Multilingual Sentence Generation

Takako Aikawa
Maite Melero
Lee Schwartz
Andi Wu

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

Abstract

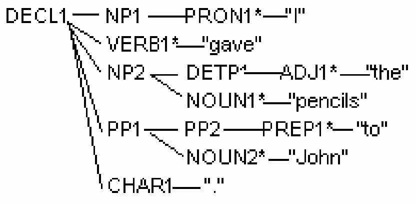
This paper presents an overview of a robust, broad-coverage, and application-independent natural language generation system. It demonstrates how the different language generation components function within a multilingual Machine Translation (MT) system, using the languages that we are currently working on (English, Spanish, Japanese, and Chinese). Section 1 provides a system description. Section 2 focuses on the generation components and their core set of rules. Section 3 describes an additional layer of generation rules included to address application-specific issues. Section 4 provides a brief description of the evaluation method and results for the MT system of which our generation components are a part.

1 System Description

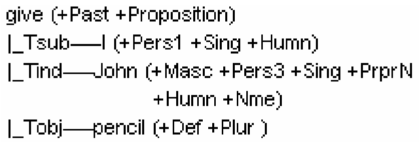
We present a natural language generation method in the context of a multi-lingual MT system. The system that we have been developing is a hybrid system with rule-based, example-based, and statistical components. Analysis and generation are performed with linguistic parsers and syntactic realization modules, the rules of which are coded by hand. Transfer is accomplished using transfer rules/mappings automatically extracted from aligned corpora.

The MT process starts with a source sentence being analyzed by the source-language parser, which produces as output a syntactic tree. This tree is input to the Logical Form module, which produces a deep syntactic representation of the input sentence, called the LF (Heidorn, G. E., 2000). The LF uses the same basic set of relation types for all languages. Figure 1 gives the syntactic tree and LF for the simple English sentence, “I gave the pencils to John”.

Tree



LF



```
give (+Past +Proposition)
|_Tsub——I (+Pers1 +Sing +Humn)
|_Tind——John (+Masc +Pers3 +Sing +PrprN
                 +Humn +Nme)
|_Tobj——pencil (+Def +Plur )
```

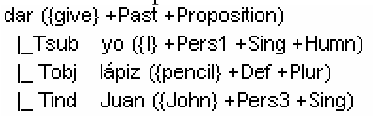
Figure 1

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).

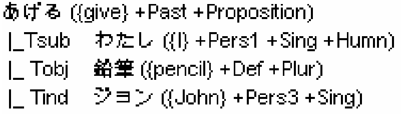
In the translation process, the transfer component searches the alignments in the MindNet for those that match portions of the LF of the sentence being translated. Mappings with larger context are preferred to mappings with smaller context and higher frequency mappings are preferred to lower frequency 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: I gave the pencils to John.
Transferred Spanish LF:



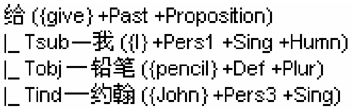
```
dar ({give} +Past +Proposition)
 |_Tsub   yo ({I} +Pers1 +Sing +Humn)
 |_ Tobj   lápiz ({pencil} +Def +Plur)
 |_ Tind   Juan ({John} +Pers3 +Sing)
```

Transferred Japanese LF:



```
あげる ({give} +Past +Proposition)
|_Tsub   わたし ({I} +Pers1 +Sing +Humn)
|_ Tobj   鉛筆 ({pencil} +Def +Plur)
|_ Tind   ジョン ({John} +Pers3 +Sing)
```

Transferred Chinese LF:



```
给 ({give} +Past +Proposition)
|_ Tsub—我 ({I} +Pers1 +Sing +Humn)
|_ Tobj—铅笔 ({pencil} +Def +Plur)
|_ Tind—约翰 ({John} +Pers3 +Sing)
```

Figure 2

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

2 Syntactic Generation Component

The different language generation modules in our system are syntactic realization components that take as input an LF characteristic of the language to be generated and produce 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 are not designed 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 that is used for all applications, and for MT, it is used for translation from all languages to that language.

