[Proceedings of a Workshop on Machine Translation, July 1990, UMIST]

AUTOMATED PRODUCTION OF DOMAIN-SPECIFIC MACHINE TRANSLATION SYSTEMS

J.D. Phillips, Centre for Computational Linguistics, U.M.I.S.T.

A technique is suggested of extending a machine translation system capable of handling the most basic areas of the source and target language, to make it capable of translating texts in a particular restricted domain or sublanguage. The technique assumes a pre-existing dictionary of technical terms in the domain which can be added to a core grammar. Remaining translation ambiguities are solved by applying the statistical method of cluster analysis to partial semantic representations produced by the system from monolingual texts, to deduce a type hierarchy for the terms of the domain, and thence generalise collocational restrictions for the lexical items.

The architecture of an automatically produced domain-specific machine translation system can be represented as a *translation engine*, using *linguistic knowledge* produced automatically by a *linguistic knowledge generator**. 'Linguistic knowledge' is here to be interpreted in the broadest sense. This paper is concerned with the linguistic knowledge generator, with how domain-specific linguistic knowledge can be discovered. It therefore largely ignores the translation engine, and presents only a brief description of the format of the linguistic knowledge. Basic grammars of the source and target languages are assumed to exist already, as is a bilingual dictionary of domain-specific terms. The methodology for their construction is outside the scope of this paper – they are in fact more general problems of natural language processing and translation respectively. Since the work has been done in the context of an English and Japanese machine translation project, examples will be given in these two languages.

Linguistic Formalism

The formalism used here for exemplification is a very simplified categorial grammar. For a more adequate version see Calder et al. (1988).

A grammar consists of just one component, the lexicon, which contains the syntactic, semantic, and pragmatic information about each word treated. The simple example below shows an analysis of the sentence 'John saw Mary':

Word	*Syntax*	*Semantics*
john	: n#J	: john(J)
mary	: n#M	: mary(M)
saw	: s#S \ n#Subj / n#Obj	: see(S)&past(S)&subject(S,Subj)&object(S,Obj)

saw mary	: s#S \ n#Subject
	: see(S)&past(S)&subject(S,Subject)&object(S,M)&mary(M)
john saw mary	: s#S
	: see(S)&past(S)&subject(S,J)&object(S,M)&mary(M)&john(J)

The Prolog convention that variables begin with capital letters is used. Entries in the lexicon are of the form: *Word:Part-of-Speech:Semantics*, where the part-of-speech is either atomic, as in the entries for

*These terms, and many other ideas in this paper, emerged during discussion with Kenji Yoshimura, Jeremy Carroll, and Jun-ichi Tsujii.

'John' and 'Mary', or complex as in the entry for 'saw'. An atomic part-of-speech is of the form *Category#Index*, where the category is one of a small set of basic categories (here just *s* and *n*), and the index can be imagined as a pointer to an entity in the discourse, e.g. an object or an event. So the entry for 'John' says that the string j-o-h-n is of category *n*, and that it is a property of the entity referred to by the word (the variable *J*) that it has the property *john* (i.e. it is named 'John').

The word 'saw' is defined as a string that would be of category *s* if it combined with a string of category *n* immediately to its right, and another string of category *n* immediately to its left. Variable instantiation, using the indices, ensures that the entity represented by the first string is marked as the object of the event of seeing, and the entity represented by the second string as its subject. In a real grammar of English there would of course be several different definitions of 'saw' representing its uses as a noun, an infinitive, etc.

Combining 'saw' with 'Mary' gives a string which would be of category *s* if it combined with a string of category *n* to its left. The semantics are the conjunction of the semantics of the parts. It follows that combining 'John' (of category *n*) with 'saw Mary' gives a string of category *s*. The semantics are again the conjunction of the semantics of the parts. Note that the instantiation of the variable indices has ensured that the semantic relations are all correctly stated in the logical form.

