Chapter 8

Input

8.1 Introduction

In the scenario we imagined in Chapter 2, the text was delivered in the form of a machine-readable document, having been prepared in such a way as to facilitate translation. This is an important time saver. In this chapter, we describe how the full potential of machine readable texts can be exploited in three ways: first, by adopting the notion of an 'electronic document' and embedding an MT system in a complete document processing system; second, by restricting the form of input by using simplified or **controlled language**; and third, by restricting both the form, and the subject matter of the input texts to those that fall within a **sublanguage** — it is here that the immediate prospects for MT are greatest. The common theme of this chapter is how the successful application of MT can be enhanced by ensuring that the input to the system is 'appropriate'. Briefly, the message is this: having texts in machine readable form is a prerequisite for sensible use of MT, but one can get much better results by (i) adopting certain standard formats for the input, (ii) controlling the input, so that problematic constructions, etc., are avoided, and (iii) where possible, tailoring the MT systems to the language of particular domains.

8.2 The Electronic Document

8.2.1 Basic Ideas

Every text that is not delivered as an electronic document on a floppy disc, a magnetic tape, or via a computer network will have to be put into the system manually. Re-typing a text into the computer solely to make it available for MT is unlikely to be cost-effective — it would often be quicker to have the text translated directly by a human translator. In recent years it has become practicable to use an optical character reader (OCR) to input text available only in printed form. Clearly this is much quicker than re-typing, but checking for and correcting scanning errors can be time-consuming.

However, if as is the case with ETRANS as described in Chapter 2, the MT system fits into an overall document production system (DPS), then text can be created, translated, re-edited and generally prepared for publication within the same electronic environment. In the first part of this chapter we will explore this notion of an electronic document in some detail.

The Risks of Office Automation

Electronic text is simply text which is available in a machine readable form. For example, electronic text is produced by ordinary office word processors. At its simplest, such a text is just a sequence of characters, and, for the characters in use in general computing (i.e. the English alphabet, normal punctuation characters, plus characters such as the 'space' character, the 'line-feed' character, etc.) there is a standard representation provided by the ASCII[1] codes, which associates each character with a seven or eight bit code (i.e. a number — e.g. *a* is ASCII 97, *b* is ASCII 98, *A* is ASCII 65, the 'space' character is ASCII 32). Unfortunately, this standard is not sufficient for encoding the letters of foreign alphabets and their accents, even those based on the Roman alphabet, let alone non-Roman alphabets, and characters in non-alphabetic scripts, such as Japanese characters (Kanji). One approach to such alphabets is to extend the ASCII codes beyond those needed by English. Another is to represent foreign accents and special characters by sequences of standard ASCII characters. For example, a German *u* with umlaut (*ü*) might be represented thus: `\"{u}`.

[1] ASCII stands for American Standard Code for Information Interchange.

One problem is that there is (as yet) no genuine accepted standard beyond basic ASCII, with the further complication that many word processors use non-ASCII representations 'internally', as a way of representing text format (e.g. information about typeface, underlining, etc.) This lack of standards means that it is necessary to use special conversion programs if one wants to freely import and export text from different languages and a variety of DPSs (such as word processors). Even when such programs exist, they do not always preserve all the information (e.g. some information about format may be lost).

Part of a general solution to these problems, however, is to distinguish two components of a printed document: the text itself (a sequence of words and characters); and its rendition — the form in which it appears on the page (or screen). For example, consider a title or heading. There are the words which make up the title — perhaps a noun phrase such as 'The Electronic Document' — and the particular presentation or rendition of those words on the page. In this book all section and chapter headings are aligned with the left margin and different levels of heading (chapter, section, subsection) are printed in a distinctive typeface and separated by a standard space from the preceding and following paragraphs of text.

If we think about this distinction between text and rendition in electronic terms, it is easy to see that we have to code both the characters in the text, and indicate how we intend parts of that text to appear on screen or in printed form. In the early days of electronic text handling, this problem was solved in a rather direct and obvious fashion: the author would type in not only the substance of the text but also some special codes at appropriate places to tell the printer to switch into the appropriate type faces and point size. For example, in typing in a title the author would carefully insert an appropriate number of carriage returns (non-printing characters which start a newline) to get a nice spacing before and after. She would also make sure the title was centred or left-aligned as required, and finally she would type in special codes (say `\[223\[-447`) before and after the title string to switch the printer into a bold typeface with 24 'points' to the inch and back to its usual font and size immediately afterwards.

