Superlinks:

A new approach to constraining transfer in machine translation

Lars Ahrenberg

Department of Computer and Information Science, Linköping University
SE – 581 83 Linköping, Sweden
lah@ida.liu.se

Abstract

This paper outlines a corpus-based method for machine translation based on link classification. The transfer dictionary is obtained by word alignment, where the output from an automatic alignment system is analyzed and corrected manually. The word links are classified by means of superlinks that express inherent and contextual properties of the participating words. We illustrate how such superlinks can be used to constrain transfer for two types of constructions where English and Swedish differ, simple main clause types and inflection within the noun phrase.

1. Introduction

Computational linguistics of today has a great variety of methods and techniques at its disposal. There is a definite trend, however, towards empiricism (cf. Church&Mercer, 1993) and the use of models that rely on large-scale lexicons for the bulk of processing.

These trends have been clearly visible also in the area of machine translation. Different methods exploring translation data have been proposed, notably statistical machine translation (Brown et al., 1990; 1993; Knight, 1997; Och & Weber, 1998; Och et al., 1999) and example-based machine translation (Nagao, 1984; Sato & Nagao, 1990; Sumita & Iida, 1991; Kitano, 1993; Brown, 1996; 1999). What these approaches have in common is the use of a large corpus of translation data. They differ in that statistical MT use the data to train a stochastic model while example-based MT employ some form of pattern-matching methods to find correspondences between (parts of) a given input sentence and the sentences or templates stored in the example-base.

There are also lexicalist approaches to machine translation. Whitelock's Shake-and-Bake approach (Whitelock, 1994) eliminates syntactic transfer rules in favour of lexical transfer. The first phase identifies the lexical units of the input sentence with their dependencies, treating them afterwards as a bag rather than as a tree. The second phase finds appropriate translations at the lexical level given the dependencies, and the last phase creates a sentence in the target language from the translations using a grammar for the target language. Thus, the translation knowledge of the system wholly resides in the transfer dictionary, while knowledge of grammatical structure is encoded only monolingually.

Lexicalist MT has certain similarities with example-based MT; in particular, both approaches eliminate syntactic transfer. This has been pointed out by Turcato et al., (1999) who also propose that the models are sufficiently similar to be unified.

Another development, related to lexicalism in that it focuses on words, is localization of syntactic processing to the word level. The idea is that complex syntactic information can be coded as tags associated with words. This idea has been used by Karlsson et al., (1994) within the Constraint Grammar Framework, and has recently been applied to parsing of Lexicalized Tree-Adjoining Grammars (LTAGs) (Bangalore&Joshi, 1999). Bangalore and Joshi use the term 'supertag' for a tag encoding complex syntactic information and 'supertagging' for the process of disambiguating words in context with respect to supertags.

The approach to MT that will be presented in the following is empirical, lexicalist and radically "localist". Representation and processing take place at the word level. The primary use of the translation data is for generating a translation lexicon based on word alignments. However, words (and alignments) are also associated with tags that encode contextual properties of the alignment. In this way the idea of supertags encoding complex information is generalized from words in monolingual texts to word alignments in parallel texts.

As with most other approaches to machine translation this approach works sentence by sentence. Thus, there is no claim that the approach can provide translations with the same quality as those produced by human translators who regularly change the sentence structure and are able to take the larger discourse context into account. Also, as the linguistic resources of the model are created on the basis of a given parallel corpus, the results will depend on the representativeness of this data for a given input text.

The paper is structured as follows. The next section presents a short overview of the word-based framework to machine translation that I assume. In Section 3 the notion of superlink is defined and in Section 4 the translation relation that goes with it. Sections 5 and 6 give two different concrete illustrations of how superlinks can be applied to translation. Section 7, finally, discusses pros and cons of the approach and suggests directions for future research.

2. Phases of localized translation

Given that processing is restricted to the word level, the translation problem can be divided into the following sub-problems:

1. **Tokenization:** Identify the sentences and the word tokens.
2. **Transfer:** For a given sentence, translate the tokens one by one.
3. **Transposition:** Order the translations into a proper sentence.