A similar example can be given for Japanese. The sentence *John ga tegami wo kakimasita* means 'John wrote a letter', as does *Tegami wo John ga kakimasita*. *Ga* and *Wo* are nominative and accusative particles respectively. Either the subject or the object can be omitted.

Word	*Syntax*	*Semantics*
john	: n#J	: john(J)
tegami	: n#M	: letter(M)
kakimasita	: s#E	: write(E)&past(E)
ga	: s#E / s#E \ n#Subject	: subject(E,Subject)
wo	: s#E / s#E \ n#Object	: object(E,Object)

john ga : s#E / s#E : subject(E,J)&john(J)

tegami wo : s#E / s#E : object(E,M)&letter(M)

tegami wo kakimasita : s#E : write(E)&past(E)&object(E,M)&letter(M)

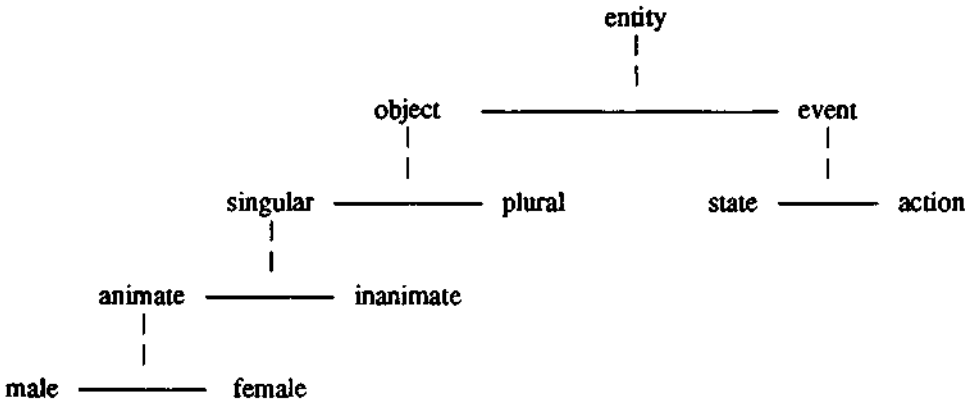
john ga tegami wo kakimasita : s#E : write(E)&past(E)&object(E,M)&letter(M)&subject(E,J)&john(J)

The advantages of this formalism include firstly its conceptual simplicity. Secondly, the grammar is declarative and can be used both for analysis and generation, both of which are very simple. Thirdly, since there is a lexical entry for each type of use of each word, it is simple to attach other information to each entry. Specifically, for this project, it may be useful to attach statistical information to each entry, and see the note on 'Precedence' below. Fourthly, the part of the grammar relating to the particular domain is simply characterised as a collection of lexical entries. A frequently mentioned disadvantage is that categorial grammar is highly redundant, expressing what should be generalisations over and over again on different lexical entries — but templates or macroes can easily be used to capture generalisations.

Co-reference: For translation it is necessary to resolve anaphoric reference, by pronouns and definite articles in English for instance. The idea of a *context* is therefore introduced. The context contains a list of all the indices introduced by the sentences of the text being translated. The context is built up gradually as analysis progresses through the text, so that a sentence acts as a function from one context to another. (The context can contain other information as well – the current topic, reference times, and background knowledge, for instance.) Analysis of a sentence both can refer to and change the context. In terms of the categorial syntax above, the semantics of certain words, such as pronouns, are defined

by reference to the context, while other words, such as indefinite articles and main verbs, change the context by introducing new potential referents.

To assist in determining the correct referent of an anaphor, indices are typed, that is, they have a place in a hierarchy of types such as that below.



Precedence: Entries in the categorial lexicon are marked with a *precedence level*. Only the highest level entries for a word are used in parsing, unless the parsing would otherwise fail. This allows, for example, for the minimal attachment strategy to be built into the definitions of prepositions. In fact there is evidence that different prepositions have different preferred attachments, so that in practice some prepositions would have their definitions as sentence modifiers given higher precedence, others their definitions as noun modifiers. In the same way, some verbs have preferred subcategorisation frames (Mitchell & Holmes, 1985), which could be similarly encoded.

