The Multilingual Generation Game: authoring fluent texts in unfamiliar languages*

Donia R Scott
ITRI, University of Brighton
Lewes Road
Brighton BN2 4AT, UK
email: Donia.Scott@itri.bton.ac.uk

Abstract

There are obvious reasons for trying to automate the production of multilingual documents. Among them are the rapidly growing need for such documents, the high cost and low availability of good translators, and the fact that translators often need more time than is available to produce good multilingual versions. These problems are compounded when equivalent versions of a document are needed in not just two or three, but many, languages — as is often the case in Europe, where there are now eleven official languages in the European Community.

This talk presents some recent developments in Multilingual Natural Language Generation (M-NLG). These allow the automatic production of high-quality multilingual documents, while avoiding many of the well-known pitfalls of the more familiar alternative of Machine Translation (MT) — for example, the difficulty of information extraction from a source document and the danger of source-language bias.

1 Introduction

Solutions from computational linguistics have long been applied to the problem of multilingual document generation — an issue of growing concern to governments and international agencies, companies marketing multinationally and global information providers. Despite the success of machine-aided translation, however, the problem of achieving high quality versions of documents in several languages within a reasonable time and at reasonable cost remains outstanding.

The need for (at least partially) automated multilingual documentation is to a large extent driven by the high cost and inherently slow production of professional manual translation. In dynamic global markets the reality often is that delays introduced by the extra time needed for translation can cost a company many thousands of dollars in lost sales. Human translation has two further drawbacks: (a) the strong potential for introducing source language bias, and (b) the poor support for management and maintenance for manually-produced multilingual versions of the same document.

Machine-aided translation brings a number of known advantages over human translation: lower production costs, higher translation speed, and improved maintenance support. Nevertheless, the quality of multilingual documents achieved through machine translation is of an undeniably inferior standard than those produced manually and the problem of source language bias remains.

The root of the problems with the MT approach lies primarily in the underlying representations of the text. First, it is rarely an accurate reflection of the content of the source text; this makes it extremely difficult for the target texts to contain the same information as its source. Second, it is very closely tied to the linguistic structure of the source text; this means that the structure of the target text is often more appropriate to the source language than to the target language.

2 Symbolic Authoring of multilingual documents

Symbolic Authoring is a recent approach to the production of good-quality multilingual documents. The main difference between it and machine translation can be seen by comparing Figures 1 and 2: Instead of writing a document in one language and submitting it to translation into other languages, the 'author' uses a special editor to create the underlying meaning of the document, and it is this symbolic source that is rendered into text. The advantage of symbolic editors coupled with multilingual natural language generation systems is that, as with machine translation, the document production costs are lower and the production speeds are higher than for human translation. The advantage over machine translation is that it provides good support for document management and maintenance, the problem of source language bias is removed (since there is no source natural language text involved) and the resulting texts are high quality drafts.

*The work described in this paper is part of an ongoing collaboration at the ITRI between the author and her colleagues Richard Power and Roger Evans.