2.1. Tokenization

It would appear that the tokenization phase is definitely the simplest one. However, it requires a computable definition of sentences and word tokens. For the moment I will assume that these definitions can be based on character patterns and rewrite rules. Applying these patterns and rules will result in a text being converted into a string of sentences, each sentence in turn consisting of a string of tokens. In this process what was given as a sequence of text words may be converted into a single token. Conversely, a single text word might be split into a sequence of tokens, and a text word containing characters that are used as separators in later processing may be converted into a string without such characters. Some examples (where the underscore is assumed not to be a separator):

New York Times --> New_York_Times
didn't --> did not
e.g. --> eg

2.2. Transfer

One of the hardest problems in machine translation is to choose contextually appropriate translations of words. There are many causes of this problem: words are often inherently ambiguous in the source language, the target language may have no single equivalent or require grammatical marking that is not required in the source language. Sometimes a word should not be translated at all. Consider the following data:

a. She will win	*Hon kommer att vinna*
b. She did win	*Hon vann faktiskt*
c. She did not win	*Hon vann inte*
d. Did she win	*Vann hon*
e. Yes, she did	*Ja, det gjorde hon*

The Swedish translations are not the only possible ones, but they serve to illustrate the problem of word choice in transfer. We find two words, *did* and *win*, with alternative translations. By considering which words in the source correspond to which words in the target, we may come up with the following list of translations for the two words:

win : vinna (a), vann (b,c,d)
did : faktiskt (b), 0 (c,d), det gjorde (e)

While alternative analyses are possible, it is clearly necessary to assign different translations to these words in the different sentences. It might be objected that *win* is translated by the same lexical item throughout; what we have should better be described as a difference in form. However, if we are to process words as they are given, i.e., locally, we have to face such choice problems also.

The rest of the words have unique translations if we restrict our universe to the given examples:

she : hon
will : kommer att
not : inte
yes : ja

How, then, can we make the right choice of translation? One way is to appeal to sentence syntax and note that sentence (b) is a negation and that sentence (c) is an interrogative and go on to say that while English employs do-support for these constructions, Swedish does not, instead marking tense on the main verb. This is of course appropriate as a description, but it is not a localised description. The description can be localised to the word level, however, as follows:

win : vann,	if it is preceded by *did* and a subject head
win : vinna,	in other cases
did : faktiskt,	if it is preceded by a subject head and followed by a verb in the infinitive
did : 0,	if it is followed by *not* or a subject head and a verb in the infinitive
did : det gjorde,	if it is preceded by a subject head and the sentence contains no other verb.

Note that the term subject head refers to a token. In the case of *she* the subject head constitutes the subject by itself. In case of a complex subject the subject head may be accompanied by a number of dependants.

Given a list of potential translations for a word we should select the translation(s) for which the contextual conditions are satisfied in the sentence at hand. What we need to accomplish this is (i) a way to express the contextual conditions in terms of word tags, and (ii) an analysis component that is able to provide these tags and disambiguate them. The analysis component can obviously be some kind of tagger, since the system is localized. However, since the categories are to express properties of the alignment, an ordinary part-of-speech tagger will not suffice. What this means in practice will be illustrated below in Sections 5 and 6.

2.3. Transposition

Given that the transfer component can provide us with contextually appropriate translations for the words of a sentence, the role of the transposition component is to arrange the words in a sequence which is grammatical in the target language and functionally equivalent to the source sentence. There are several issues that have to be resolved for this purpose. For example, should the order given by the source sentence be used as a starting point, or should the translations be treated as a bag of words? Another question is what information should be used for ordering. One possibility is that the tags used for the target words express relevant syntactic features of the target language that can be referred to by rules or n-gram language models. Another possibility is that the tags express reordering operations directly. This would move the burden of reordering from the transposition component to the tag sets; it also assumes that there is an initial order to operate on. The only thing that the transposition component would do is to search the sequence of words for reordering operators and apply them.

3. Tags, supertags and superlinks

Tagging usually means automatic part-of-speech tagging, i.e. the assignment of appropriate parts-of-speech to words in running text by a computer program. A pioneering program of this sort was the TAGGIT program (Greene and Rubin, 1971) which used the 77 tags of the Brown tagset to tag the Brown corpus.

It is useful, however, to make a distinction between tagging as a process, whereby words in running texts are assigned categories from a given finite set, and part-of-speech tagging as a special case of tagging, where the categories express lexical and morphological properties of words. In principle, tags could express any property of a word in running text that is found useful for a given purpose. Thus, word sense disambiguation is also a form of tagging.

Taggers are often given a reductionist design. This means that initially a word is assigned a disjunction of all its possible tags (e.g. as given by a lexicon) and then a set of rules are applied that reduce the number of tags as far as possible, preferably to one. These rules are also called disambiguation rules. This design was used in TAGGIT and is an essential feature of the Constraint Grammar Parsers (Karlsson et al., 1994). The constraint grammars, however, do not use simple tags, but rather tag combinations that amount to a specific “reading” of a word.

