DICTIONARY ORGANIZATION FOR MACHINE TRANSLATION: THE EXPERIENCE AND IMPLICATIONS OF THE UMIST JAPANESE PROJECT

Mary McGee Wood, Elaine Pollard, Heather Horsfall,
Natsuko Holden, Brian Chandler, and Jeremy Carroll

Centre for Computational Linguistics
UMIST, P.O. Box 88
Manchester M60 1QD U.K.

ABSTRACT

The organization of a dictionary system raises significant questions for all natural language processing applications. We concentrate here on three with specific reference to machine translation: the optimum grain-size for lexical entries, the division of information about separate languages, and the level of abstraction appropriate to the task of translation. These are discussed, and the solutions implemented in the UMIST English-Japanese translation project are described and illustrated in detail.

The importance of the dictionaries in a machine translation system

In any machine translation system, the dictionaries are of critical importance, from (at least) two distinct aspects, their content and their organization. The content of the dictionaries must be adequate in both quantity and quality: that is, the vocabulary coverage must be extensive and appropriately selected (cf. Ritchie 1985), and the translation equivalents carefully chosen (cf. Knowles 1982), if target language output is to be satisfactory or indeed even possible.

The organization of a dictionary system also raises significant questions in translation system design. The information held about lexical items must be stored efficiently, accessed easily in a perspicuous form by the system and by the user, and readily extendable as and when required by the addition either of new lexical entries to a dictionary or of new information to existing entries. In this paper we discuss the way in which these issues have been addressed in the design and implementation of our English-Japanese translation system.

The UMIST Japanese project

At the Centre for Computational Linguistics, we are designing and implementing an English-to-Japanese machine translation system with monolingual English interaction. The project is funded jointly by the Alvey Directorate and International Computers Limited (ICL). The prototype system runs on the ICL PERQ, although much of the development work has been done on a VAX 11/750, a MicroVAX II, and a variety of Sun equipment. It is implemented in Prolog,in the interests of rapid prototyping, but intended for later optimization. For development purposes we are using an existing corpus of 10,000 words of continuous prose from the PERQ's graphics documentation; in the long term,the system will be extended for use by technical writers in fields other than software, and possibly to other languages.

At the time of writing, we have well-developed system development software, user interface, grammar and dictionary handling facilities, including dictionary entry in kanji, and a range of formats for output of linguistic representations and Japanese text. The English analysis grammar handles almost all the syntactic structures of the corpus. The transfer component and Japanese generation grammar currently handle a significant subset of their intended final coverage, and are under rapid development. A facility for interactive resolution of structural ambiguity has been implemented, and the form of its surface presentation is also being refined.

Foundations in linguistic theory

We are committed to active recognition of the mutual benefit of machine translation and linguistic theory, and our system has been designed as an implementation of independently motivated linguistic-theoretic descriptions. The informing principles are those of modern 'lexicalist' unification-based linguistic theories: the English analysis grammar is based on Lexical-Functional Grammar (Bresnan, ed. 1982) and Generalized Phrase Structure Grammar (Gazdar et al 1985), the Japanese generation grammar on Categorial Grammar (Ades & Steedman 1982, Steedman 1985, Whitelock 1986). These models share a general principle of holding as much information as possible as properties of individual lexical items or as regularities within the lexicon, rather than in a separate component of syntactic grammar rules; our system concurs in this, as will be detailed below.

The demands of translation

Many of the important questions in dictionary design for machine translation are common to all nlp applications. Before describing our actual implementation, we will briefly discuss three issues with specific reference to translation: the optimum grain-size for lexical entries, the division of information about separate languages, and the level of abstraction appropriate to the task of translation.

Firstly, what units should the entries in a machine translation dictionary system describe? In the interests of efficient and accurate translation, one should try to bring together all and only that information which is most likely to be used together. A grouping based on lexical stems of specified category appears to be optimal. Change of verb voice or valency across translation equivalents will not be uncommon. For example, an action with unexpressed agent will normally be described in English with the passive, in French by an active verb with impersonal subject, and in Japanese by an active verb with no expressed subject. Change of lexical category is more often not necessary; when it is, wider structural change is likely to be involved, and is better handled by syntactic than lexical relations.

Secondly, the optimum organization of multilingual information we take to be the clear separation of source from target languages. Our analysis and generation dictionaries are purely monolingual, with each entry including, not a direct translation equivalent, but a pointer into the transfer dictionary where such correspondences are mapped. For mnemonic reasons these pointers normally take the form of the lexical stem of the translation equivalent or gloss, but this is purely a convenience for the user, and should not obscure their formal nature, or the fact that contrastive information is held only in the transfer dictionaries.