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 LF become subtrees of this tree and the LF relations become complement/adjunct relationships between the subtrees. This basic tree 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 Spanish transferred LF in Figure 2.

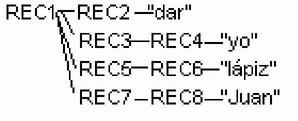


Figure 3

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.

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

(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 is thus visible from any given node via these relations. When a rule is applied to a 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. In this sense, our 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 nodetypes 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.

For the simple example in Figure 2 above, the Spanish, Chinese, and Japanese generation components all have an initial pass that assigns nodetypes and functional roles and a final pass that inserts punctuation marks.

In addition, the Spanish component, in a first pass that identifies syntactic functions, deletes the pronominal subject and inserts a dative clitic pronoun. It also inserts the definite article and the personal marker “a”. In a second pass, it checks agreement between indirect object and doubled clitic as well as between subject and verb, assigning the appropriate person, number, and gender agreement information to the terminal nodes.

Reordering operations, such as moving the clitic in front of the verb, if the verb is finite, or after, if it is non-finite, come later. The last pass takes care of euphonic issues, such as contractions or apocopated adjectives. Figure 4a shows the resulting tree.

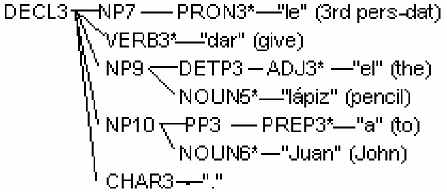


Figure 4a

The Chinese component has a node-modification pass, which adds the FUNCW node headed by 了 (le) to indicate past tense. In this pass the direct object is also turned into a prepositional phrase introduced by 把 (ba) to show the definiteness of the NP. Following this pass, a movement pass moves the subject in front of the verb.

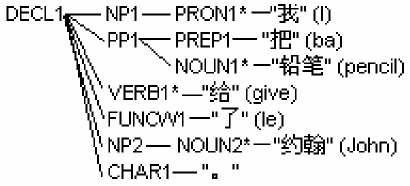


Figure 4b

The Japanese component has a pass in which case-markers or modifiers are inserted. In Figure 4c, the nominative, the accusative, and the dative case markers are inserted in the subject, direct object, and indirect object NPs, respectively. Also, the demonstrative corresponding to English "that" is inserted at the beginning of the definite NP (pencil).

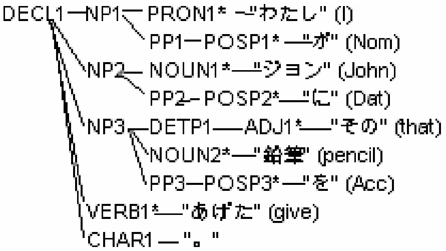
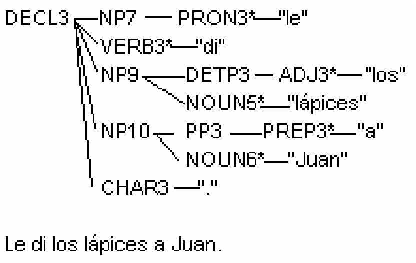


Figure 4c

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 “dar” with the “past”, “singular” and “1st person” features is spelled out as “di”. 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, as exemplified for Spanish in Figure 5.



Le di los lápices a Juan.

Figure 5

3 Application-Driven Generation

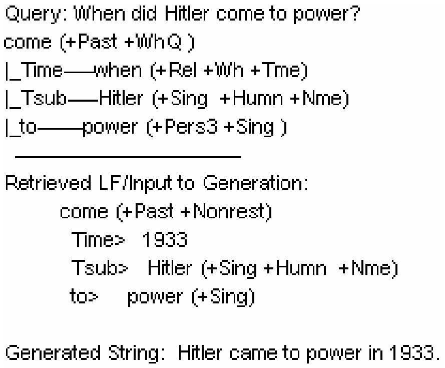
The example used in the previous sections is quite simple, and not representative of the actual problems that arise in MT. Applications, such as MT, that automatically create input for the generation component for a language will not always produce ideal LFs for that language, i.e., LFs that could have been produced by the analysis modules for that language.