Tags can also be used as abbreviations for complex structural information associated with words. This use of tagging has been explored as a pre-process to parsing by Bangalore and Joshi (1999):

*“The idea underlying the approach is that the computation of linguistic structure can be localized if lexical items are associated with rich descriptions (**supertags**) that impose complex constraints in a local context.”* (ibid. p 237)

In this case the tags are used as abbreviations for trees that express the syntactic potential of a word within the LTAG formalism. Using structural and statistically induced disambiguation rules for the supertags, the method enables much faster processing, since tree operations are replaced with tag operations. Moreover, a proper parse in the form of a tree can easily be derived from the end result, which motivates the name ‘almost parse’ for the latter.

In the sequel I will use the term ‘supertag’ in the general sense of the quotation above, irrespective of the formal means used for the syntactic descriptions.

As the translation of a word token depends to a large extent on local context, it seems worth-while to investigate whether supertag encoding of local contexts could prove useful also in machine translation systems. In the case of translation, however, there are two local contexts to consider, the source sentence and the target sentence. The relation can be depicted graphically as in Figure 1.

We arrive at the following definition: A **superlink** is a pair of supertags, where the first element of the pair encodes the local context of a source token, and the second element of the pair encodes the local context of its translation.

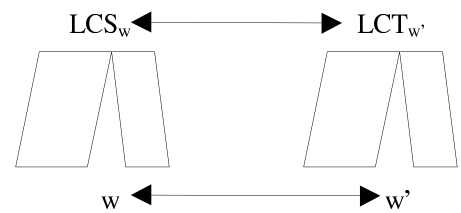

Figure 1. Word alignments (links), local contexts and superlinks

Given a source token, w, that has been translated by a target token, w’, we identify the properties LCS_w and $LCT_{w'}$ of the local contexts of w and w’, respectively, as decisive for the particular translation <w,w’> and encode them as supertags. The corresponding pair of local contexts, <LCS_w, $LCT_{w'}$>, is a **superlink** that categorizes the translation.

4. A translation model based on superlinks

Now, given the notion of a superlink we can give a characterization of the translation relation between the sentences of two languages for a given text type:

A sentence T is a translation of a (well-formed) sentence S iff

- There is a set of token alignments <s_i, t_i> such that S’ = s_1 s_2 … s_n is a proper tokenization of S and T is a well-formed permutation of T’ = t_1 t_2 … t_n, where some of the t_i may be null tokens.
- For every pair <s_i, t_i> there exists a superlink <LCS_i, LCT_i> such that (i) LCS_i is a proper supertag for s_i, and (ii) LCT_i is a proper supertag for t_i.

This definition of the translation relation decomposes it into three relations corresponding to the three phases that were introduced in the previous section. Thus, S is related to S’ via tokenization, S’ to T’ via transfer, and T’ to T via transposition (or permutation). To arrive at a system that actually translates, however, we must find ways to compute these relations, and, more specifically, given a sentence in the source language, compute at least one sentence in the target language that satisfies the translation relation. Apart from the tokenization relation, which was discussed above, the most important knowledge sources required for this task seem to be the following:

- A source dictionary <s-token, s-supertag> associating source tokens with supertags for the source language,
- A set of disambiguation rules for the source language,
- A transfer dictionary <s-token, t-token>,
- A target dictionary <t-token, t-supertag> associating target tokens with supertags for the target language,
- A superlink dictionary <s-supertag, t-supertag> associating supertags from the two languages that may be aligned in translation,
- A set of rules for the target language that filter out impossible combinations of t-supertags,
- A set of rules for the target language that ensures proper word order for sentences of the target language.

Every knowledge source that is referred to as a dictionary in this list can be obtained from a translation corpus, or by generalizing results obtained from such a corpus. The source dictionary is obtained from a tagged translation corpus and disambiguation rules can be obtained in very much the same way as disambiguation rules for part-of-speech tagging, e.g. by training a stochastic model or a Brill tagger, or by formulating constraints in the Constraint Grammar formalism. The transfer dictionary is obtained from (possibly corrected) links generated by a word alignment program. Filtering rules are of the same type as disambiguation rules, but not necessarily identical as it cannot be ensured that the target tokens are properly ordered with respect to the rules of the target language. The superlink dictionary can be obtained in the same way as the transfer dictionary by looking at the tags rather than the words.