There are three evident problems with such a procedure:

1. The codes used are likely to be specific to particular printers or word processing set-ups and hence the electronic document will not be directly portable to other systems for revision, integration with other documents or printing.
2. The author is required to spend some of her time dealing with rendition problems — a task that (prior to the advent of electronic systems) had always been conveniently delegated to the compositor in a printing house.
3. If at some point it is decided that a different rendition of headings is required, someone has to go through the entire document and replace all the codes and characters associated with the rendition of each heading.

The printer codes are a sort of tiny little program for a particular printer. The next development was to replace these rather specific programs by some means of stating di-

rectly "I want this in 24 point Roman boldface" — perhaps by a 'markup' like this: '`\roman\24pt\bf `'. Each printer or word processor can then be equipped with a special program (a so-called 'driver') which interprets this high-level code and sends the printer or screen appropriate specific low-level codes. Providing everyone used exactly the same high-level codes in all systems, the problem of portability would be solved.

However, there is another way of tackling the rendition problem. When one thinks about it abstractly, the only thing that the author really needs to put into the text is some markup which says (in effect) 'This is a heading', or 'This is a footnote' or 'This is an item in an item list' and so on. Each piece of text is thus identified as being an instance of some class of text elements. With such markup, the author no longer has to worry about *how* each such marked document element is going to be printed or shown on screen — that task can be delegated to the document designer (the modern equivalent of a compositor). The document designer can specify an association between each type of document element and the high-level rendition codes she wants it to have. In other words, she can say that she wants all headings to be printed in 24 point boldface Roman. The document handling system then ensures that headings etc. are displayed and printed as required.

This type of markup, where the author simply identifies particular pieces of text as being instances of particular document elements, is known as descriptive or 'intensional' ('intentional') markup. This notion is fundamental to all modern document processing systems and techniques. Not only does this provide flexibility in how text is rendered, provided that the way in which markup is made is consistent from system to system, the result is that electronic documents can be freely passed between systems.

We can now be a little more precise about the notion of an electronic document: it contains electronic or machine-readable text with descriptive markup codes which may be used to determine the rendition and other usages of the document. Before we go on to give an idea of how this can be exploited for MT, it may be worth a brief description of the standard descriptive markup: SGML (Standardised General Markup Language) which is specified by the International Standards Organization. It is our belief that in the next few years no serious commercial MT system will be supplied without some means of handling SGML.

SGML specifies that, ordinarily, text will be marked up in the way shown in the last example above, i.e. with document elements surrounded by their names in angle brackets. An office memo marked up in SGML might look like the example below. In addition to the actual text, various pairs of SGML tags delimiting the memo elements can be seen here. The memo as a whole starts with `<Memo>` and ends with `</Memo>` (where / indicates the closing delimiter). In between the Memo tag pair we find the sub-elements of the memo, also marked-up with paired tags (`<To>... </To>`, `<From> .. </From>`, `<Body>... <P>..</P>...</Body>`).

The relationship between SGML tags, and the way text is actually rendered is given by an association table, such a table might say, e.g. that the body of a memo should be separated from the previous part by a horizontal line. When actually printed, this memo might look as in Figure 8.1:

A Memo Marked Up in SGML

```
<Memo>
<To>Mary Dale, Purchasing</To>
<From>Tony Burrows</From>
<Body>
<P>We would like to order 4 Sun ELCs with an
additional 8M of memory.  We don't need any ex-
ternal drives.</P>
<P>By the way, have you managed to get any
more info on SGML parsers for PCs?  Or on SGML
parsers for anything?</P>
</Body>
</Memo>
```

The tagging principles of SGML are intended to extend to very complex and highly structured documents. Imposing such a structure not only allows very fine, and flexible control of how documents are printed, it can also allow easy access to and manipulation of information in documents, and straightforward consistency checking[2].

