Generation for Multilingual MT

Takako Aikawa, Maite Melero, Lee Schwartz, Andi Wu

Microsoft Research
One Microsoft Way
Redmond, WA 98008, USA
takakoa@microsoft.com
maitem@microsoft.com
leesc@microsoft.com
andiwu@microsoft.com

Abstract

This paper presents an overview of the broad-coverage, application-independent natural language generation component of the NLP system being developed at Microsoft Research. It demonstrates how this component functions within a multilingual Machine Translation system (MSR-MT), using the languages that we are currently working on (English, Spanish, Japanese, and Chinese). Section 1 provides a system description of MSR-MT. Section 2 focuses on the generation component and its set of core rules. Section 3 describes an additional layer of generation rules with examples that address issues specific to MT. Section 4 presents evaluation results in the context of MSR-MT. Section 5 addresses generation issues outside of MT.

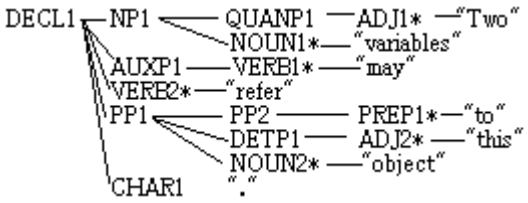
Keywords

Natural language generation, multilingual MT system, robustness, application-independence, transfer

1. MSR-MT System Description

The proposed method of natural language generation will be presented in the context of MSR-MT, a multilingual MT system being developed at Microsoft Research. MSR-MT is a hybrid system with rule-based, example-based, and statistical components. Analysis and generation are performed with linguistic parsers and syntactic realization modules, the rules of which are coded by hand. Transfer is accomplished using transfer rules/mappings that are automatically extracted from aligned corpora.

The MT process begins with the analysis of a source language sentence by the source language parser. The output, an annotated syntactic tree, is the input to the Logical Form module. This module produces a deep syntactic representation, or LF, of the source sentence (Heidorn, G. E., 2000). The LF uses the same basic set of relation types for all languages.




```
refer1 (+Modal +Pres +Proposition +I0)
  |_Modals--may1 (+Pres +I0)
  |_Tsub----variable1 (+Plur +Count)
       |_Lops--two1 (+Quant +Plur +Num)
  |_to -------object1 (+Def +Proxl +Pers3 +Sing
                      +BndPrp +Conc +Count)
  |_LTopic--variable1 ({Noun} +Plur +Count)
```

Figure 1: Syntactic Tree and Logical Form

Figure 1 gives the syntactic tree and LF for the simple English sentence "Two variables may refer to this object".

The LF is the final output of the analysis phase and the input to the transfer phase. Transfer extracts a set of mappings from the source-target language MindNet (Richardson, 2000), a translation knowledge database, and applies these mappings to the LF of the source sentence to produce a target LF. The translation MindNet for a language pair is a repository of aligned LFs and portions of LFs (produced by analyzing sentence-aligned corpora). An alignment of two LFs is a set of mappings between a node or set of nodes (and the relations between them) in the source LF and a node or set of nodes (and the relations between them) in the target LF (Menezes & Richardson, 2001).

In the translation process, the transfer component searches the alignments in the MindNet for those that match portions of the LF of the sentence being translated. Mappings with larger context are preferred to mappings with smaller context and higher frequency mappings are preferred to lower frequency mappings. The lemmas in any portion of the LF of the input sentence that do not participate in a mapping are mapped to a target lemma

using a bilingual dictionary. The target LF fragments from the transfer mappings and dictionary mappings are stitched together to produce the target LF (Menezes & Richardson, 2001).

For our example in Figure 1, the transfer component produces the following target LFs for Spanish, Japanese, and Chinese (Figure 2).[1]

Source sentence: Two variables may refer to this object.