Collocational Restrictions: The Japanese word *kakimasita* has several possible translations into English. In the sentence *John ga tegami wo kakimasita* it means 'wrote', but in other contexts it can be translated 'spelt', 'drew' or 'scratched' (an itch). One simple way of getting the right translation on most occasions is to extend the type hierarchy above to include all concepts in the domain. The definitions of the English translations of *kakimasita* can then be given as

Word	*Syntax*
wrote	: s#S \ n#Human / n#Literature
	: write(S)&past(S)&subject(S,Human)&object(S,Literature)
spelt	: s#S \ n#Human / n#Word
	: spell(S)&past(S)&subject(S,Human)&object(S,Literature)
drew	: s#S \ n#Human / n#Picture
	: draw(S)&past(S)&subject(S,Human)&object(S,Picture)
scratched	: s#S \ n#Animate1 / n#Animate2
	: scratch(S)&past(S)&subject(S,Animate1)&object(S,Animate2)

Collocational restrictions such as these can determine many structural ambiguities during parsing, and can determine the correct lexical choice during generation. They are of limited use in normal natural language work, since they encode typical uses of words and words are often used in untypical ways. However in formally written sublanguage texts, metaphor and 'creative writing' play little part; content words — the technical terms of the sublanguage — are normally used in their literal senses, and collocational restrictions between them can be expected to be reasonably reliable.

The Linguistic Knowledge Generator

The linguistic knowledge generator is a procedure which produces the linguistic knowledge for a particular sublanguage. It takes as inputs: core grammars for both languages, a bilingual dictionary of sublanguage terms, and sample monolingual texts in both languages. It also has a correction facility which can search for indeterminacy in the linguistic knowledge and allow a human expert to make corrections using a graphical structure editor.

The core grammars: The core grammars contain the basic grammars of the source and target languages. With a categorial grammar this means including all function words, other closed class words, and morphology in the lexicon, along with templates for the syntax of the major parts of speech. The core grammars would thus treat the translation of such difficult areas as pronouns and other anaphoric elements, indexicals, tense and aspect, comparatives, plurality, and genericity. The basic idea here is that all the linguistically hard parts of translation are dealt with in the core grammar. Material which is not in the core grammar can only be translated term-for-term (not necessarily word-for-word). This seems reasonable for the translation of formally written texts in a restricted domain, where it can be assumed that the vocabulary will map in a direct way onto a well-structured conceptual field which will be the same in both languages.

Extension to a sublanguage, stage 1, the dictionary: It is assumed that a machine-readable dictionary will be available, giving translation equivalents and broad syntactic categories for the technical terms of the sublanguage.

For each translation pair, a new semantic property primitive is created, and entries are added to the core grammars for the words with appropriate syntactic definitions and with semantic representations using the new semantic primitive.

Extension to a sublanguage, stage 2, collocational restrictions: The grammars created in stage 1 are used to analyse monolingual sublanguage texts. Parsing the text should succeed, but there will be many possible analyses, with a large amount of structural ambiguity. Statistical techniques are brought to bear on the information obtained from this analysis to compute a hierarchy of types and represent collocational restrictions in the grammar. This stage is done separately for the two languages as follows.

Cluster Analysis

Cluster analysis is a statistical technique for grouping items according to the similarity or otherwise of their properties. Aldenderfer & Blashfield (1985) give an easily-comprehensible overview of the technique. Cluster analysis can be used to group the index types of the semantic representation described above according to the contexts in which they occur. By recursively grouping similar items together in a hierarchy, and basing the collocational restrictions on this generalisation rather than directly on the data in the text, some of the inadequacy of the data, due to their comparatively small size, can be overcome.

Some experiments on cluster analysis of syntactically annotated sublanguage texts were conducted in the 1970's (Hirschman et al. 1975). The cluster analysis was used then as an aid in the manual construction of a sublanguage grammar. Though the technique was successful, the necessity for manual intervention in the clustering and the lack of a principled way of incorporating its results into the grammar made it less useful than it might have been. The present work attempts to make the whole process automatic.

