A Corpus-Based Statistics-Oriented Two-Way Design for Parameterized MT Systems: Rationale, Architecture and Training Issues

[*,+]Keh-Yih Su, [*]Jing-Shin Chang and [+]Yu-Ling Una Hsu

[*]Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, Taiwan 30043, R.O.C.

[+]Behavior Design Corporation
No. 5, 2F, Industrial East Road IV
Science-Based Industrial Park
Hsinchu, Taiwan 30077, R.O.C.

Table of Contents

Current Status of MT Operations

- Current Status of MT Operations:

  1. Many international companies require that the MT output be "publishable" with their long established translation styles.

  2. High quality translation is currently possible only in restricted domains or sublanguages.

  3. Capability to operate in various domains is necessary for an MT system to survive or operate economically.

- Desired Mode for MT Operations:

  1. covering a wide range of applications.

  2. capable of producing high quality translation in each subdomain.

  3. capable of adapting to customer preferred styles and user feedback.

What Is One-Way Design?

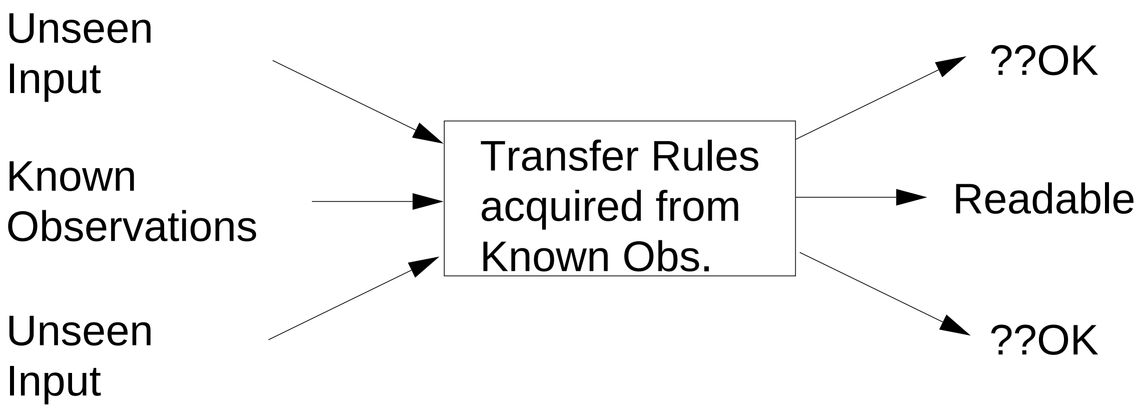
- One-Way Design:

  almost all the translation knowledge is derived, explicitly or implicitly, based on the training corpus of the *source* language.

  a. The transfered structures are usually a locally modified version of the source constructs, which usually tend to have fewer number of required translation operations and produce readable translations.

  b. The translation may not fall within the range of the target language.

  - The transfer rules are usually designed explicitly or implicitly based on the source training corpora; Such transfer rules may not be fired correctly for the unseen sentences and thus produce illegal sentences.



c. The output target sentence may not be in a preferred style under the discourse context, because the generation grammar may not take the target text around the current sentence into consideration. (Usually, only the discourse context of the source sentence is used for analysis)

Problems with One-Way Design Transfer-Based MT

- Current one-way design transfer-based MT systems fail to meet user requirements and desirable mode of MT operations:
  - The translation is usually too literal because the output is strongly affected by the sentence patterns of the source language due to the one-way processing steps from analysis, transfer to generation [Somers 93, Su 93].

Please Release the Botton.

MT: 請 把 按鈕 放開

Post-Editor: 請 放開 按鈕

- The transfer knowledge is usually maintained as a large set of rules which are not parameterized.
  - It's not easy to adapt the translation knowledge to particular domains.
  - Customized styles cannot be established easily.
  - It's very costly to construct different sets of rules for different domains.