Transferred Spanish LF:

```
hacer ({make} +Pres +Proposition)
|_Modals--poder ({may} +Pres)
|_Tsub----variable ({variable} +Plur)
     |_LOps--dos ({two} +Plur)
|_Tobj----referencia ({reference})
      |_a--objeto ({object} +Def +Proxl +Pers3 +Sing)
```

Transferred Japanese LF:

```
参照1 ({reference} +VN +Pres +Proposition +T1)
 |_Modals---かもしれない1 ({may} +Pres +I0)
 |_Tsub ----変数1 ({variable} +Plur)
       |_LOps---二つ1 ({two} +Quant +Plur)
 |_Tobj ---オブジェクト1 ({object} +Def +Proxl)
```

Transferred Chinese LF:

```
指2 ({refer to} +Pres +Proposition)
|_ Modals— 可以2 ({may} +Pres)
|_ Tsub ——变量2 ({variable} +Plur)
            |_ LOps   两2 ({two} +Plur)
|_ Tobj ——对象2 ({object} +Def +Proxl +Pers3 +Sing)
```

Figure 2: Transferred Spanish, Japanese and Chinese LFs

The transferred LF is the input to the generation component, which we will discuss in detail below.

2. Generation Component

The different language generation modules in our system are syntactic realization modules that take as input an LF characteristic of the language to be generated and produce as output a syntactic tree and surface string for that language. In this sense, they are functionally similar to the REALPRO system (Lavoie and Rambow, 1997).

The generation modules have been developed in a monolingual environment, taking as input LFs produced by the analysis module of the language and producing as output regenerated versions of the original sentences. The corpora used in the development and testing of the various generation modules include text from grammar books, encyclopedias, newspapers, and technical documents.

The generation modules are not intended specifically for MT, but rather are application-independent. They can take as input an LF produced by a dialog application, a critiquing application, a database query application, an MT application, etc. They only require a monolingual dictionary for the language being generated and an input LF that is characteristic of that language. For each language there is only one generation component. This component is used for all applications, and for MT, it is used for translation from all languages to that language.

Generation proceeds in the following way. At the beginning of generation, the input LF is converted into a basic syntactic tree that conforms to the tree geometry of the NLP system. The nodes in the LF become subtrees of this tree and the LF relations become complement/adjunct relationships between the subtrees. This basic tree, in which the connections between the syntactic nodes and the LF nodes are maintained, can be set up in different ways. For English, Spanish, and Chinese, we set it up as strictly head-initial with all the complements/adjuncts following the head, resembling the tree of a VSO language. For Japanese, we set it up as strictly head-final, with all the complements/adjuncts preceding the head.

Figure 3 gives the basic Spanish generation tree produced from the transferred Spanish LF in Figure 2.

```
REC1   REC2   "hacer"
       REC3   REC4   "variable"
              REC5   REC6  "dos"
       REC7   REC8   "referencia"
              REC9   REC10   "objeto"
```

Figure 3: Basic Syntactic Tree

The generation rules apply to the basic tree, transforming it into a target language tree. In the application of the rules, we traverse the tree in a top-down, left-to-right, depth-first fashion, visiting each node and applying the relevant rules. Each rule can perform one or more of the following operations:

(1) Assign a syntactic label to the node. For example, the "DECL" label will be assigned to the root node of a declarative sentence.
(2) Modify a node by changing some information within the node. For example, a pronoun might be marked as reflexive if it is found to be co-referential with the subject of the clause it is in.
(3) Expand a node by introducing new node(s) into the tree. For example, the "Definite" (+Def) feature on a node may become a determiner phrase attached to the syntactic subtree for that node.
(4) Delete a node. For example, for a pro-drop language, a pronominal subject may be removed from the tree.
(5) Move a node by deleting it from Position A and inserting it in Position B. For example, for an SVO language, the subject NP of a sentence may be moved from a post-verbal position to a pre-verbal position.
(6) Ensure grammatical agreement between nodes. For example, if the subject of a sentence is first person singular, those number and person features will be assigned to the main verb.
(7) Insert punctuation and capitalization.