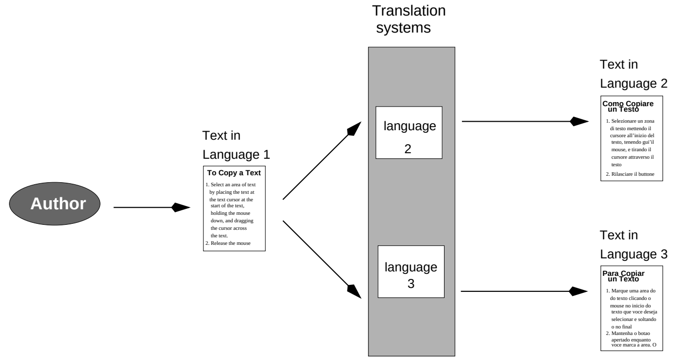

Figure 1: Multilingual documents through machine translation

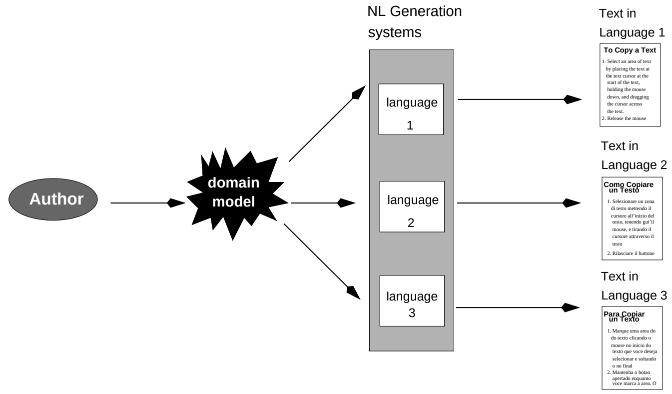

Figure 2: Multilingual documents through symbolic authoring

Symbolic editing allows the author to construct a non-linguistic knowledge source which can be used for a variety of purposes. It is particularly attractive for natural language generation since it gives the author of the document explicit control over the *meaning* of generated text, leaving the *rendition* of this meaning up to the NL generator that receives it as input. This method of automatically generating text is especially useful for multilingual documents, since all language versions are derived from the same non-linguistic source and multilinguality can be achieved without language translation.

With this approach the author can guarantee that all language versions will have identical content. Consistency is also guaranteed not only between documents but within them. Although the same information may be mentioned in various different parts of the same document and its various language versions, it will have a single representation in the domain model; updating the texts becomes a matter of editing the domain model. By then regenerating the model, the updated texts are both internally consistent and pragmatically congruent across language versions. A growing number of systems make use of this new approach.

- EXCLASS is an intelligent support tool for personnel officers writing (bilingual English and French) job descriptions. The user builds the job description by composing and editing conceptual representations; these representations are trees of concepts from a structured conceptual dictionary. Concepts are presented to the user through diagrammatic trees with natural language labels [Caldwell and Korelsky, 1994].

- DRAFTER-I is an authoring tool to support technical authors and software developers in writing (bilingual English and French) software manuals. The user directly builds the domain model (semantic knowledge base) describing the procedures for using a selected software application. As it is being constructed, the model is presented to the user through diagrams and fragments of text [Paris *et al.*, 1995].

- GIST is an authoring tool to support forms designers. It generates (multilingual English, German, Italian) forms in the domain of social administration. The user's interaction with GIST is very similar to that with DRAFTER-I. [Power and Cavallotto, 1996].

- A tool to support inventors in the authoring of (English only) patent claims allows the user to build a semantic model of the invented apparatus by selecting (via multiple-choice menu options) the apparatus parts, their functions and relations to each other [Sheremetyeva *et al.*, 1996].

These systems all comprise one or more natural language generators coupled to an interface that supports the manual creation of the generator's input (i.e., the authoring of the symbolic (conceptual) content of the output document). This interface is a type of knowledge-editing tool with a graphical browser, similar to the Generic Knowledge-Base Editor [Paley, 1996] and the CODE4 Knowledge Management System [Skuce and Lethbridge, 1995]. Graphical knowledge-editing tools have the advantage of allowing the author to edit knowledge *directly* rather than *indirectly* via text editing.

In principle, writers of symbolically-authored documents should be experts in the subject of the document, but they should not have to be linguists or computer scientists. Although the systems described above do not require knowledge of linguistics or familiarity in any of the languages of the output texts, they do however require the user to be fairly highly skilled in areas of computer science and AI. For example, the graphical representation of the knowledge is often a network diagram; these cannot be understood and developed without training in such technical notions as 'object', 'attribute' and 'value restriction'. Equivalent drawbacks hold for plan-based representations. Empirical studies have shown that domain experts typically produce high error rates when using such graphical object-oriented modelling tools [Kim, 1990].

Obviously, for symbolic authoring to be an effective alternative to machine translation, the author must be offered a more familiar representation of the knowledge being constructed. Despite the well-known disadvantages of natural language as a knowledge representation language, there is strong evidence to suggest that text offers a clear advantage over graphics for understanding complex knowledge structures [Petre, 1995]. This problem thus becomes one of how to achieve symbolic authoring with the advantage of a textual interface but with none of the disadvantages of textual interpretation that continue to hound Machine Translation.

3 Authoring multilingual documents with WYSIWYM

This talk presents a new method of symbolic authoring we have developed that can be used by anyone familiar with menus and hypertext browsing, which we call *WYSIWYM editing*[1] [Scott *et al.*, 1998; Power and Scott, 1998; Power *et al.*, 1998]. Using this method, the 'author' constructs the symbolic source (i.e., the domain model) by selecting and expanding portions of a natural language text that is generated in his or her language of choice; each expansion updates the domain model, which is then regenerated in one of two modes:

a feedback text: this is a specially constructed text with hypertext-like anchors that are the entry points to further expansion of the domain model.

an output text: this is the polished text that the author is aiming to achieve.