Contexts and objects for clustering: As the basic descriptors of objects and events respectively, the head index of the root of each domain-specific noun and verb is given a unique type. The items to be clustered are then these newly-created index types. The contexts used to cluster them are their

properties and their semantic relations to other index-types.

Of course, the relevant contexts for clustering are more complex than the single level of relations so far discussed. In the sentence *John looked at the moon through a telescope*, the moon belongs to a class of things which can be looked at through a telescope, but the relevant context for *moon* here is something like *object-of-subject-of-mode-whose-object-is-telescope*. In general, any context of any complexity might be relevant to the clustering, but it is obviously impracticable to compute all such contexts and use them in the clustering. An ideal method would use a clustering algorithm which did take all possible complex contexts into account and dynamically discarded those which had no effect on the clusters (i.e. the great majority). This may be possible using an algorithm similiar to that used to calculate variable-length n-grams but it has not yet been investigated. The approach used at present is to try and decide in advance, in the core grammar, which complex contexts will be relevant, and list them there. This does not solve the whole problem, only some of the easier parts of it. The listed complex contexts correspond to some extent to the implicatures of particular linguistic forms.

Calculating the distances between objects: There are two steps to the simple clustering algorithm used here. First the 'distance' between each pair of items is calculated as a measure of how similar they are, then items 'close' to each other are merged.

Having decided on a set of items to be clustered and a set of relevant contexts, for each item to be clustered, a list is made of the number of times it occurs in each context. The point of the clustering algorithm is then to group similar lists, and hence similar items, together. The Manhattan metric is a measure of the similarity of any two lists. For two lists i and j, the distance between them, d_{ij}, is given by $d_{ij}=\sum_c |p_{ic}-p_{jc}|$ where p_{ic} is the proportion of the occurrences of item i which fall in context c. This metric is particularly quick to calculate for these data since contexts in which neither item occurs (the great majority) have no effect and can be ignored.

A matrix is constructed using this metric to show the distances between every pair of items.

Finding the clusters: For a given distance, starting with one item per cluster, merge two clusters if a pair of items exist, one from each cluster, the distance between which is less than the given distance. Continue doing this until no more clusters can be merged, to give the clusters separated by at least the given distance. If the given distance is increased gradually, clusters at all levels will be shown.

There are many other, more sophisticated, clustering methods which might give better results than the one just described (single link with Manhattan distance). Their results are however more difficult to interpret and require considerable computer time to obtain. The algorithm used is convenient for the initial experimental work reported here.

A Simple example

This simple example was produced by a small prototype implementation. There is some cheating in that the lexicon contains only the noun-modifier definition of 'with'.

Text: John had a book with a red cover. He read it and liked it. Mary borrowed it and read it too, but she didn't like it. Mary had an encyclopaedia. She read some of it. It had a green cover.

Core vocabulary: a, and, but, didn't, he, it, John, Mary, of, she, some, too, with (n#H\n#H/n#M : have(E)&subject(E,H)&object(E,M)). 'John' and 'he' have indices of type *male*, 'Mary' and 'she' have ones of type *female*. This is necessary to get the anaphora right.