One thing the SGML standard does not do is try to specify a standard inventory of all possible document elements. Users are perfectly free to define their own document types and to specify the elements in those documents. SGML provides a special method of doing this known as a **Document Type Definition** (DTD). A DTD is a sort of formal grammar specifying all such relations in a particular type of document. For example, such a grammar might say that all Memos (all our Memos at least) contain a *To* element followed by a *From* element followed by a *Body* element, which itself contains at least one *Paragraph* followed by zero or more *Paragraphs*. This means that a *Memo* has the following sort of DTD (grossly simplified):

```
Memo → To, From, Body

Body → Paragraph, Paragraph*
```

Using a DTD has several advantages:

1 The DTD makes sure that documents are truly portable between different SGML

[2]For example, suppose one has a printer manual marked up in this way, with special markup used for the names of printer components wherever they occur. It would be very easy to extract a list of printer parts automatically, together with surrounding text. This text might be a useful addition to a parts database. As regards consistency, it would be easy to check that each section conforms to a required pattern — e.g. that it contains a list of all parts mentioned in the section.

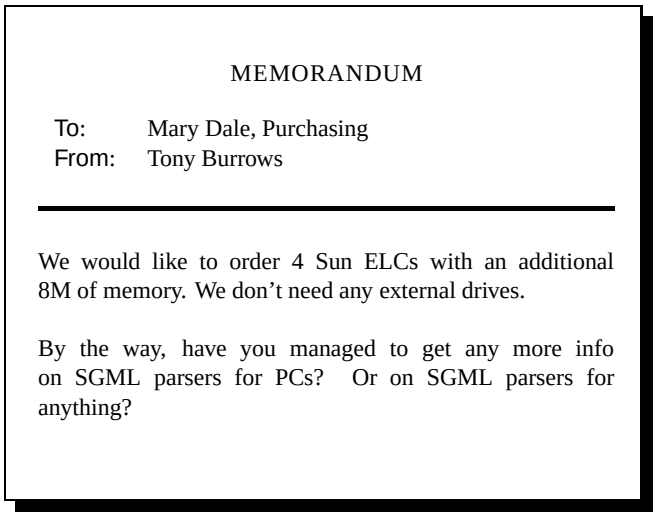

Figure 8.1 How a Memo Marked Up in SGML Might Appear When Printed

document systems; the document system reads the accompanying DTD to find out what sort of elements will be in the document and how they will be arranged with respect to each other. Thus, the document processing system knows what to expect when it encounters a document which is an instance of a certain DTD.

2 It ensures that documents of a particular type (e.g. user manuals) are always structurally consistent with each other. It suffices to define a DTD for the class of user manuals and then the SGML document processing system will ensure that all documents produced by that DTD will indeed have the same overall structure. In short, DTDs help to promote a certain rigour which is extremely desirable in technical documentation.

3 The use of DTDs in document preparation allows authors to deal directly with the content of texts whilst having little or no direct contact with the actual markup used. What happens with the usual sort of SGML system is that there is a window offering the author a choice of document entities appropriate for the document she is preparing or revising. This list of document entities is obtained by reading the DTD for the document. For example, in a memo, there will be a choice of *To*, *From*, and *Body*. The author clicks on the appropriate element and the markup is entered into the text (perhaps invisibly). When actually typing in the *Body*, the choice is narrowed down to *Paragraph*. Whilst this is not particularly interesting for simple documents like memos, it is clear that it would be be immensely useful in constructing complex documents, and in document retrieval.

With this general idea of Electronic Documents and markup, we can look at how an MT system can exploit the fact that texts are represented in this way.

8.2.2 SGML Markup and MT Input

An MT system should only attempt to translate things that are translatable. Suppose that some text contains the acronym 'MAT', which refers to a company called 'Machine Aided Translation Ltd'. Clearly the correct translation of this is either just MAT again or some new acronym that reflects the translation of the underlying name — perhaps TAO in French, being the acronym for *Traduction Assistée par Ordinateur*, which itself is the translation of *Machine Aided Translation*. What is unquestionably incorrect is a translation of the form *pallaison*, this being the sort of mat that a cat might sit on. The reader may think that the MT system ought to have spotted that MAT cannot be a standard concrete noun because it is capitalised; but many MT systems routinely ignore capitals because they need to recognise ordinary words which can appear with an initial capital letter at the start of a sentence.