While in feedback mode, the author can perform routine editing operations such as *cut, copy,* and *paste.* In doing this, the author appears to be editing text, but this is illusionary: the operations are taking place on the domain model, which is then immediately re-described in generated text.

This method of editing a knowledge base kills two birds with one stone: the source is still a knowledge base, not a text, so no problem of interpretation arises; but the source is presented as a text, and so is easily understandable to the author.

Figure 3 shows the architecture of WYSIWYM authoring of multilingual documents. The basic WYSIWYM system consists of three components:

- A module for building and maintaining knowledge bases.[2]
- Natural language generators for the languages required.
- A user interface which presents the feedback and output texts to the author.

At any point in the authoring process, and in either mode, the author can switch to another language.

[1] WYSIWYM stands for "what you see see is what you meant". This method of knowledge base editing is patent pending.

[2] This includes a 'T-box' which defines the concepts and relations from which assertions in the knowledge base, the 'A-box', will be formed.

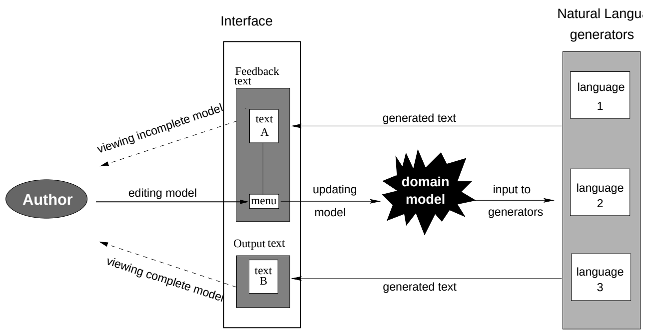

Figure 3: Architecture of WYSIWYM editing

Switching languages is equivalent to changing font in a text editor: it simply re-presents the very same content in a different way.

We have developed three authoring tools of multilingual documentation using WYSIWYM, in the domains of software applications (DRAFTER-II), patient information leaflets to accompany pharmaceutical products (PILLS) and explanations of legal reasoning in the field of maritime law (CLIME).

4 Significance of WYSIWYM editing

This method of writing multilingual documents provides several advantages over human translation, machine translation and over previous methods of symbolic authoring:

- WYSIWYM authoring allows a writer speaking *one* of the supported languages, but with no technical knowledge of linguistics or computation, to produce good output texts in *all* of them without the need for translation.
- Authors can understand feedback texts much better than they understand alternative methods of presenting knowledge bases, such as network diagrams. They require no formal knowledge of specialised knowledge representation languages or of the target languages, or any special training in a controlled language.
- Since the knowledge base is presented as a document, large knowledge bases can be navigated by the methods familiar from books and from hypertext documents (e.g., content page, index, hypertext links) obviating any need for special training in navigation.
- Our experience has been that people can learn to write documentation with WYSIWYM within a few minutes.[3]

[3] We are now embarking on formal usability trials.

- Since the knowledge base is presented in natural language, it becomes immediately accessible to anyone peripherally concerned with the documentation project (e.g., management, public relations, domain experts from related project). System documentation, often a tedious and time-consuming task, becomes automatic.
- The knowledge base can be viewed and edited in any natural language supported by the system; further languages can be added as needed. When supported by multilingual generators, WYSIWYM editing obviates the need for all but the most trivial aspects of traditional language localisation of the interface to the knowledge base.
- In contrast to typical expert systems, the knowledge base constructor need no longer be a knowledge engineer engaging in knowledge elicitation with a domain expert. In contrast to typical documentation scenarios, the author of multilingual documentation need no longer be a technical writer or translator. Instead, the 'author' using WYSIWYM can now be the domain expert himself.
- The style of the output text is achieved by tuning of the generators. New styles can be added at any time (e.g., company house styles, terminology suitable for novices rather than experts).
- WYSIWYM editing is ideal for facilitating knowledge sharing and transfer within a multilingual project. Speakers of different languages can view and modify the knowledge base in his or her own language.
- In many fields, documents (whether initially or after version updates) are the result of several authors. This makes it difficult to maintain consistency of content and style. With WYSIWYM authoring, such consistency is guaranteed both within and across all documents, whatever the target audience (e.g., software maintenance manuals vs user manuals, drug leaflets for patients, doctors or pharmacists) and whatever the target language.