The nodes in the generated tree are linked to each other by relations such as "head", "parent" and "sibling". The entire tree (and the entire LF) is thus visible from any given node via these relations. When a rule is applied to a

[1] English gloss is provided in Figure 2 for readability purposes only.

node, the decisions made in that rule can be based not just on features of that node, but also on features of any other node in the tree. This basically eliminates the need for backtracking, which would be necessary only if there were local ambiguities resulting from the absence of global information. This approach is similar to that of other large-scale generators (Tomita and Nyberg, 1988).

The generation rules operate on a single tree. Rule application is deterministic and thus very efficient. If necessary, the tree can be traversed more than once, as is the case in the generation modules for the languages we are currently working on. There is a "feeding" relationship among the rules. The rules that assign punctuation and capitalization, for example, do not apply until all the movement rules have applied, and movement rules do not apply until syntactic types and functional roles are assigned.

To improve efficiency and to prevent a rule from applying at the wrong time or to the wrong structure, the rules are classified into different groups according to the passes in which they are applied. Each traversal of the tree activates a given group of rules. The order in which the different groups of rules are applied depends on the feeding relations among them.

For the simple example in Figure 2 above, the Spanish, Chinese, and Japanese generation components all have an initial pass that assigns syntactic types and functional roles and a final pass that inserts punctuation marks. In addition, the Spanish component, in a first pass of rules, creates new syntactic nodes, using the information present in the LF. It turns the modal operator into the full verb *poder*, and inserts the demonstrative adjective *este* and the preposition *a* into the definite NP. In a second pass, it checks agreement, both inside the NPs and between the subject and verb, assigning the appropriate person, number, and gender information to the terminal nodes. In a later pass generation orders the subject in front of the verb and the quantifier *dos* in front of the noun. The last pass would take care of euphonic issues, such as contraction and apocopation, if such arose. Figure 4a shows the resulting tree.

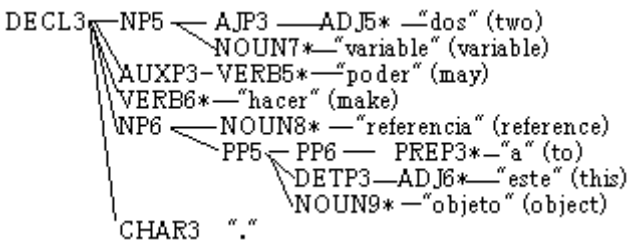


Figure 4a: Spanish generated tree

The Chinese component has a node-modification pass, which adds the classifier 个 to both 变量 (variable) and 对象 (object) and the determiner 这 to 对象 (object). In the movement pass that follows, the subject moves to preverbal position, the modal verb (may) moves between the subject and the verb, and the determiner phrase (DMP) and the number-measure phrase (NMP) move in front of their respective head nouns.

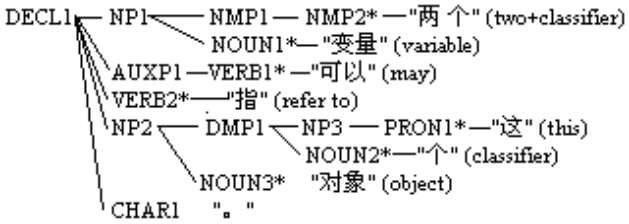


Figure 4b: Chinese generated tree

The Japanese component also has both a node-modification pass and a movement pass. In Figure 4c, three different types of syntactic elements are inserted: (i) the nominative case-marker が and the accusative case-marker を are inserted after the subject and the object, respectively; (ii) the demonstrative この is inserted at the beginning of the definite NP; and (iii) the so-called Japanese light verb する ('to do') is inserted after the verbal noun 参照 ('reference') so that the combination of the two (i.e., 参照する) can function as the verb, "refer to". In the movement pass that follows, the modal verb かもしれない ('may') is moved after the verb and the quantifier NP 二つ ('two') is moved after the subject.