The way to deal with this sort of problem is to ensure that acronyms are recognised as a particular class of text elements and marked up as such. This might be done (a) either by the author when the text is being created or (b) by special tools used before translation which help translators to find acronyms and the like and mark them up accordingly. For example, a specialised search and replace tool inside the document pre-editor could look for all sequences of capitalised words and, after querying the translator to check whether a particular candidate sequence really is an acronym, insert the appropriate markers in the text. The point is that once the text is marked up, the MT system is in a much better situation to know that it is dealing with an untranslatable acronym and to treat it accordingly.

Similarly, consider figures and diagrams in a document. These consist usually of pictorial material, which is untranslatable, and a translatable text caption which characterises the pictorial material. Recognising the markup tags which indicate that the following material in the document is pictorial, the MT system can simply ignore everything until it encounters another tag telling it that it is about to see the caption, which it can translate as a normal piece of text. Equally, it is easy to ask the MT system to translate (say) just a single chapter, because the markup in the document will clearly identify the piece of text that constitutes the chapter. Markup is thus a powerful tool in controlling the MT process.

DTDs are particularly useful in MT. Some MT systems keep a copy of each sentence they have already encountered together with its translation (the post-edited version, if available). This habit is known in the industry as **Translation Memory**. Over the years, MT vendors have found that in some organizations much of the translation workload consists of entirely re-translating revised editions of technical manuals. These revised editions may contain as much as 90% of the material that was already present in the previous edition — and which was already translated and post-edited. Hence automatically recognising sentences already translated and retrieving the post-edited translation - as the Translation Memory technique allows — results in a 90% reduction in post-editing costs (and an enormous increase in the overall speed of the translation process). This is clearly very significant.

However, these sort of performance improvements are really the result of a defective doc-

umentation process. The problem is that the organization paying to have the translation done is not keeping proper track of which parts of a revised document really are different from the original version. Clearly only new or altered material really needs to be even considered for translation.

Within the SGML standard it is possible to add features to text elements to record when they were last altered, by whom and so on. This version control information can be maintained by the document system and it allows the user to extract revised elements. Indeed, the principle can be extended so that earlier versions of a given revised element are also kept, allowing the user to reconstruct any previous version of a document at any point.

The result of exercising proper version control in documentation is that only new elements for which there are no existing translations will be submitted to the translation process. In this way, the document processing system takes some of the burden otherwise carried by the MT system (viz, the 'Translation Memory' facility).

Another advantage of using DTDs in MT involves generalizing the notion of a document slightly, to introduce the notion of a 'multilingual document'. In SGML, this is largely a matter of altering the DTDs of monolingual document types. Take the Memo example: we can get a multilingual version by specifying that there is a copy of each document element for each language. Here is a revised (and still simplified) Memo DTD for two languages:

```
Memo → To, From, Body

Body → Paragraph, Paragraph*

Paragraph → Paragraph-L1, Paragraph-L2
```

There are now two types of Paragraph — Paragraphs in language one and Paragraphs in language 2. Each Paragraph element will contain one language 1 paragraph followed by one language 2 paragraph. (There are no language specific **To** and **From** elements because it is assumed that they contain only proper names). This sort of technique can be generalised to allow a document to carry text in arbitrarily many languages. Though this allows a document to contain text for more than one language, it does not require it — document elements can be empty — this would be the case for target language elements where the source element has not yet been translated.[3]

The important thing to understand here is that just because the simple multilingual DTD we have described 'interleaves' the elements for different languages (we have a paragraph for

[3]Although most elements of the structure are exactly matched, there may sometimes be differences. For example, if the document element Paragraph is composed of document element Sentence(s), it is perhaps unwise to insist that each Sentence in each language is paired exactly with a single corresponding Sentence in every other language, since frequently there is a tendency to distribute information across sentences slightly differently in different languages. However, at least for technical purposes, it is usually perfectly safe to assume that the languages are paired Paragraph by Paragraph, even though these units may contain slightly different numbers of sentences for each language.

L1 followed by the corresponding paragraph for L2, etc.), this does not mean that we have to view the document that way. For example, a Memo in English, French and German can be viewed on the screen of a document processing system with all the English paragraphs, printed together, and the French paragraphs printed alongside, with the German paragraphs not shown at all. Part of the flexibility in the rendition of a marked-up document is that the text content of classes of elements can be hidden or shown at will. In practical terms, this means that a translator editing a multilingual document will have considerable flexibility in choosing the way in which that document is presented (on screen or on paper) and in choosing the type of element she wishes to see.