The rules concerned with target language word order can take on a variety of forms. For languages that are fairly close in word order properties, such as English and Swedish, it is probably a good strategy to use the word order of the source sentence as a point of departure. To this input a list of explicit reordering rules is applied, where a rule is triggered when a certain tag or tag sequence is found that violates the rules of the target language.

5. An example

In this section we will look at the sentences (a) – (e) again and develop a set of supertags and superlinks that can handle them. The sentences were:

a. She will win	*Hon kommer att vinna*
b. She did win	*Hon vann faktiskt*
c. She did not win	*Hon vann inte*
d. Did she win	*Vann hon*
e. Yes, she did	*Ja, det gjorde hon*

We make a distinction between *inherent tags* that signal monolingual lexical or syntactic properties of a token, and *positional tags* that signal a token's position in relation to surrounding tokens. Inherent tags thus correspond to parts-of-speech such as ADV or ADJ, particular morphological properties such as PAST or INF or grammatical function at the word level such as SUBJ for subject head.

Positional tags are of different kinds. If a token is known always to appear in the first position of a sentence we may use an absolute tag such as FIRST. Most positional tags, however, are relational indicating that a token appears either to the left or to the right of a token carrying another tag. We use the following symbols, where X indicates an arbitrary tag:

LOFX	LeftOfX
LADJX	LeftAdjacentToX
ROFX	RightOfX
RADJX	RightAdjacentToX
NOX	NoXToTheLeftOrRight
NOLX	NoXToTheLeft
NORX	NoXToTheRight

In addition we employ tags that signal constraints or operations that are necessary on the target side. Some examples are:

X	A position name
REPLX	Move to position X
SWAPX	Swap with token at X
AFTERX	Move to position after X
BEFOREX	Move to position before X
ATTACHR	Attach to right neighbor
ATTACHL	Attach to left neighbor

Tags that may be used for the description of examples (a)-(e) are shown in Table 1 for English and in Table 2 for Swedish.

Table 1: Tags for some English tokens.

English Tokens	**EnglishTags**
she	SUBJ
did	DO&PAST&LADJVINF
did	DO&PAST&LADJNEG&LOFINF
did	DO&PAST&LOFSUBJ&LOFINF
did	DO&PAST&NOINF
win	INF&ROFDO
win	INF&ROFAUX
not	NEG
will	AUX&PRES
yes	INDEP1

Table 2: Tags for some Swedish tokens.

Swedish Tokens	**Swedish Tags**
hon	SUBJ
kommer_att	PRES_INFM
vinna	INF
vann	PAST&REPLZERO
vann	PAST&SWAPDOADV
0	ZERO
inte	NEG
faktiskt	DOADV
det_gjorde	[OBJ_PAST]&FIRST
ja	INDEP1

In the simplest cases the translation of a token has the same properties as the source. For instance, a subject head such as *she*, is regularly translated into a subject head, as in all sentences (a)-(e) above. This relation can be captured by a superlink such as <SUBJ, SUBJ>. It should be noted, though, that while we use the same symbols for both languages in this case, they are not identical since they belong to different sets of supertags. The same simple relation holds for the negations and the affirmative interjections *yes*, *ja*.

In other cases, however, we need to appeal to the positional tags. From the examples we can derive the superlinks shown in Table 3.

Table 3: Superlinks instantiated in sentences (a)-(e).

EnglishTags	**Swedish Tags**
SUBJ	SUBJ
NEG	NEG
AUX&PRES	PRES_INFM
INF&ROFAUX	INF
INF&ROFDO	PAST&REPLZERO
INF&ROFDO	PAST&SWAPDOADV
DO&PAST&LADJVINF	DOADV
DO&PAST&LADJNEG& LOFINF	DO
DO&PAST&LOFSUBJ& LOFINF	DO
DO&PAST&NOINF	[OBJ_PAST]&FIRST
INDEP1	INDEP1

Now, given sentence (a) as input the first stage of the transfer process involves retrieving all possible tags for the tokens of this sentence:

She/SUBJ
will/AUX&PRES
win/INF&ROFDO | INF&ROFAUX

Disambiguation rules for English will determine that ROFDO is not satisfied in this case so source language tags will be unique for each token.

In the next stage we retrieve possible translations from the transfer dictionary and possible target language tags by combining the target dictionary (Table 2) and the superlink dictionary (Table 3). For this sentence there is only one option for each token, as follows:

Hon/SUBJ
kommer_att/PRES_INFM
vinna/INF

Note that the other possible translations for *win*, i.e. *vann* with different tags will be filtered out as they require the tag ROFDO on *win*.