We have designed the generation components, therefore, to add a degree of robustness to our applications. To some extent, and based only on information about the language being generated, the generation components will *fix* incomplete or inconsistent LFs and will verify that the structures they generate comply with the constraints imposed by the target language.

The core generation rules are designed to be application-independent and source-language-independent. Expanding the rule base to cover all the idiosyncrasies of the input would contaminate the core 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 components.

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 problems solved in the pre-generation layers of our different language generation modules. These illustrate not just the source-language independence, but also the application-independence of the generation modules.

We start with the English generation component, which was used in experimental question-answering applications before being used in MT. Among the pre-generation rules in this component is one that removes the marker indicating non-restrictive modification (Nonrest) from LF nodes that are not in a modification relationship to 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 6, and sends it to the English generation component.

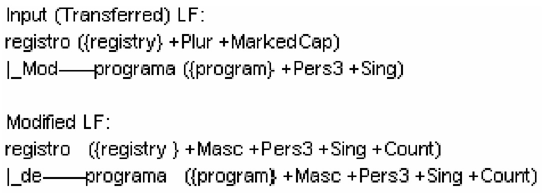


```
Query: When did Hitler come to power?
come (+Past +WhQ )
|_Time——when (+Rel +Wh +Tme)
|_Tsub——Hitler (+Sing  +Humn +Nme)
|_to———power (+Pers3 +Sing )

Retrieved LF/Input to Generation:
      come (+Past +Nonrest)
       Time>  1933
       Tsub>   Hitler (+Sing +Humn  +Nme)
       to>    power (+Sing)

Generated String:  Hitler came to power in 1933.
```

Figure 6

The LF that is the input to generation in this example 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 that 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 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, a pre-generation rule simply cleans 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.

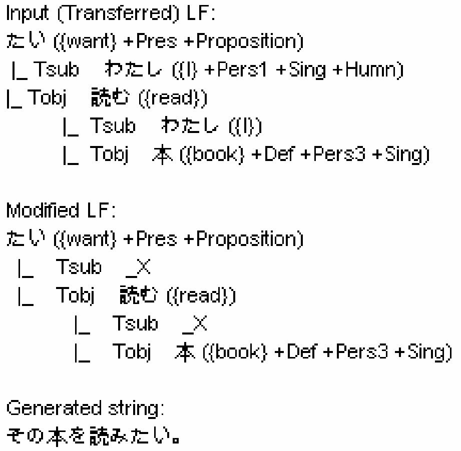
An example of a rule motivated by MT, but useful for other applications, is the pre-generation rule that changes the quantifier “less” to “fewer”, and vice versa, in the appropriate situations. When the LF input to the English generation component specifies “less” as a quantifier of a plural count noun such as “car,” 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 makes no reference to the source of the input to generation. This has the advantage that it will apply in a grammar-checking application as well as in an MT application (or any other application). If the input to English generation were the LF produced for the ungrammatical sentence "He has less cars," the generation component would produce the correct "He has fewer cars," thereby effectively grammar checking the sentence. And, if the ultimate source of the same input LF were the Spanish sentence "Juan tiene menos coches, " the result would be the same, even if "menos" which corresponds to both "less" and "fewer" in English, were not transferred correctly. Another type of problem that a generation component might encounter is the absence of necessary information. The Spanish generation component, for instance, may receive as input underspecified nominal relations, such as the one exemplified in Figure 7, in which a noun (registro) is modified by another noun (programa). The relationship between the two nouns needs to be made explicit, in Spanish, by means of a preposition when the modifying noun is not a proper noun. Absent the necessary information in the incoming LF, a pre-generation rule introduces the default preposition "de" to specify this relationship.



```
Input (Transferred) LF:
registro ({registry} +Plur +MarkedCap)
|_Mod——programa ({program} +Pers3 +Sing)

Modified LF:
registro   ({registry } +Masc +Pers3 +Sing +Count)
|_de——programa   ({program} +Masc +Pers3 +Sing +Count)
```