Turning back to the MT case, recall that in the scenario in Chapter 2, ETRANS takes the German text and then makes available the English translation in the multilingual document. It should now be much clearer how this works. Translatable elements from the source text are passed to the ETRANS system which then translates them. The translated text is then placed under the corresponding target language text elements (which, up that point, have been entirely empty of text). So far as is linguistically possible, the structure of the document is preserved.

In summary, it should be clear that the general idea of the Electronic Document is important within the context of MT and can make a considerable contribution to the successful integration of MT within the office environment.

8.3 Controlled Languages

The notion of controlled languages was introduced in Chapter 2 where we described it as a form of language usage restricted by grammar and vocabulary rules. The original idea arose during the 1930s, when a number of influential linguists and scholars devoted considerable effort to establishing a 'minimal' variety of English, a variety specifically designed to make English accessible to and usable by the largest possible number of people world wide. *Basic English*, as it was called, differed from previous attempts to construct universal languages in that it was a perfectly well-formed part of English, rather than some entirely artificial or hybrid construction such as Esperanto. One of the central ideas of the Basic English movement was that the number of general-purpose words needed for writing anything from a simple letter of receipt through to a major speech on the world economic situation could be a few hundred rather than the 75 000 upward available to skilled native speakers. This lexical economy was to be achieved in part by using 'operator verbs' with the set of nouns and adjectives to stand in for the vast number of derived verbs which are frequently used. For example, whereas in ordinary English we might write *The disc controller design was perfected over numerous revisions*, Basic English would say ... *was made perfect ...*, where *make* is one of the operator verbs and *perfect* one of the licensed Basic English adjectives.

The authors of Basic English explicitly recognised that the dictionary would need to be extended with special terminology for scientific and technical writing. However, even if a text contained terminology specific to a certain subject field, the general language

component of the text could perfectly well be accommodated within Basic English. The important point remains that, for writing in a particular subject field, no more is needed than the Basic English dictionary together with a (relatively small) technical vocabulary for that field.

The idea was later taken on by English-language based (predominantly North American) corporations marketing capital goods on a world-wide basis. Rather than try to translate engine manuals and the like into every possible language that might be required, it was assumed that if they were written with sufficient care and attention, they could be read fairly easy by service engineers and mechanics with limited English skills.

Although controlled languages were introduced partly to avoid or reduce human translation costs, two important additional benefits were discovered. First, the readability and clarity of a controlled language technical text often seems better than uncontrolled texts — even for native English readers. Second, controlled languages produce better results with MT than uncontrolled languages.

The reasons for controlled languages' superior MT performance are easy to understand. First, the restricted vocabulary means that fewer words need to be added to the MT system dictionaries and more effort can be put into getting the entries which are required right. Second, the grammar component of the system can be tailored to handle all and only those constructions which are licensed by the controlled language specification, a specification which excludes the most difficult and ambiguous constructions anyway.

A flavour of what is involved can be obtained by looking at the writing rules given above and the dictionary excerpt on page 149, which are based on those of *PACE*, the controlled English used by the UK Engineering company Perkins Engines.[4] As will be clear from the dictionary excerpt, the general principle is 'one word, one meaning', for example, the only use of the verb *advise* is 'to give advice'. Thus, a usage such as *Please advise us of the availability of parts at your earliest convenience* would not be allowed, since here it means 'tell'. A useful development of such a dictionary for MT purposes would be to add information about how these words translate.

Using a restricted pool of words and terms also means that the system dictionaries can be tailored (by the MT supplier or responsible translator) to cover exactly that set of words and their translations. Being consistent about the use of terms will also help to improve the overall consistency and quality of the texts being translated. After all, one of the simplest and most direct benefits of MT for technical texts is that terms are always translated consistently because they are simply looked up in an electronic bilingual term dictionary.

In general, it can be seen that the rules are mainly advice on constructions that should be avoided, usually because they lead to ambiguity. The rules for controlled languages tend to be stylistic guidelines rather than hard and fast grammar specifications. In general, much of the success of controlled languages as corporate language tools stems from the emphasis placed on critical analysis of the text and precise presentation of ideas. This is

[4] 'PACE' stands for 'Perkins Approved Clear English'.