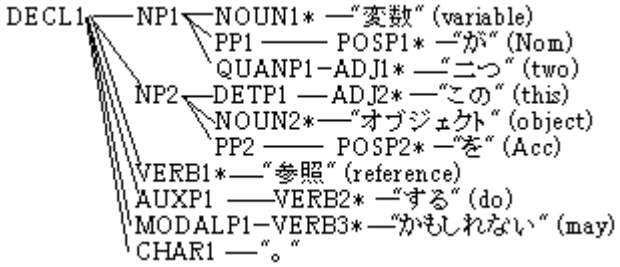


Figure 4c: Japanese generated tree

After the grammatical rules apply, the morphological rules apply to the leaf nodes of the tree. Since each node in the tree is a feature matrix and agreement information has already been assigned by the generation rules, morphological processing simply turns the feature matrices into inflected forms. For instance, in our Spanish example, the verb *poder* with the "present", "plural" and "3rd person" features is spelled out as *pueden*. Once all the words are inflected, the inflected form of each leaf node is displayed to produce the surface string. This completes the generation process.

Figure 5: Spanish generated tree with inflected nodes

3. MT-Specific Generation Issues

The generation modules described in the previous section are part of an MT system that is run on actual Microsoft technical documentation. Such real texts present challenges that are not easily reflected in model examples like the one used above to illustrate the generation process.

The Logical Forms that are automatically produced by transfer will not always be *perfect* LFs from the perspective of the target language. Some information may be missing, some may be contradictory, and some may simply not be *native* enough. To some extent, and based only on information about the language being generated, the generation components *fix* LFs, converting them into LFs that comply with the constraints imposed by the target language.

The core generation rules are application-independent and source-language-independent. Expanding the rule base to cover all the idiosyncrasies of the input would contaminate these rules and result in loss of generality. In order to maintain the integrity of the core rules while accommodating imperfect input, we have opted to add a pre-generation layer to our generation modules.

Pre-generation rules apply before the basic syntactic tree is built. They can modify the input LF by adding or removing features, changing lemmas, or even changing structural relations. Below we give examples of MT problems solved in the pre-generation layers of our different language generation modules.

The Spanish component, for example, has a pre-generation rule to deal with input LFs in which nominal nodes are assigned verbal bits (such as tense or aspect). Based both on the role of such a node in the LF and on the information present in the dictionary entry for the noun, this rule decides whether to turn the noun into a verb, remove the verbal bits from the noun, or generate a support verb for the noun. Figure 6 gives an example of an application of this rule in which the noun *acceso* (access), which is the head of a conditional clause, is replaced by the verb *acceder* (to access). This verb is retrieved from a link in the dictionary entry for the noun *acceso*.

```
Input (Transferred LF):
acceso ({Noun} (si) +Pres +Prog +Proposition)
|_a----registro ({Noun} +Indef +Pers3 +Sing)

Modified LF:
acceder ({Verb} (si) +Pres +Prog +Proposition )
 |_a----registro ({Noun} +Indef +Pers3 +Sing)

 Generated string: Si está accediendo a un registro
```

Figure 6: Spanish pre-generation example

From Chinese, we give an example of a rule that actually changes the structure of an LF. In our system, it is possible for the source and target languages to have different LF representations for similar structures. In English and other European languages, for example, the verb "BE" is required in sentences like "He is smart". In Chinese, however, no copula is used. Instead, an adjectival predicate is used. While we might attempt at the LF level to unify these representations, we have not yet done so. Moreover, the LF in our system is not intended to be an interlingua representation. Differences between languages and their LFs are tolerated. Therefore, Chinese uses a pre-generation rule to transform the be-predicate adjective LF into its Chinese equivalent as shown in Figure 7, though we soon expect transfer to do this automatically.

```
Input (Transferred) LF:
是 ({BE} +Pres +Proposition)
|_ Tobj   聪明 ({smart} +Proposition)
      |_ Tsub   他 ({he} +Pers3 +Sing +Humn)

Modified LF:
聪明 ({smart} +Pres +Proposition)
|_ Tsub   他 ({he} +Pers3 +Sing +Humn)
```

