Format) has started in the OTELO project, resulting in a first version of a standard. Since then, a second version has been developed, and a special consortium for OLIF has been founded, hosted by SAP (www.olif.net).

OLIF V2 is XML-based. An OLIF file consists of a header and a body which contains the entries. Each entry consists of groups, groups are: monolingual description (lexicographic and terminological), multilingual description (i.e. transfers), cross-reference description (like links to ontologies), and possibly a user-defined section for specific needs. Each of these groups consists of lists of atomic features and values.

OLIF is concept-based; the basic unit is the lexicon entry, which is a (language-specific) concept. It is oriented towards lexical description rather than terminological description, and many linguistic features can be represented. Links to other projects (like SALT or EAGLES) have been established to ensure compatibility of efforts in this domain.

A detailed overview of the OLIF exchange format is given by S.McCormick [McC2000]. The V2 proposal will be published on the OLIF website (www.olif.net).

3.3 *MT-API*

Activities on standardising APIs for MT engines have their history in the meantime; they started at the MT-Summit in San Diego. They have become more relevant in the context of large internet MT environments where usually several engines of different vendors must be integrated into a common system, to provide the language coverage which such sites usually require.

Taking up such activities, a proposal for minimal functionality will be developed within TQPro, and published on the TQPro web side. This activity is driven by Lotus.

4 TQPro Tools

TQPro uses these standards to provide the translation quality tools to be developed. All texts use OText format, all memories use TMX, all lexical resources read and write OLIF. This way it is guaranteed that a multi-vendor use is possible as soon as the different vendors provide links into these standards.

4.1 Lexicon Extraction

Significant effort is given to the possibility to create lexical resources, esp. for MT systems, from corpora in a semi-automatic way. Studies in corpus linguistics have shown that much useful information can be extracted from corpora, not just terms but also their attributes, and even their translations (if bilingual corpora are available). TQPro wants to exploit this, and provide a chain of tools to do that:

- Term Extraction is the first step; it creates a list of term candidates for a given corpus. Term candidates (single and multiword terms) will have basic annotations, like part-of-speech, term structure, and the like. Such approaches have been reviewed by [HV1999].
- Terms need to be enriched by linguistic information in order to be useful for MT. This can be performed by defaulting mechanisms (e.g. for the inflection type), or by corpus analysis, to find gender, syntactic subcategorisation, and even semantic relations.

- The enriched term list will be compared with existing memories (in TMX format), to find the translation equivalents. This way, bilingual entries can be created.
- The candidates are then compared to the lexicons of the backend systems (can be both MT systems or term databases), and terms which are already known are eliminated.
- A special OLIF editor is provided to review the remaining candidates, and correct mistakes if necessary.
- The final file is converted into the import format of the respective backend system, and imported from there.

The whole workflow operates on OLIF files: Components read and write OLIF files, and basically extend the set of annotations until full-fledged MT entry candidates are created. Quality issues are: improvement of term extraction (the first step), and bilingual term extraction from memories in case of l:n equivalents.

4.2 Term Quality Tools

Monolingual Term Verification

While the Lexicon Extraction Tool is intended to support a situation where a new list of terms needs to be created (e.g. which a new domain must be covered), the Term Verification is a comparison of a text with a legal set of terminology; so it supports control of terminology.

The component checks terms whether they are legal or illegal in a given subject area or not. Legal and illegal terms are stored in a database, and used in OLIF form in this component. General vocabulary terms can be stopped. It can also identify unknown terms (which tend to be illegal in a terminological environment which is set up properly).

Result of this component is a report which presents the legal / illegal / unknown terms, and additional information (context, term to be used instead, etc.).

This component can run on the source text, for translation preparation to provide some terminological control. It can also run on the translated text, to verify translations used by the translation agencies involved. Unknown terms in target text need to be reviewed by the posteditors.

Bilingual Term Verification

Bilingual term verification investigates the equivalents used, i.e. it checks if a given translation is "legal". This is done on the basis of bilingual text, i.e. memories (available in TMX format). Source and target terms are identified, and then their relation is investigated: Is the current translation correct, given parameters like subject area selection, transfer tests, and others? Possibly incorrect translations are flagged for the revision process.

Challenges for this component are source-target term identification, and 1:n translations.

4.3 Memory Quality

One task in TQPro is to support memory quality. Special attention is given to the identification of possible misalignments. They will be tried to be identified using a series of heuristics beyond the length of the segments: bilingual term verification, occurrence of certain literals in the texts, non-translated segments, and the like.

The component flags candidates for review. It would run as a verification component for newly built memories.

4.4 Text Quality

Some components deal with the quality of the text itself, both in its source and target form.

Pattern Matching

TQPro will provide an intelligent pattern matcher which is able to identify certain types of expressions in the text, and mark them up (as OText literals or phrases). Its core will be a standard regular expression analyser, it is extended by additional features like list lookup and some linguistic

improvements. A special user Interface will be designed to ease the definition of the regular expressions.

Applications can be: recognition of abbreviations and constants in texts; recognition of names (person, place, company names etc.), recognition of dates, currency units and measurement expressions; protection of certain text elements from being translated; text correction features (punctuation, spacing), and others.

The tool will receive an OText file and produce another OText file, applying the pattern rules specified by the users. It is a text correction tool, and it also can improve translation quality by flagging the literals.

Completeness

This tool checks if a translation is complete. Completeness is defined on a content level, and on a formatting level.

- On a content level it is checked if all segments are translated, and if the proper translation of terminology has been used, if target texts are significantly shorter / longer than their source language counterparts.
- On a formatting level it needs to be checked if all figures and tables are there, if the sequence of headings of the target matches the source text, if all inline tags are transferred, and the like. Unlike the other components, formatting checks cannot be done on an OText level but need to be done for the native editing system.