The PACE Writing Rules

- **Keep it short and simple**:

  1 Keep sentences short.

  2 Omit redundant words.

  3 Order the parts of the sentence logically.

  4 Don't change constructions in mid-sentence.

  5 Take care with the logic of *and* and *or*.

- **Make it explicit**:

  6 Avoid elliptical constructions.

  7 Don't omit conjunctions or relatives.

  8 Adhere to the PACE dictionary.

  9 Avoid strings of nouns.

  10 Do not use *-ing* unless the word appears thus in the PACE dictionary.

A sample from the PACE Dictionary

advantage	n	Benefit
adverse	adj	Unfavourable
advice	n	Specialist Intelligence
advise,d	v	To provide advice
aerosol container	n	
affect,ed	v	To have an effect on
after	adv,prep	Being behind in succession, following something
again	adv	Once more
against	prep	In contact with
agglomerator	n	
agricultural	adj	Appertaining to agriculture
air	n	The gases that surround the earth
air charge cooler	n	

particularly apparent in the first example on page 151, which illustrates the dramatic effect of using a controlled version of English.

It is not particularly difficult to train people to write controlled language text i.e. text which generally observes some set of fairly simple writing rules. For example, the Xerox corporation currently offers its technical writers a one-day course in writing with MCE (Multinational Customised English, a Xerox proprietary language). British Aerospace teaches the rudiments of Simplified English (a general purpose technical English for the international aerospace industry) in a few fairly short training sessions.

The Effect of Using Controlled English

BEFORE:
It is equally important that there should be no seasonal changes in the procedures, as, although aircraft fuel system icing due to water contaminations more often met with in winter, it can be equally dangerous during the summer months.

AFTER:
Use the same procedure all the time, because water in the fuel system can freeze during winter or summer.

BEFORE: Loosen the dynamo or alternator mounting and adjustment link fasteners.

AFTER: Loosen the pivot fasteners of the dynamo or alternator mounting. Loosen also the fasteners of the adjustment link.

BEFORE: Reference to renewing the joints and cleaning of joint faces has to a great extent been omitted from the text, it being understood that this will be carried out where applicable.

AFTER: Normally the text does not include instructions to clean joint faces or to renew joints. These operations must be done, if necessary.

8.4 Sublanguage MT

In the previous section, we looked at a method of controlling the input to an MT system, simplifying it by avoiding certain uses of words, and avoiding potentially ambiguous constructions. Since the success of the METEO MT system, which we mentioned briefly in Chapter 1, an important strand of MT has involved concentrating on what we could loosely call 'MT for Special Purpose Languages', or sublanguage MT. Here, rather than imposing controls or simplifications on writers, one tries to exploit the restrictions in terms

of vocabulary and constructions that users of the language for specialized purposes normally accept, or simply observe without reflection. The term sublanguage refers to the specialized language used (predominantly for communication between experts) in certain fields of knowledge, for example, the language of weather reports, stockmarket reports, the language of some kinds of medical discussion, the language of aeronautical engineering. Specialized vocabulary is one characteristic of such 'languages' (they typically contain words not known to the non-specialist and also words used in different or more precise ways). However sublanguages are also often characterised by special or restricted grammatical patterns. In MT, it is quite common to use the term sublanguage rather loosely to refer not just to such a specialized language, but to its use in *a particular type of text* (e.g. installation manuals, instruction booklets, diagnostic reports, learned articles), or with *a particular communicative purpose* (communication between experts, giving instructions to non-experts, etc).