New vocabulary: book borrowed cover encyclopaedia green liked read red

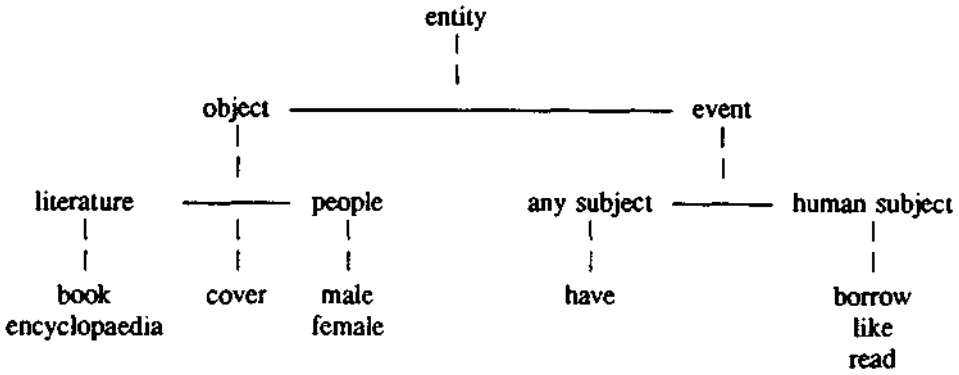
Logical form showing only predicates and index types.
[cover(cover), green(cover), have(have), subject(have,encyclopaedia), object(have,cover), some(encyclopaedia), read(read), subject(read,female), object(read,encyclopaedia), encyclopaedia(encyclopaedia), have(have), subject(have,female), object(have,encyclopaedia),

mary(female), like(like), object(like,book), not(like), subject(like,female), conj(e,e), conj(e,like), too(e), read(read), subject(read,female), object(read,book), conj(e,borrow), conj(e,read), borrow(borrow), subject(borrow,female), object(borrow,book), mary(female), like(like), subject(like,male), object(like,book), conj(e,read), conj(e,like), read(read), subject(read,male), object(read,book), cover(cover), red(cover), have(have), subject(have,book), object(have,cover), book(book), have(have), subject(have,male), object(have,book), john(male)].

Clusters — Manhattan distance, single link, on entity types. The first three columns show the distance level at which the cluster was merged with another, the distance level at which the cluster was formed, and the difference between the two numbers. A small number in this third column (e.g. 0.06 {like read}) shows the cluster to be insignificant.

End	*Start*	*Span*	*Cluster*
2.00	1.60	0.40	{ book cover encyclopaedia female male }
2.00	1.33	0.67	{ borrow have like read }
1.33	1.11	0.22	{ borrow like read }
1.60	1.10	0.50	{ book encyclopaedia }
1.11	1.05	0.06	{ like read }
1.60	0.93	0.67	{ female male }

Type hierarchy



Vocabulary with modified types:

book	: n#Book
borrow	: s#Action \ n#People / n#Literature
cover	: n#Cover
encyclopaedia	: n#Encyclopaedia
green	: n#Cover / n#Cover
have	: s#Action \ n#Object1 / n#Object2
John	: n#Male
like	: s#Action \ n#People / n#Literature
Mary	: n#Female
read	: s#Action \ n#People / n#Literature
red	: n#Cover / n#Cover
with	: n#Object1 \ n#Object1 / n#Object2

Future Work

Applying this methodology to larger, more realistic texts exposes some serious weaknesses. Though the clusters produced from such texts (e.g. a 400-word text of bakery recipes) are mostly intuitively acceptable and fall into a definite hierarchy just as in the example above, a few of the clusters are nonsensical and a few of the index types are not clustered with any other types though they intuitively should be. These problems seem to be due to the nature of the data and may be amenable to correction by statistical methods.

Linguistic data are inherently different to the data clustering algorithms are designed to handle. The clustering algorithm assumes that the data it is dealing with is correct, as far as it goes. This is the case in other applications, where the data might be the physical characteristics of animals, or the results of tests on medical patients. The purpose of the cluster analysis is to find natural classifications based on these definite data. The data dealt with here though are no more than a tiny sample of language. The non-appearance of a word in particular context does not mean that it cannot occur in that context, only that it does not in the text analysed. The cluster analysis is being used here to group similar words together and thereby gloss over possible gaps in the coverage, that is, it is being used to generalise from unreliable data. This is all very well when the gaps in the coverage are small, but this is not always the case in practice. About half the words in a typical text occur just once. For instance this paper has a vocabulary of 954 words of which 508 occur just once. The proportion decreases only very gradually with the length of the text. That a word only occurs once is not a bar to obtaining useful data from it, since the discourse entity of which that one occurrence is a property usually enters into many semantic relationships. Arguments, governors, modifiers, and subsequent anaphoric reference can together give a rich context to an an index type introduced by a single occurrence of a word.