Figure 7: Chinese pre-generation example

Another example of a pre-generation rule, this time from Japanese, deals with the unspecified 1st/2nd person pronominal subject for particular types of predicates. The 1st/2nd person pronoun (わたし/あなた) is not used as the subject in sentences that express the speaker's/listener's desire (unless there is some focus/contrast on the subject). So, a Japanese pre-generation rule deletes the subject in input LFs that contain such predicates. For instance, below is the input LF, the modified LF, and the string produced from the English sentence "I want to read the book."

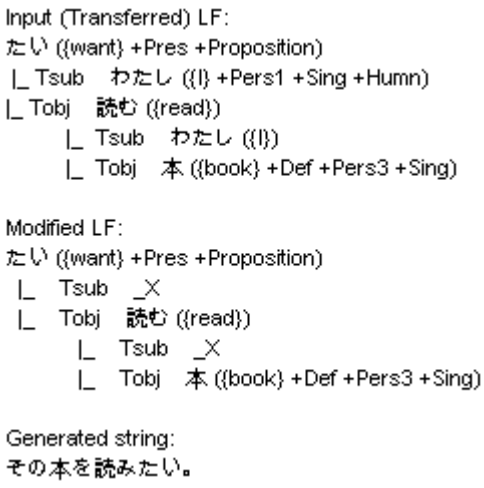


Figure 8: Japanese pre-generation example

Pre-generation rules such those described above, which modify the transferred LF, often become inactive as the MT system is trained on larger and more varied data and produces more and more target-like target LFs. At the current state of development of our MT system, however, pre-generation rules help add a degree of robustness to the system. By adjusting the structures that are passed to generation, they provide generation with the best possible input to work with. Moreover, many of the pre-generation "fixes", as we discuss in section 5, are useful for input coming from applications other than MT.

4. Evaluation

While we have not yet evaluated our generation components in isolation, we have evaluated the MT systems in which they are used. Tables 1 and 2 present the results of a comparison of our Spanish-English and English-Spanish systems with two well-known systems: Babelfish and Lernout & Hauspie (Metal), respectively[2].

Spanish-English Systems	***Mean preference score (7 raters)***	***Sample size***
MSR-MT (4/01) vs. Babelfish	0.32 ± 0.11 (at .99)	250 sentences

Table 1: Spanish-English MSR-MT vs. Babelfish

English-Spanish Systems	***Mean preference score (5 raters)***	***Sample size***
MSR-MT (4/01) vs. L&H	0.19 ± 0.14 (at 0.99)	250 sentences

[2] A rating of 1 means that raters uniformly preferred the translation produced by our system; a rating of 0 means that they did not prefer either translation; a rating of -1 means that they preferred the translation produced by the alternative system.

Table 2: English-Spanish MSR-MT vs. Lernout & Hauspie

5. Application-independent Generation

The English generation module of the Microsoft NLP system has been used in experimental question-answering, dialog, and grammar-checking applications as well as in our MT application. The same module, with the same pre-generation and core rules, is used for all the applications. Here we describe some pre-generation rules motivated by applications other than MT.