The chief attraction of sublanguage and text type restriction to MT researchers is the promise of improved output, without the need to artificially restrict the input. Restricting the coverage to texts of particular types in certain subject domains will allow one to profit from regularities and restrictions in syntactic form and lexical content. This may be important enough to permit significant simplification of the architecture, and certainly leads to a reduction in the overall coverage required. We reproduce an example from English to French output from METEO:

METEO: English-French Translation

```
METRO TORONTO.
TODAY...  MAINLY CLOUDY AND COLD WITH OCCA-
SIONAL FLURRIES. BRISK WESTERLY WINDS TO 50
KM/H. HIGH NEAR MINUS 7.
TONIGHT...  VARIABLE CLOUDINESS. ISOLATED FLUR-
RIES. DIMINISHING WINDS. LOW NEAR MINUS 15.
FRIDAY...  VARIABLE CLOUDINESS. HIGH NEAR MINUS
6.

LE GRAND TORONTO.
AUJOURD HUI...  GENERALEMENT NUAGEUX ET FROID
AVEC QUELQUES AVERSES DE NIEGE. VENTS VIFS
D'OUEST A 50 KM/H. MAXIMUM D'ENVIRON MOINS 7.
CETTE NUIT ...  CIEL VARIABLE. AVERSES DE NIEGE
EPARSES. AFFAIBLISSEMENT DES VENTS. MINIMUM
D'ENVIRON MOINS 15.
VENDREDI...  CIEL VARIABLE. MAXIMUM D'ENVIRON
MOINS 6.
```

Of course, the language of meteorological reports is special in happening to combine a

rather small vocabulary with a simple, telegraphic style of writing (notice in particular the complete absence of tenses from these extracts — the few verbs there are in non-finite forms). Nonetheless, a simplification of lexical and possibly syntactic coverage can be expected in less extreme cases. To give an example with respect to lexical coverage, it is reported that 114 of the 125 occurrences of the verb *to match* in a computer software manual translate into the Japanese *icchisuru-suru*, which is listed as one of the less frequent of the 15 translations given in a small-size English-Japanese dictionary. In the extract from a corpus of telecommunications text given below, *traffic* always corresponds to the French *trafic* and never to *circulation* (which applies only to road traffic). Moreover the dictionary writer can safely ignore the meaning of both *trafic* and *traffic* concerning dealings in illegal merchandise ('drug traffic'). Also, for an increasing number of sublanguages one can rely on the availability of a termbank (an on-line (multilingual) terminological dictionary) defining and stating equivalences for many of the technical terms that will be encountered. This greatly eases the job of dictionary construction. Such examples can be multiplied almost at will.

As for syntactic coverage, examples of instruction manuals and other forms of informative documentation typically share a number of common features. There will probably be no idioms, and a restricted set of sentential patterns. Another common feature is the relatively simple temporal dimension of the text, e.g. predominant use of the simple present. There is also the common occurrence of enumeration as a form of conjunction, usually either numbered or inset by dashes, etc. Some of these features can be seen by comparing the examples of English and French given below, which are drawn from a corpus of texts about Telecommunications. All are of great benefit to the developer or user of an MT system. For the developer, they mean that there are fewer problems of ambiguity, and development effort can be concentrated on a smaller range of constructions. For the user, this should mean that better coverage is obtained, and that the system performs better.

It is not, of course, the case that expository texts in different languages always exploit the same devices for a particular communicative purpose. The following extracts from the same corpus show that English and French differ in their use of impersonal constructions, with French favouring such constructions with the impersonal subject pronoun *il* ('it') far more in this type of text than English does. But even in these cases, it is generally easier to choose the correct translation, simply because the range of possibilities in such texts is smaller. (Literal translations of the phrases we have picked out would be: 'It is advisable to take account of...', It is manifestly much more difficult...', and 'It is advisable to take...'.)

(1) a. In this framework, the progressive evolution of the earth segment should be considered.
Dans ce contexte, il convient de prendre en compte l'évolution progressive du secteur terrien.

(2) a. Setting up a new satellite system, which may be either a regional system with the participation of a group of countries, or a purely national (domestic) system, is obviously much more difficult than using an existing

Extract from Telecommunications Bilingual Corpus

French:

La décision de mettre en oeuvre un nouveau système à satellites est la conséquence d'un processus à long terme qui peut être précédé des phases énumérées ci-après:

- utilisation du secteur spatial d'un système à satellites existant, généralement par location d'une certaine capacité de ce secteur;
- études économiques et techniques préliminaires de la validité et de la rentabilité d'un nouveau système, en tenant compte de la croissance du trafic et d'éventuels besoins de nouveaux services de télécommunications;
- expériences techniques et d'exploitation préliminaires, par exemple avec un satellite existant, si cela est possible, ou en lancant un satellite expérimental ou pré-opérationnel.