Thirdly, one must consider the level of abstraction appropriate to the task of translation and thus to the components of a machine translation system. Conventionally, in a bilingual transfer system, the transfer dictionaries will whenever possible specify correspondences between actual words of the source and target languages, as is done in our system. (This will be discussed and illustrated below.) However some interesting points of principle are raised when a system either handles more than two languages or is interlingual in design (the two criteria are of course orthogonal).

It is sometimes suggested, or assumed, that the appropriate base for a machine translation system, perhaps especially an interlingual system, should be language-independent not just in the sense of 'independent of any particular language(s)' but also 'independent of language in general', and 'knowledge-based' translation systems using Schank's 'conceptual dependency' framework (eg Schank & Abelson 1977) are presented in, for example, Nirenberg (1986). We believe this approach to be misguided. The task of translation is specifically linguistic: the objects which are represented and compared, analysed and generated are texts, linguistic objects. The formal representations built and manipulated in formalized translation should therefore, to be appropriate to the task, also be specifically linguistic (cf Johnson 1985).

As well as this issue of principle, there are purely practical arguments against the use even of non-language-specific, let alone non-linguistic representations in machine translation. An interlingual system must (aim to) hold in its 'dictionaries', and/or in the knowledge representation component which supplements or supplants them, any and all information which could in principle ever be needed for translation to or from any language, while the information in a transfer system will be decided on a need-to-know basis given the specific languages involved. Thus for a transfer system the amount of dictionary information needed will be smaller, and the problem of selecting what to include will be more easily and objectively decidable, than for an interlingual system. On this interpretation, it is possible in principle, although complex in practice, to construct a single unified lexicon of mappings among three or more languages which would still properly be classed as a transfer dictionary; and this task would still be simpler than the construction of a satisfactory interlingual 'lexicon'.

Should one take·the further step to a fully non-linguistic inter-'lingua', the complications will ramify yet further. It will be necessary to construct not only a fully adequate and genuinely neutral knowledge-base, but also lexically driven access to it, presumably through a more-or-less conventional lexicon, for each language in question, in a way which enables this language-neutral core accurately to map specific lexical equivalents across particular languages.

This is not to deny that a complex and sophisticated semantics is necessary, and some recourse to world-knowledge would be helpful, for the resolution of ambiguities and the determination of correct translation equivalents. We reject only the claim that an appropriate or realistic level of underlying representation for machine translation can be either non-linguistic or language-universal, let alone both at once.

The dictionaries and the user

Given these three underlying design principles - dictionary entries for lexical stems of specified category, strictly monolingual analysis and generation dictionaries, and transfer dictionaries based on language-pair-specific information - we have tried to organize our dictionary system to offer efficient and perspicuous access to both the end-user and the system itself. We have implemented on-line dictionary creation routines for our intended monolingual end user, which elicit and encode the values for a range of features for an open class English word (noun, verb, or adjective - see Whitelock et al 1986 for details), but which do not ask for translation equivalents in Japanese. This information is sufficient for a parse to continue, with the word in question retained in English or transcribed in katakana in the output (as happens also for proper nouns).

The English entries thus created are stored within the dictionary system in separate '.supp' files, where they are accessible to the parser, (thus allowing translation to continue) but clearly isolated for later full update. This will be carried out by the bilingual linguist, who will add an index to the transfer dictionary and create corresponding full entries in the transfer and Japanese dictionaries. At present, during system development, these stages are often run together. In the final version of the system, for monolingual use, the bilingual updates will be supplied by specialist support personnel.

Although this might appear restrictive, it is less so than the alternatives. Given our objective of offering reliable Japanese output to a monolingual English user, we cannot expect that user to carry out full bilingual dictionary update. Equally, we do not wish to constrain the user to operate within the necessarily limited vocabulary of the dictionaries supplied with the system. This organization of information goes some way towards overcoming this dilemma, by enabling the user to extend the available working vocabulary without bilingual knowledge.

The dictionaries, the user, and the system

The dictionary creation routines, whether in monolingual mode for the end user or in bilingual mode for the linguist, build 'neutral form' dictionary entries consisting of a simple list of features and values. Regular inflected forms are supplied dynamically during dictionary creation and lookup, by running the morphological analyser in reverse. All atomic feature values are listed explicitly. This ensures that all the information held about each word is clearly available to the user. The compilation process for these neutral forms is so designed that values for a new feature can be added throughout without totally rebuilding the dictionary file in question.

ENTRIES FROM DICTIONARY CREATION

```
nf([word=trees,stem=tree,stemtyp=noun,
    cntype=count,plural=[]]).

nf([word=live,stem=live,stemtyp=verb,
    thirdsing=[],pres_part=[],past=[],
    past_part=[]]).

nf([stem=difficult,stemtyp=adj,adverb=[],
    forms_comp=no]).
```

The neutral form dictionaries are automatically compiled into 'program form' entries in the format expected by the parser. These are kept as small as possible, firstly by storing only irregular inflected forms, as in the neutral form entries described above. Secondly, we factor out predictable atomic feature values into feature co-occurrence restrictions. These derive largely from the fcrs of Generalized Phrase Structure Grammar (Gazdar et al 1984), which are in fact classical redundancy rules as in Chomsky (1965), Chomsky & Halle (1968).