Nevertheless, there are many cases where the poverty of the context is such as to make clusters based on it unreliable. Practical tests show that such items can sometimes create ill-founded clusters by appearing to be close to other entity types, or can be left outside the clustering altogether by appearing distant from all other entity types. This type of error due to a small sample is common in statistical analyses, and here are statistical methods for dealing with it: the following modification to the single-link clustering algorithm could be implemented.

The sampling error in a tally of n can be estimated to be $\sqrt{n}$ – for justification see Foster et al. (1990, chapter 3). Therefore if a word occurs n times in some context, we can estimate that the likely range of occurrences is between $n_{max}=n+\sqrt{n}$ and $n_{min}=n-\sqrt{n}$. Hence we can estimate the maximum likely Manhattan distance between two items as

$$d_{ij}=\sum_c\left(\left|\frac{n_{ic}}{t_i}-\frac{n_{jc}}{t_j}\right|+\frac{\sqrt{n_{ic}}}{t_i}+\frac{\sqrt{n_{jc}}}{t_j}\right)$$

where n_{ic} is the number of occurrences of item i in context c and t_i is the total number of occurrences of item i. Running the single-link clustering algorithm with these maximum distances should give reasonably reliable clusters, but leave some rarely occurring items outside any clusters. Clustering could be continued, using a stricter clustering algorithm (e.g. complete-link) with the minimum distances, to incorporate these rarely occurring items into clusters where possible.

There are many other problems in the design of the system which have not been addressed here. The most obvious is in the construction of an adequate core grammar, but other less obvious problems might be more important in practice. Some types of text contain syntactic constructions not found in ordinary English – the omission of articles in newspaper headlines for instance. However, studies of the syntax of sublanguages (e.g. Kittredge, 1982) suggest that syntactic 'aberrations' from the standard language are confined to ellision of articles, object noun phrases, and the copula. These ellisions are common to very-restricted-domain writing as a genre rather than to particular subject domains, though they are more prevalent in certain domains. The phenomena could perhaps be handled by incorporating all such syntax in the core grammar (by having an empty article, pronoun, and copula, in the formalism sketched above), and allowing the linguistic knowledge generator to adjust the precedence of lexical entries according to what is found in the sample texts. It may be though that the

extra ambiguity thus introduced into the analysis of that text would make the procedure unworkable without manual disambiguation. On the other hand the same studies suggest that some difficult syntactic phenomena, pronominalisation and tense in particular, are simpler in restricted sublanguages than elsewhere. Pronouns for instance are rarely or never used in many sublanguages.

Conclusion

There are two conclusions to be drawn from this work. Firstly, cluster analysis of logical forms seems a promising avenue to explore in sublanguage machine translation work. Secondly, a system for the automatic production of sublanguage machine translation systems does not seem a total impossibility.

References

Mark S. Aldenderfer & Roger K. Blashfield: *Cluster Analysis*. Beverley Hills:Sage, 1984.

John C. Foster et al. (editors): *Language Modelling for Automatic Speech Recognition*. MIT Press (forthcoming).

L. Hirschman, R. Grishman & N. Sager: 'Grammatically-based automatic word class formation', in *Information Processing & Retrieval* 11 (1975), pp. 39-57.

R. Kittredge: 'Variation and homogeneity of sublanguages', in *Sublanguage*, ed. R. Kittredge & J. Lehrberger, Berlin:de Gruyter, 1982.

D.C. Mitchell & V.M. Holmes: 'The role of Specific Information about the verb in parsing sentences with local structural ambiguity', in the *Journal of Memory and Language*, 25 (1985).

J. Calder, E. Klein & H. Zeevat: 'Unification Categorial Grammar: a concise, extendable grammar for natural language processing', in the *Proceedings of Coling*, Budapest (1988).