As none of the tags on the target side are operational tags, the transposition phase amounts to nothing more than removing the tags and deleting the underscore.

For sentence (b)-(d) there are four different options for *did* and two options for *win*. In each case, however, disambiguation rules can reduce the number of tags to one for all tokens. For instance, sentence (c) will receive the following tags:

She/SUBJ
did/DO&PAST&LADJNEG&LOFINF
not/NEG
win/INF&ROFDO

In the transfer dictionary there are three options for *did*. In the superlink dictionary, however, there is only one tag available for the tag of *did* and only one item in the target language dictionary that is compatible with this tag, 0. For *win* there are two different translations, but only one of them is compatible with the tag INF&ROFDO, namely *vann*. This word, on the other hand, is compatible with two tags, as follows:

Hon/SUBJ
0/ZERO
inte/NEG
vann/PAST&REPLZERO | PAST&SWAPDOADV

There are two operational tags in this analysis, REPLZERO and SWAPDOADV. The former has a contextual condition that is satisfied, i.e., the occurrence of a ZERO tag, while the latter has not. Thus it is removed and we are left with the following analysis after application of the replace operator:

Hon/SUBJ
vann/PAST
inte/NEG

The processing of sentence (d) will proceed in the same way, the only difference being that in this case 0 appears in the first position so that *vann* will appear there after replacement.

For sentence (b) the difference will be that the tag LADJVINF will trigger the selection of the adverb *faktiskt* with tag DOADV as the only possible translation of *did*. Then only the contextual condition for SWAPDOADV, i.e., the occurrence of a DOADV applies and we get the following derivation:

Hon/SUBJ
faktiskt/DOADV
vann/PAST&SWAPDOADV

→

Hon/SUBJ
vann/PAST
faktiskt/DOADV

Finally, for sentence (e) the tag NOINF will be applicable, yielding the following analysis for the sentence:

Yes/INDEP1
she/SUBJ
did/DO&PAST/NOINF

The tag NOINF will then trigger the selection of *det_gjorde* as translation with tag [OBJ_PAST] &FIRST:

Ja/INDEP1
hon/SUBJ
det_gjorde/[OBJ_PAST]&FIRST

Here FIRST is an operational tag, which we interpret in such a way that it should move to the first position of the clause. This seems to give the wrong result, however, as the expression *det_gjorde* should not move to the left of *Ja*. The first position may be defined in such a way as to exclude independent words such as interjections, thus giving the result

Ja/INDEP1
det_gjorde/OBJ_PAST
hon/SUBJ

from which we obtain the final target string by removing the tags and the underscore.

6. Another example

The next example illustrates some phenomena in the translation of noun phrases from English to Swedish. In particular we consider

- The translation of the definite article in various contexts;
- The translation of nominal compounds;
- Choice of appropriate form for adjectives (both attributive and predicative).

The data to be considered is given below.

(f) *The table is clean.*
Bordet är rent.

(g) *The red table is clean.*
Det röda bordet är rent.

(h) *The kitchen table is clean.*
Köksbordet är rent.

(j) *The red chair is clean.*
Den röda stolen är ren.

(k) *The tables are clean.*
Borden är rena.

(l) *The red kitchen chairs are clean.*
De röda köksstolarna är rena.

(m) *A clean table in the kitchen.*
Ett rent bord.i köket

From this and additional data of the same kind we can derive word alignments as follows:

the : 0 | den | det | de
a : en | ett
table : bordet | bord
tables : borden | bord
kitchen : köket | köks | kök[1]
chair : stolen | stol
chairs : stolarna | stolar
is : är
are : är
red : röd | rött | röda
clean : ren | rent | rena
in : i

[1] This means that we handle kitchen table as a compositional compound. An alternative in this case is of course to recognise the compound in the tokenisation phase, which would make the translation of (h) exactly parallel to that of (f).

To make the right choice of translation for these English words, it seems that we have to consider contextual properties such as the following:

- Is the word following the definite article an adjective or a noun?
- Is a noun a head noun or a dependent noun?
- Is a head noun singular or plural?
- Is an adjective attributive or predicative?
- Is the article (if any) definite or indefinite?
- What gender has the Swedish translation of the head noun?

All these properties, except the last one, can be determined on the basis of the English source. The last one, however, cannot be determined until the transfer phase when Swedish tokens have been introduced into the processing.

The details of the tagging phase will be left out as in the previous section. It can be no doubt, however, that the information contained in the source sentences is sufficient to determine the relevant contextual properties as follows:

(f) *The table is clean.*
DDS&LADJN NSDH IS3S JPS

(h) *The red table is clean.*
DDS&LADJJ JASD NSDH IS3S JPS.