FEATURES

```
featset(daughters,[subj,obj,obj2,
  pcomp,vcomp,acomp,scomp, ....]).
featset(roles,[arg1,arg0,arg2,adjunct,
          compound, .....]).
```

FEATURE CO-OCCURRENCE RESTRICTIONS

```
fcr(inf=_,[fin=nonfin]).
fcr(tense=_,[fin=finite,stemtyp=verb]).
fcr(fin=_,[cat=verb]).

jfcr(noun=yes,[verb=no,adnom=no,
              tensed=no]).
jfcr(adj=yes,[adverb=no,adnom=no,
              tensed=no]).
```

This is one possible implementation of the 'virtual lexicon' strategy proposed by Church 1980, and widely used since. A similar technique is used in the LRC Metal system (Slocum & Bennett 1982). The use of defaults in dictionary design for machine translation, or natural language processing in general, is a complex issue which lies beyond the scope of the present paper.

Thus the maximum load is given to generalized lexical redundancy patterns rather than to individual lexical entries. However this is not 'procedural' as opposed to 'declarative'. It is simply a declarative statement in which the maximum number of regularities are stated explicitly as such.

This two-layered dictionary structure and automatic compilation ensures that any change in the parser which implicates its dictionary format requires at most a recompilation from the neutral form rather than labour-intensive rewriting. It also makes dictionary information available both in a form perspicuous to the human user and, independently, in a form optimally adapted to the design of the parser.

The dictionaries and the system

The program form dictionaries factor out different types of information to be invoked at different stages in parsing and interpretation of English input. In the first stage, grammatical category and morphological and semantic-feature information is looked up in 'edict' dictionaries.

EXAMPLES FROM ENGLISH DICTIONARIES

NOUN

```
edict(file,[pred=file,cntype=count]).

edict(information,[pred=information,
       cntype=mass]).

edict(manual,[pred=manual_book,cntype=count]).

edict(storage,[pred=storage,cntype=mass]).
```

VERB

```
edict(consist,[pred=consist,stemtyp=verb]).

edict(correspond,[pred=correspond,stemtyp=verb]

edict(provide,[pred=provide,stemtyp=verb]).

edict(put,[pred=put,stemtyp=verb]).
irreg(put,[pred=put,tense=past]).
irreg(put,[pred=put,nfform=en]).

edict(be,[pred=be,block=[1,1,1,0,1,1,1|_]]).
irreg(are,[pred=be,tense=pres,subj/agrpl=yes]).
irreg(been,[pred=be,nfform=en]).
irreg(is,[pred=be,tense=pres,subj/agrpl=no]).
irreg(was,[pred=be,tense=past,subj/agrpl=no]).
irreg(were,[pred=be,tense=past,subj/agrpl=yes])

edict(become,[pred=become,stemtyp=verb]).
irreg(became,[pred=become,tense=past]).
irreg(became,[pred=become,nfform=en]).
```

ADJ

```
edict(graphical,[pred=graphical,stemtyp=adj])

edict(manual,[pred=manual_hand,stemtyp=adj]).
```

DET

```
stop(the,det,[spec=def]).

stop(a,det,[spec=indef,agrpl=no,artpl=no]).

stop(many,det,[quan=many,agrpl=yes]).

stop(much,det,[quan=much,agrpl=no]).

stop(some,det,[spec=indef,artpl=yes]).
```

This information is used in parsing to produce LFG-ish functional structures. Optional and obligatory subcategorization features are then looked up in separate 'subcat' dictionaries.

EXAMPLES FROM SUBCAT
 - PROVIDING A SUBCATEGORIZATION FRAME

```
subcat(consist,[intrans,ofarg,loc]).
oblig(consist,[arg1]).

subcat(correspond,[intrans,toarg,loc]).

subcat(provide,[trans,forben,loc]).

subcat(put,[trans,locgoal]).
oblig(put,[arg0,arg2]).

subcat(be,[predadj,aux],predadj).
subcat(be,[pass,aux],passive).
subcat(be,[prog,aux],prog).
subcat(be,[exist,objess],be_exist).
subcat(be,[intrans,objess]).

subcat(become,[intrans,objess,loc]).
```

Using this additional information, the functional structures can go through function-argument mapping to produce semantic structures for those which are valid. The transfer component consists solely of a dictionary of mappings between source and target language lexical items, or, where necessary (eg for idioms), more complex quasi-syntactic configurations.

