

Tutorial: Using MT in Practice—Helpful Hints and Advice

Karin Spalink and Muriel Vasconcellos

In today's information age, the speed factor that originally fueled the development of machine translation is becoming less and less important. To the extent that time (and money) can no longer be saved by translating faster, text-independent factors become more important. Saving money with a machine translation system now boils down to ensuring a maximum fit with the operational setting.

The tutorial focuses mainly on batch-processed machine translation, as opposed to interactive machine translation or machine aids for the translator. However, mention will be made of the recent trend toward environments of bundled software in which both batch machine translation and machine aids, including terminology lookup and translation memory, can be called as the user proceeds works on a given text.

Assessing an MT system today means checking for incompatibilities with the operating environment that it will be part of. In evaluating an MT system, it's important to distinguish between formal and functional approaches.

An important factor in maximizing the benefit of MT is using the contribution of posteditors as efficiently as possible, not only in terms of their practical skills but also in terms of introducing improvements in the MT system.

In the pre-translation phase, the text-independent factors that need to be taken into account include the language combination(s), the platform(s) being used, the requirements for file conversion and/or compatibility with networks, the usability of the system itself (mainly the user interface and facilities for updating the dictionary), and the cost of purchasing, licensing, or installing the software. On the other hand, the factors that are text-dependent include the text type or genre, the complexity of the text, its repetitiveness, the format of the data, and the need to handle graphics and formulas.

Once the automatic translation phase is completed, the practical skills of the posteditor come into play. Many of these skills are procedural and involve reacting to the text that the machine has presented. In a truly efficient environment, however, the posteditor also assumes a proactive role and regularly feeds information back to the system so that it will be increasingly tailored to the environment and require less reactive postediting in the future. This involves formal control. For example, in the case of updating the dictionaries, each step, from finding a missing or incorrect entry to getting the correct data in the dictionary, including a testing phase.