Among the pre-generation rules is one that removes the marker for non-restrictive modification (Nonrest) from LF nodes that are not in a modification relationship with another LF node. So, for example, when the question-answering application is presented with the query "When did Hitler come to power," the NLP system analyzes the question, produces an LF for it, searches its Encarta Mindnet (which contains the LFs for the sentences in the Encarta encyclopedia), retrieves the LF fragment in Figure 9, and sends it to the English generation component. The LF that is the input to generation is a portion of the LF representation of a complete sentence that includes the phrase "Hitler, who came to power in 1933." The part of the sentence that answers the question is the nonrestrictive relative clause "who came to power in 1933." Yet, we do not want to generate the answer to the question as a non-restrictive relative clause (as indicated by Nonrest in the LF), but as a declarative sentence. So, rather than pollute the core generation rules by including checks for implausible contexts in the rule for generating nonrestrictive modifiers, we use a pre-generation rule to clean up the input. The rule is application-independent (though motivated by a particular application) and can only serve to clean up bad input, whatever its source.

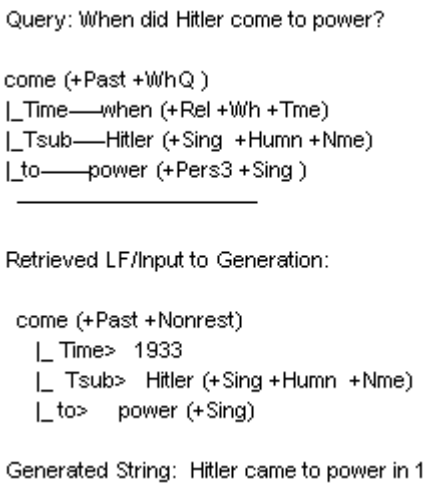


Figure 9: English pre-generation example

An example of a rule useful for an application such as grammar checking is the pre-generation rule that changes the quantifier "less" to "fewer", and vice versa, in the appropriate contexts. When the LF input to the English generation component specifies "less" as a quantifier of a plural count noun such as "cars," this rule changes the quantifier to "fewer". Conversely, when an input LF has

"fewer" specified as a quantifier of a mass noun such as "luck", the rule changes it to "less." This rule would help transfer in the machine translation of Spanish "menos" to English "less" or "few", and it would help the non-native, careless, or non-prescriptive English writer who interchanges "few" and "less". The rule in no way hurts in the generation of English from LFs that do not demonstrate this "non-native" characteristic. Rather this rule, along with the other pre-generation rules and the core generation rules, make the generation module robust and application-independent.

5. Conclusion

In this paper we have presented an overview of the natural language generation component developed at Microsoft Research and have demonstrated how this component functions within a multilingual Machine Translation system. In addition to demonstrating the basic operation of the generation component, we have shown how the component provides robustness to our real-life MT system by combining input-checking/modification techniques with core generation rules.

6. References

Heidorn, G. E. (2000): Intelligence Writing Assistance. In Dale R., Moisl H., and Somers H. (eds*.), A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text.* Marcel Dekker, New York, 1998 (published in August 2000), pages 181-207.

Jensen, K., Heidorn G., and Richardson S. (eds.) (1993): *Natural Language Processing: The PLNLP Approach*, Boston, Kluwer.

Lavoie, Benoit and Owen Rambow. (1997): A fast and portable realizer for text generation. In *Proceedings of the Fifth Conference on Applied Natural-Language Processing (ANLP-1997)*, pages 265-268.

Melero, M. and Font-Llitjos, A. (2001): Construction of a Spanish Generation module in the framework of a General-Purpose, Multilingual Natural Language Processing System. In *Proceedings of the VII International Symposium on Social Communication*, Santiago de Cuba, pages 283-287.

Menezes A. and Richardson S. (2001): A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. To appear in *Proceedings of the ACL 2001*, Toulouse, France.

Reiter, E. and Dale, R. (2000): *Building Natural Language Generation Systems,* Cambridge University Press.

Richardson, S. (2000): *The evolution of an NLP System. NLP Group Microsoft Research.* Presentation at the LREC'2000 Athens, Greece.

Tomita, M. and Nyberg E. (1988): *The GenKit and Transformation Kit User's Guide. Technical Report* CMU-CMT-88-MEMO, Centre for Machine Translation, Carnegie Mellon University.