4.5 Machine Translation Quality

There is a series of components dealing with MT quality. TQPro wants to cover the full workflow: translatability check, MT quality proper, diagnosis.

Translatability

This component defines if a given text can be translated by MT or not. To answer this question before any further processing is relevant in large internet translation services, to define the trade-off between quality and cost; it is also relevant in conventional translation services: Setting up all the texts and the lexica for MT, to find out later that the text is still not suitable for MT translation is a waste of resources.

Unlike previous projects, the translatability checker will use basic linguistic information (esp. part-of-speech taggers) to define patterns of easy translation or difficult ambiguities. It will not in itself check the lexicon coverage, as it is intended to be MT system (and therefore MT lexicon) independent.

The tool will read OText files, and create diagnostic information, both on sentence and on text level. The OText file can be a complete text or a delta file containing only segments which are not in the memory.

MT Quality

TQPro is also looking into the issue of improving the quality of the MT engines themselves. Three engines will be investigated: Comprendium (Sail Labs), Logos, and PaTrans (CST). Research directions will go into adding statistical information in both the analysis and generation phase, and adding semantic information (both lexical and rule semantics) in both analysis and transfer phase.

The different partners will focus on different aspects of this program, and do research on these quality issues.

The result of mis task will not be improved ready-to-buy solutions; this would go far beyond the possibilities of a project like TQPro. Rather, it will explore certain research aspects and new directions, and evaluate, on a corpus basis, if such approaches can improve the large-scale systems, and by how much. Up to now, translation quality is mainly tied to morphosyntactic analysis; and it is time to go a step beyond, not just in small experimental systems but in an application using significant corpus material.

Diagnostics

A third tool for MT support looks into diagnostics of MT output Given a limited time for post-editing, users should be presented the segments which need postediting effort most urgently, to spend postediting time efficiently.

So the diagnostics tool provides a kind of ranking to present the "very bad" segments first, then the "bad" ones, etc. Depending on the amount of time available, posteditors can work down this list.

The component takes an OText file, and adds scores to the translated segments which can be used for ranking.

To be independent of the MT system cores, the technology which will be applied is statistic modelling. Probabilities can be assigned to certain combinations of words / word sequences / category patterns. If a segment deviates from "standard" behaviour, it is a candidate for postediting.

6 System Aspects

6.1 Integration

At present (November 2000) the project has completed its specification phase, and implementation is starting. Implementation will use the DTO backbone as provided by Lotus, and link the different components into this environment. A special client GUI will be designed along with the project progress, to provide uniform access to all tools.

Experience shows, however, that a very modular implementation is required to support the different configurations of the TQPro tools for the different workflows. Therefore all tools are also implemented in a stand-alone way, to be tested independently of the TQPro system configuration.

6.2 Test and Evaluation

The different components of TQPro need to use different testing methods. Testing will be done in a corpus-based way; corpora (mainly memory content) have been provided in significant size by the project members.

- **Lexicon Extraction:** Test criteria will be recall and precision, on different levels: Term extraction will ask how many term candidates have been correctly found, using reference terminology for a given corpus. Lexicon annotation will be tested for the percentage of correct annotations. Bilingual term extraction will be tested for correctness of these candidates.
- **Term Verification:** Test criteria will again be recall and precision. Recall would test if all term candidates have been identified, and precision would check the correct assignment of the different output classes.
- **Translation Memory** will test if the "wrong" segments of a memory have been correctly identified, and how much noise needs to be taken into account to find the incorrect ones. Also, the contribution of the different heuristic strategies to the overall results will be examined.
- **MT Quality:** Translatability and diagnosis will be compared to the actual output of the different MT engines: Are all segments which are predicted to be difficult really bad translations? Are all segments which the diagnosis tool claims to be difficult really bad translations?

  The MT quality research proper will be done based on a corpus of three domains in two languages. Reference translations to existing engines will be used to determine if the quality of the TQPro components will be superior or not.

Tests will use corpus data provided by the TQPro partners with real translation projects (SAP, Logoscript, Lotus, CAT). Reference data (terminology, MT entries etc.) will also be available.

Literature

BCG00 Bel, N., Busa, F., Calzolari, N., Gola, E., Lenci, A., Monachini, M., Ogonowski, A., Peters, Y., Peters, W., Ruimy, N., Villegas, M., Zampolli, A., 2000: SIMPLE - A general Framework for the development of Multilingual Lexicons. Proc. LREC Athens.

HV1999 Heidemann, B., Volk, V.: Evaluation of Terminology Extraction Tools. Univ. Zurich

McC2000 McCormick, S., 2000: Using OLIF2 to Share and Re-use Your Lexical and

Terminological Data. Proc. TC22, London

MMM94 Mitchell, M., Kim, G., Marcinkiewicz, M.A., Maclntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B., 1994: The Penn TREEBANK: Annotating Predicate Argument Structure. http://www.ldc.upenn.edu/doc/treebank2/arpa94.html

MS1999 Manning, C.D., Schütze, H., 1999: Foundations of Statistical Natural Language Processing. MIT Press.

OpT1998 OpenTag Format Specifications, V1.2, Nov. 1998. http://www.opentag.com/opentag.htm

Thul997 Thurmair, Gr., 1997: Exchange Interfaces for Translation Tools. Proc. MTSummit San Diego

Thu2000 Thurmair, Gr., 2000: Machine Translation. Presentation at Leuven Univ., Kortrijk (unpublished)

TMX2000 TMX Format Specifications, Version 1.2, June 2000. http://www.lisa.org/tmx/