(h) *The kitchen table is clean.*
DDS&LADJN NA NSDH IS3S JPS.

(j) *The red chair is clean.*
DDS&LADJJ JASD NSDH IS3S JPS.

(k) *The tables are clean.*
DDP&LADJN NPDH ISP JPP.

(l) *The red kitchen chairs are clean.*
DDP&LADJJ JAPD NA NPDH ISP JPP.

(m) *A clean table in the kitchen.*
DIS JASI NSIH S DDS&LADJN NSDH

Explanations for the tags are given below.

DDS	Determiner, Definite Singular
DDP	Determiner, Definite Plural
DIS	Determiner, Indefinite Singular
IS3S	be – present, 3rd person Singular
ISP	be – present, Plural
JASD	Adjective, Attributive, Singular, Definite
JASI	Adjective, Attributive, Singular, Indefinite
JAPD	Adjective, Attributive, Plural,

	Definite
JAPI	Adjective, Attributive, Plural, Indefinite
JPS	Adjective, Predicative, Singular
JPP	Adjective, Predicative, Plural
LADJJ	LeftAdjacentToAdjective
LADJN	LeftAdjacentToNoun
NSDH	Noun, Singular, Definite, Head
NSIH	Noun, Singular, Indefinte, Head
NPDH	Noun, Plural, Definite, Head
NPIH	Noun, Plural, Indefinite, Head
NA	Noun, Attributive
S	Preposition

It should be observed that the adjectival tags are motivated by the task at hand, i.e. translation into Swedish. Similarly, the marking of English nouns as definite or indefinite has the same motivation.

A similar set of tags is needed for the Swedish words. The main difference is that the Swedish tags for nouns, adjectives and determiners also include a slot for gender. Thus we distinguish NN… from NT…, where NN indicates nouns with n-endings and NT indicates nouns with t-endings. Adjectival forms are also divided into a weak (JW…) and a strong class (JS…).

Table 4: Tags for some Swedish tokens.

Swedish Tokens	**Swedish tags**
0	DDZERO
den	DNDS
det	DTDS
de	DDP
en	DNIS
ett	DTIS
är	COP&PRES
ren	JSNS
rent	JSTS
rena	JSP I JW
röd	JSNS
rött	JSTS
röda	JSP I JW
bord	NTIS I NTIP
bordet	NTDS
borden	NTDP
stol	NNIS
stolen	NNDS
stolarna	NNDP
kök	NTIS I NTIP
köks	NA&ATTACHR
köket	NTDS
i	S

A list of Swedish tags is shown in Table 4 and a list of superlinks in Table 5.

This way of constructing the supertags will ensure that the only dependency that remains to be resolved on the target side is the choice between n-forms and t-forms of articles and adjectives. For this purpose filter rules will be required.

Consider first sentence (f):

(*f*) *The table is clean.*
DDS&LADJN NSDH IS3S JPS

The combined use of transfer dictionary, target dictionary (Table 4) and superlinks (Table 5) yields the following alternatives:

0/DDZERO
bordet/NTDS
är/COP&PRES
ren/JSNS I rent/JSTS

Note that *bord* is not an alternative translation of *table*, since this word has been tagged as definite in the tagging phase.

Table 5: Superlinks instantiated in sentences (f)-(m).

EnglishTags	**Swedish Tags**
DDS&LADJN	0/DDZERO
DDS&LADJJ	DNDS I DTDS
DDP&LADJJ	DDP
DIS	DNIS I DTIS
JASD I JAPD	JW
JPS	JSNS I JSTS
JPP	JSP
NSDH	NNDS I NTDS
NPDH	NNDP I NTDP
NA	NA&ATTACHR
NSIH	NNIS I NTIS
NPIH	NNIP I NTIP
IS3S	COP&PRES
ISP	COP&PRES
S	S

To select the proper form of the predicative adjective we introduce a filter. Filters could be formulated as negative constraints or as selection rules involving positive criteria. Here we use a negative filter to the effect that the sequence "NTDS COP&… JSNS" is disallowed. This filter is obviously not general enough to handle cases where the subject and the predicative are complex phrases, but it can easily be generalized. What is needed is a filter that allows us to recognize the heads of these two phrases. Also, to get the final result we need to delete the zero item.