English:

The decision to implement a new satellite system usually results from a long term process, which may be preceded by the phases outlined below:

- Utilization of the space segment of an existing satellite system - usually by leasing space segment capacity.
- Preliminary economic and technical studies of the validity and profitability of a new system - considering the traffic growth and the possible need for new telecommunication services.
- Technical and operational preliminary experiments e.g. by using an existing satellite, if available, or even by launching an experimental or pre-operational satellite.

system:
Il est manifestement beaucoup plus difficile de mettre en place un nouveau système à satellites (système régional auquel participe un groupe de pays ou système purement national) que d'utiliser un système existant:

(3) a. Simultaneously, arrangements should be made for recruitment and training of staff for installation, operation and maintenance.
b. En même temps, il convient de prendre des dispositions pour le recrutement et la formation du personnel qui sera chargé de l'installation, de l'exploitation et de la maintenance.

Text type can strongly influence translation, not just because certain syntactic constructions are favoured (e.g. conjunction by enumeration), but also by giving special meanings to certain forms. An example of how the text type can be useful in determining translational equivalents is the translation of infinitive verb forms from French or German into English. Infinitives normally correspond to English infinitives, but are usually translated as English imperatives in instructional texts. Thus, in a printer manual one would see (4b) as the translation of (4a), rather than the literal translation.

(4) a. Richtige Spannung einstellen
'correct voltage to set'

b. Set correct voltage

(5) a. Exécuter les commandes
'to execute the commands'

b. Execute the commands

Thus, concentration on a sublanguage not only restricts the vocabulary and the number of source and target language constructions to be considered, it can also restrict the number of possible target translations. Given the potential that sublanguages provide for improvements in the quality of output of MT systems, and the fact that most commercial institutions do in fact have their major translation needs in restricted areas, it is not surprising that many research prototypes concentrate on restricted input in various ways, and that the design of tools and resources supporting sublanguage analysis is a major area of research.

8.5 Summary

In this chapter we have discussed three ways in which one can increase the likelihood of MT being successful by taking care with the input to the system. We first concentrated on the importance of integrating MT into the general document preparation environment, introducing the notion of the electronic document. We stressed the importance of standards in the encoding of texts, and showed how the process of MT can be aided by the adoption of the SGML markup language. In the following section, we turned to the content of texts themselves and introduced the notion of controlled languages, in which one adopts a simplified form of the language in order to communicate simply and unambiguously with the reader. Using a controlled language input greatly enhances the quality of output of MT systems. Finally we discussed sublanguage MT, or MT in restricted domains, observing that the language used in specialized technical domains is often quite different from and more restricted in style and content than the 'general language', and it is possible to take advantage of these characteristics by tailoring an MT system to the language of particular domains.

8.6 Further Reading

SGML is defined in ISO 8879, 1986, which is extensively discussed in the standard reference book on SGML: Goldfarb (1986). An excellent introduction to SGML is provided in van Herwijnen (1990).

The examples of the use of controlled language that we give in the text are based on those in Pym (1990). See Pym (1990); Newton (1992b) for discussion of the use of PACE as part of the translation operation in Perkins.

A noteworthy example of a controlled language is *Simplified English* (SE), which is described in the *AECMA/AIA Simplified English Guide* AECMA. This grew out of work done in the late 1970s, on behalf of the Association of European Airlines (AECMA) into readability of maintenance documentation within the civilian aircraft industry. As a result, an AECMA working group researched the procedural texts in maintenance manuals. It contains a limited general vocabulary of about 1500 words and a set of Writing Rules, similar to those we will describe above.

On sublanguage, Arnold (1990) provides a short overview. Lehrberger (1982) and Grishman and Kittredge (1986) are collections of articles on the subject. More detailed discussions can be found in Kittredge (1982), Kittredge (1987), Sager (1982), Slocum (1986), Teller et al. (1988) and Hirschman (1986).

Météo is described in (Hutchins and Somers, 1992, Chapter 12), see also Isabelle (1987). Recent developments are described in Chandioux (1976), Chandioux (1989a), Chandioux (1989b), and Grimaila and Chandioux (1992).

The example concerning the English-Japanese translation of *match* in software manuals is reported in Tsujii et al. (1992).