EXAMPLES FROM TRANSFER DICTIONARY

NOUNS

```
xdict(file,fairu).

xdict(information,zyouhou).

xdict(manual_book,manyuaru).

xdict(storage,kiokusouti).
```

VERBS

```
xdict(be_exist,a,[vmorph=aru]).

xdict(become,na,[gloss=become]).

xdict(consist,na,[gloss=consist]).

xdict(provide,sonae).
```

ADJECTIVES

```
xdict(graphical,gurafikku).

xdict(manual_hand,syudou).
```

Japanese generation proceeds through an inverse sequence.

EXAMPLES FROM JAPANESE DICTIONARIES

NOUN

```
jdict(fairu,[pred=fairu,kform=kata,gloss=file,
    stemtyp=noun]).

jdict(jouhou,[pred=jouhou,kform='情報',
    gloss=information,stemtyp=noun]).
```

```
jdict(kiokusouti,[pred=kiokusouti,
    kform='記憶装置',gloss=storage,stemtyp=noun]

jdict(manyuaru,[pred=manyuaru,kform=kata,
    gloss=manual,stemtyp=noun]).

jdict(syudou,[pred=syudou,kform='手動',
    gloss=manual,stemtyp=noun]).

jdict(gurafikku,[pred=gurafikku,kform=kata,
    gloss=graphical,stemtyp=noun]).
```

U-VERB

```
jdict(i,[pred=i,vmorph=1-i,kform=hira,gloss=be,
    stemtyp=uverb]).

jdict(ire,[pred=ire,vmorph=1-e,kform='入れ',
    gloss=put,stemtyp=uverb]).

jdict(na,[pred=na,vmorph=5-r,kform='成',
    gloss=become,stemtyp=uverb]).

jdict(na,[pred=na,vmorph=5-r,kform='成',
    gloss=consist,stemtyp=uverb]).

jdict(sonae,[pred=sonae,vmorph=1-e,kform='備',
gloss=provide,stemtyp=uverb,tensem=punct]).
```

Conclusions

The organization of the dictionaries in a machine translation system raises a number of significant issues, some general to natural language processing and others specific to translation. In the course of implementing our English-Japanese system, we have arrived at one possible set of answers to these questions, which we hope to have shown are both computationally practicable and of wider theoretical interest.

ACKNOWLEDGEMENTS

The work on which this paper is based is supported by International Computers Limited (ICL) and by the UK Science and Engineering Research Council under the Alvey programme for research in Intelligent Knowledge Based Systems. We are indebted to our present and former colleagues for their tolerance and support, especially to Pete Whitelock and Rod Johnson.

REFERENCES

Ades, Antony, & Mark Steedman. 1982. On the Order of Words. **Linguistics and Philosophy.**

Bresnan, Joan, ed. 1982. **The Mental Representation of Grammatical Relations.** MIT Press, Cambridge, Mass.

Chomsky, Noam. 1965. **Aspects of the Theory of Syntax.** MIT Press, Cambridge, Mass.

Chomsky, Noam, & Morris Halle. 1968. **The Sound Pattern of English.** Harper & Row, New York.

Church, Kenneth. 1980. **On Memory Limitations in Natural Language Processing.** MIT Report MIT/LCS/TR-245.

Gazdar, Gerald, Ewan Klein, Geoff Pullum, & Ivan Sag. 1984. **Generalized Phrase Structure Grammar.** Blackwells, Oxford.

Johnson, R. L. 1985. Translation. In Whitelock et al, eds.

Knowles, Francis. 1982. The Pivotal Role of the Dictionaries in a Machine Translation System. In Lawson, Veronica, ed. **Practical Experience of Machine Translation.** North-Holland.

Nirenberg, Sergei. 1986. **Machine Translation.** Tutorial Introduction, ACL 1986, New York.

Ritchie, Graeme. 1985. The Lexicon. In Whitelock et al, eds.

Schank, Roger, & Robert Abelson. 1977. **Scripts, Plans, Goals and Understanding.** Erlbaum.

Slocum, Jonathan, and W. S. Bennett. 1982. The LRC Machine Translation System. Working Paper LRC-82-1, LRC, University of Texas, Austin.

Steedman, Mark. 1985. Dependency and Coordination in the Grammar of Dutch and English. **Language.**

Whitelock, Peter. 1986. A Categorial-like Morpho-syntax for Japanese.

Whitelock, Peter, Mary McGee Wood, Brian Chandler, Natsuko Holden, & Heather Horsfall. 1986. Strategies for Interactive Machine Translation. **Proceedings of Coling86.**

Whitelock, Peter, Mary McGee Wood, Harold Somers, R. L. Johnson, & Paul Bennett, eds. Forthcoming. **Linguistic Theory and Computer Applications.** Academic Press, London.