Sentence (g) is different in that the subject noun phrase contains an adjective. This adjective is initially also given two alternative translations, as is the definite article:

den/DNDS I det/DTDS
röd/JSNS I rött/JSTS
bordet/NTDS
är/COP&PRES
ren/JSNS I rent/JSTS

Again, filter rules can ensure that only one of the eight combinations that these disjunctions allow are judged as well-formed. Simple filters that will do the job for these examples are:

*JSNS NTDS
*DNDS JSTS

Sentence (h) is handled in a similar way to (f) the only difference being that we have a nominal compound:

(h) The kitchen table is clean.
DDS&LADJN NA NSDH IS3S JPS.

0/DDZERO
köks/NA&ATTACHR
bordet/NTDS
är/COP&PRES
ren/JSNS | rent/JSTS

ATTACHR is an operational tag that will make its token attach to the nearest token to the right. After disambiguation of the adjective and deletion of the zero article, the translation is obtained.

Sentence (j) is handled exactly as sentence (g) though different forms of the article and adjectives will be selected due to the different gender of the head noun. With sentence (k) the superlinks will ensure that plural forms are selected. Sentence (l) introduces a complication, since the head noun is not the noun closest to the article and adjectival attribute, but this complication is handled in the tagging phase:

(l) The red kitchen chairs are clean.
DDP&LADJJ JAPD NA NPDH ISP JPP.

de/DDP
röda/JW
köks/NA&ATTACHR
stolarna/NNDP
är/COP&PRES
rena/JSP

Thus, there is no real difference with sentence (h). Finally, sentence (m) illustrates the case of an indefinite noun phrase:

(m) A clean table in the kitchen.
DIS JASI NSIH S DDS&LADJN NSDH

en/DNIS | ett/DTIS
ren/JSNS | rent/JSTS
bord/NTIS
I/S
0/DDZERO
köket/NTDS

But filters can be given for the indefinite case also:

*JSNS NTIS
*DNIS JSTS

7. Discussion

The great advantage of this approach to automatic translation is its simplicity. It is conceptually clear and the bulk of processing is done by efficient and well-understood processes such as tokenization, tagging, table lookup and simple comparisons between atoms or strings. Furthermore, it is empirically based thus ensuring that the categories employed are observationally adequate. Every superlink expresses a relation that actually occurs for the given language pair and text type. Statistical data on relative frequencies for the superlinks may also be derived from the training corpus. Another advantage is that the analysis phase is partial. Only as much information as is required for the task of translation need be recorded in the tags and the superlinks – a complete syntactic analysis is not required.

The negative twist to this argument is that the tags depend on the given language pair; in fact, no phase of the model can be assumed to be purely monolingual. This is basically the price one has to pay for keeping the categories close to the data.

It must of course also be admitted that a few illustrative examples involving some twenty words on the source side do not prove the feasibility of the framework. The only way to test the model is to develop systems for larger and more varied corpora. This model construction could develop along two directions simultaneously. One way to go is to develop systems for specific applications and text types such as weather forecasts or air travel information systems (ATIS). Another line of research is to develop descriptions for grammatical shifts that are known to be hard for machine translation. A case in point is the Head Shift in examples such as German *Er schwimmt gern* corresponding to English *He likes to swim*.

At the level of tokens we can easily identify correspondences such as

Er – He
schwimmt – to swim
gern – likes

The problem is to identify the dependence between the choice of the two latter correspondences. Going in the direction from English to German, we make the translation of to *swim* as *schwimmt* dependent on the translation of *likes* as *gern* by noting that *likes* is a "gern-verb" (in the same vein as *do* is marked as a "zero-verb" in translations to Swedish in the previous examples). Thus, tagging of the source may give us an analysis of the following type:

He/SUBJ
likes/PRES&GERN
to/NIL
swim/INF&ROFGERN&ROFPRES

The corresponding target sequence might be:

Er/SUBJ
gern/GERNADV
0/NIL
schwimmt/PRES&BEFOREGERNADV

After application of the operational tag the wanted translation is derived. For other variants of the same construction such as questions and negations we would need to add some more tags, but the principle should be clear.

It may be asked how this model can capture semantics and world knowledge. Well, basically it doesn't. However, semantic tags can be taken into account, perhaps at a separate level of analysis. As in other models words can be classified in terms of semantic categories and, when they are relational, in terms of the typical categories that they relate. For instance, the word *(be) in* typically signals a relation between an Object and a Container, as in the classical sentences below (Bar-Hillel, 1953).

The pen is in the box.
The box is in the pen.