Figure 7

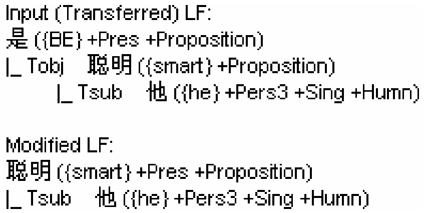
Another example of a pre-generation rule, this time from Japanese, deals with the unspecified 1st/2nd person pronominal subject for particular types of predicates. The 1st/2nd person pronoun (わたし/あなた) is not used as the subject in sentences that express the speaker's/the listener's desire (unless there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



```
Input (Transferred) LF:
たい ({want} +Pres +Proposition)
 |_ Tsub    わたし ({I} +Pers1 +Sing +Humn)
|_ Tobj    読む ({read})
        |_ Tsub    わたし ({I})
        |_ Tobj    本 ({book} +Def +Pers3 +Sing)

Modified LF:
たい ({want} +Pres +Proposition)
  |_    Tsub    _X
  |_    Tobj    読む ({read})
          |_    Tsub    _X
          |_    Tobj    本 ({book} +Def +Pers3 +Sing)

Generated string:
その本を読みたい。
```

Figure 8

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 9, though we soon expect transfer to automatically do this.



```
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 9

4 Evaluation

The generation components described in the previous sections are part of an MT system that has been run on actual Microsoft technical documentation. The system is frequently evaluated to provide a measure of progress and to yield feedback on its design and development. In evaluating our progress over time and comparing our system with others, we have performed several periodic, blind human evaluations. We focus here on the evaluation of our Spanish-English and English-Spanish systems.

For each evaluation, several human raters judge the same set of 200-250 sentences randomly extracted from our technical corpora (150K sentences).[2] The raters are not shown the source language sentence; instead, they are presented with a human translation, along with two machine-generated translations. Their task is to choose between the alternatives, using the human translation as a reference.

Table 1 summarizes a comparison of the output of our Spanish-English system with that of Babelfish (http://world.altavista.com/). Table 2 does the same for our English-Spanish system and Lernout & Hauspie's English-Spanish system (http://officeupdate.lhsl.com/). In these tables, a rating of 1 means that raters uniformly preferred the translation produced by our system; a rating of 0 means that they did not uniformly prefer either translation; a rating of -1 means that they uniformly preferred the translation produced by the alternative system.[3] Beside each rating is a confidence measure for the mean preference at the .99 level (Richardson, S., et al (2001)).

Spanish-English Systems	***Mean preference score (7 raters)***	***Sample size***
Our 4/01 (2001) MT vs. Babelfish	0.32 ± 0.11 (at .99)	250 sentences

Table 1. Our Spanish-English MT vs. Babelfish

English-Spanish Systems	***Mean preference score (5 raters)***	***Sample size***
Our 4/01 (2001) MT vs. L&H	0.19 ± 0.14 (at 0.99)	250 sentences

Table 2. Our English-Spanish MT vs. Lernout & Hauspie

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. We have provided motivation for the generation architecture, which consists of a set of core rules and a set of application-driven pre-generation rules, within a wide-coverage, robust, application-independent, multilingual natural language processing system. In addition we have presented evaluation figures for Spanish-English and English-Spanish, two of the language pairs of the MT system in which our generation components are used.

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.

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

Richardson, S., et al (2001): Overcoming the customization bottleneck using example-based MT, Paper submitted for Data-driven MT Workshop at ACL 2001, Toulouse, France.

[2] The human raters used for these evaluations work for an independent agency and played no development role building the systems they test.

[3] In interpreting our results, it is important to keep in mind that our MT system has been customized to the test domain, while the Babelfish and Lernout & Hauspie systems have not.

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.