The crucial advantage of this method of automating the production of multilingual documents compared to machine translation is that it eliminates the usual problems associated with parsing and semantic interpretation. Feedback texts with menus have been used before in the NL-menu system [Tenant *et al.*, 1983], but only as a means of presenting *syntactic* options, guiding the user by listing the extension of the current sentence that are covered by its grammar. A similar approach has been used in the Adaptive Forms system [Frank and Szekely, 1998]. Both make parsing more reliable by enforcing adherence to a sublanguage, but parsing and interpretation are still required.

Of course, text-based interfaces to knowledge bases do not always provide the clarity of expression that can be achieved with knowledge representation languages. One of the questions we are currently exploring is what is the appropriate style for the feedback texts; since they are not required to be as elegant and fluent as the output texts, it seems appropriate to, for example, expect them to be free of ambiguity where possible, even at the expense of elegance. Obvious places to start are co-reference specification (i.e., should the feedback text clearly show all co-references by artificial means, e.g., subscript indices, colour etc.; or should they be restricted to natural language, e.g., using pronouns etc.) [van Deemter and Power, 1998]. Similar issues apply to the well known problem of scope of quantifiers, which can be difficult to express unambiguously in text. Some of these problems, however, may not be easily resolvable in the feedback text without adding new non-textual interface devices which may eventually lead to such a complicated interface that we are no better off than if we had stuck to graphics in the first place. Just where the boundaries of this tradeoff between clarity and ease of use lie remains an open issue.

References

[Caldwell and Korelsky, 1994] D. Caldwell and T. Korelsky. Bilingual generation of job descriptions from quasi-conceptual forms. In *Proceedings of the Fourth Conference on Applied Natural Language Generation*, 1994.

[Frank and Szekely, 1998] M.R. Frank and P. Szekely. Adaptive forms: an interaction technique for entering structured data. *Knowledge-Based Systems*, 11:37–45, 1998.

[Kim, 1990] Y. Kim. Effects of conceptual data modelling formalisms on user validation and analyst modelling of information requirements. PhD thesis, University of Minnesota, 1990.

[Paley, 1996] S. Paley. Generic knowledge-base editor user manual. Technical report, SRI International, California, 1996.

[Paris *et al.*, 1995] Cécile Paris, Keith Vander Linden, Markus Fischer, Anthony Hartley, Lyn Pemberton, Richard Power, and Donia Scott. A support tool for writing multilingual instructions. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1398–1404, Montreal, Canada, 1995.

[Petre, 1995] M. Petre. Why looking isn't always seeing: readership skills and graphical programming. *Communications of the ACM*, 38(6):33–42, 1995.

[Power and Cavallotto, 1996] R. Power and N. Cavallotto. Multilingual generation of administrative forms. In *Proceedings of the 8th International Workshop on Natural Language Generation*, pages 17–19, Herstmonceux Castle, UK, 1996.

[Power and Scott, 1998] R. Power and D. Scott. Multilingual authoring using feedback texts. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pages 1053–1059, Montreal, Canada, 1998.

[Power *et al.*, 1998] R. Power, D. Scott, and R. Evans. What you see is what you meant: direct knowledge editing with natural language feedback. In *Proceedings of the 13th Biennial European Conference on Artificial Intelligence*, pages 675–681, Brighton, UK, 1998.

[Scott *et al.*, 1998] D. Scott, R. Power, and R. Evans. Generation as a solution to its own problem. In *Proceedings of the 9th International Workshop on Natural Language Generation*, pages 256–265, Niagara-on-the-Lake, Canada, 1998.

[Sheremetyeva *et al.*, 1996] S. Sheremetyeva, S. Nirenburg, and I. Nirenburg. Generating patent claims from interactive input. In *Proceedings of the Eighth International Workshop on Natural Language Generation*, pages 61–70, Herstmonceux, Sussex, June 1996.

[Skuce and Lethbridge, 1995] D. Skuce and T. Lethbridge. CODE4: A unified system for managing conceptual knowledge. *International Journal of Human-Computer Studies*, 42:413–451, 1995.

[Tenant *et al.*, 1983] H. Tenant, K. Ross, R. Saenz, C. Thompson, and J. Miller. Menu-based natural language understanding. In *Proceedings of the Association of Computational Linguistics*, 1983.

[van Deemter and Power, 1998] K. van Deemter and R. Power. Coreference in knowledge editing. In *Proceedings of the COLING-ACL workshop on the Computational Treatment of Nominals*, pages 56–60, Montreal, Canada, 1998.