If semantic categories are included in the source and target language dictionaries, we may have access to information saying that a fenced area is typically a Container, while a writing utensil is not. Conversely, a writing utensil is a good example of an Object, while a fenced area is not.

Similarly we may classify token translations in terms of subject areas as is done in many translation systems. The subject area of a text or paragraph may then be determined on the basis of the tokens that occur in it.

Another important issue is whether it is possible to get away with contextual tags that refer to simple positional relations such as left-of and adjacent-to. An analysis that registers dependencies between tokens by structural tags might be necessary. Without such dependencies it would be difficult to handle cases where complex constituents have a different position in the translation than they have in the source. A typical example in English-Swedish translations is imperative sentences, where there is some constituent preceding the verb, as in the following examples:

To delete it, press the X button.
Tryck ner X-knappen för att ta bort den.

For such cases it seems mandatory to have operations that instructs a word to move with its head. What is also almost certainly required is an ability of the analysis component to identify clause boundaries. The success of the Constraint Grammar Framework depends to a large extent on its ability to do just this. Both of these extensions are important issues for further research.

Another issue is the difference between this framework and other frameworks of corpus-based translation. Obviously it differs from statistical approaches in not using statistics to get at alignments. But it also differs from most example-based approaches by working at the word level. Brown (1996; 1999) also works with words and word strings directly, but attempts to capture structural differences by introducing categories with associated patterns that may differ for the source and target language.

Finally, it can be observed that this approach is not incompatible with other uses of translation examples, such as sentence templates. If a given sentence is found to match the source side of a template, the parts of it that match the variables of the template can be translated by the approach sketched here.

8. References

Bangalore, S. And A. K. Joshi, 1999. Supertagging: An Approach to Almost Parsing. *Computational Linguistics* 25(2): 237-265.

Bar-Hillel, Y. (1953). Machine translation. *Computers and Automation* 2(5), 1-6.

Brown, Peter, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer and Paul S. Roossin, 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2): 79-85.

Brown, Peter F., S. A. Della Pietra, V. J. Della Pietra and R. L. Mercer, 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2): 263-311.

Brown, Ralf D., 1996. Example-based machine translation in the PANGLOSS system. In *Proceedings of the* 16*th* *International Conference on Computational Linguistics*, pp. 169-174.

Brown, Ralf D., 1999. Adding linguistic knowledge to a lexical example-based translation system. *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation*. Chester, England August 23-25 1999: 22-32.

Church, K. W. And R. L. Mercer, 1993. Introduction to the Special Issue on Computational Linguistics Using Large Corpora: I. *Computational Linguistics* 19(1): 1-24.

Greene, B. B. And G. M. Rubin, 1971. Automatic grammatical tagging of English. Providence R. I., Dept of Linguistics, Brown University.

Karlsson, Fred, A. Voutilainen, J. Heikkilä and A. Antilla, 1994. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, Berlin and N. Y.

Kitano, Hiroaki, 1993. A Comprehensive and Practical Model of Memory-Based Machine Translation. *Proceedings of the 13th International Joint Conference of Artificial Intelligence,* Champery, France, pp. 1276-1282.

Knight, Kevin, 1997. Automating Knowledge Acquisition for Machine Translation. AI Magazine 18(4) 81-96.

Nagao, makato, 1994. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn and R. Bernerji (eds.) *Artificial and Human Intelligence*, North-Holland, pp. 173-180.

Och, F. J. And H. Weber, 1998. Improving statistical natural language translation with categories and rules. In *Proceedings of the 35th Annual Conference of the ACL and the 17th International Conference on Computational Linguistics*, Montreal, Canada, August 1999, pp. 985-989.

Och, F. J., Chr. Tillman and H. Ney, 1999. Improved Alignment Models for Statistical Machine Translation. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora 21-22 June 19*99, pp. 20-28.

Sato, S. And M. Nagao, 1990. Toward memory-based translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90),* Helsinki, Finland, vol. 3: 247-252.

Sumita, Eiichiro and Hitoshi Iida, 1991. Experiments and prospects of example-based Machine Translation. *Proceedings of the 29th Annual Conference of the ACL,* Berkely, CA, USA, pp. 185-192.

Turcato, D., P. McFetridge, F. Popowich and J. Toole, 1999. A unified example-based and lexicalist approach to machine translation. In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation*. Chester, England August 23-25 1999: 33-43.

Whitelock, Pete, 1994. Shake and bake translation. In C. J. Rupp, M. A. Rosner and R. L. Johnson (eds.): *Constraints, Language and Comput*ation, London, Academic Press, pp. 339-359.