Wish List for a Better MT Architecture and System Design Methodology

Quality Concerns:

- producing publishable translation in each domain
- respecting users' preferred styles

Adaptability:

- easily adaptable to different domains
- adapting to post-editor feedback
- automatic learning from bilingual corpora with little human intervention

Prospective Solutions

- Quality Issues
  - use a *two-way* design method for acquiring source-independent knowledge to:
    - produce publishable translation in each domain
    - respect users' preferred styles
- Domain/User Adaptation Problems
  - use a *parameterized* MT architecture for easy adaptation to
    - easily adapt to different domains
    - adapt to post-editor feedback
    - automaticly learn from bilingual corpora with little human intervention

A Typical Parameterized MT System

$$P(T_i \mid S_i) = \sum_{I_i} P(T_i, I_i | S_i)$$

$$\cong \sum_{I_i} \{ [P(T_i | PT_t(i)) \times P(PT_t(i) | NF1_t(i)) \times P(NF1_t(i) | NF2_t(i))] \quad \ldots(1)$$

$$\times \; [P(NF2_t(i) | NF2_s(i))] \quad \ldots(2)$$

$$\times \; [P(NF2_s(i) | NF1_s(i)) \times P(NF1_s(i) | PT_s(i)) \times P(PT_s(i) | S_i)] \} \quad \ldots(3)$$

where

S_i: Source Sentence T_i: Target Sentence

I_i: intermediate forms for the source-target pair

PT: parse tree (s: source & t: target)

NF1: level-1 normal form NF2: level-2 normal form

Block Diagram for A Typical Parameterized MT System

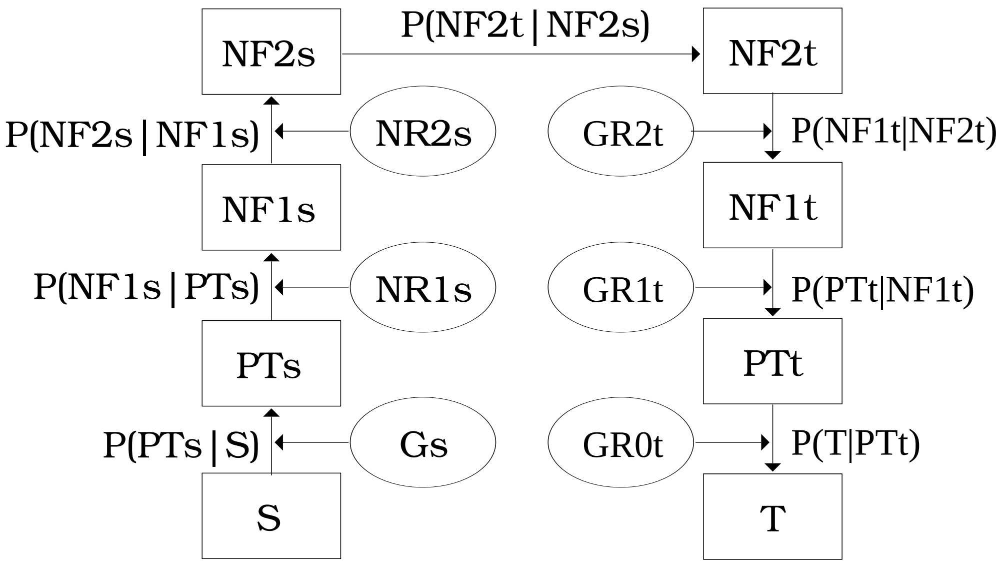


Figure 1 Translation Flow for a Parameterized MT System

S: Source Sentence

PT: parse tree

NF2: level-2 normal form

NR: normalization rules

T: Target Sentence

NF1: level-1 normal form

Gs: source grammar

GR: generation rules

- Characteristics
  1. The intermediate knowledge representation is characterized by sets of normalized constructs.
  2. Phrase structure grammars, normalization rules, generation rules are still in linguistically familiar forms (not in simplified forms like the Chomsky normal form.)
  3. Ambiguity resolution or preference evaluation is governed by sets of parameters (probabilities).
  4. Training mechanism is characterized by a quantitative optimization criterion for acquiring the language parameters (probabilities or scores) from real text corpora.

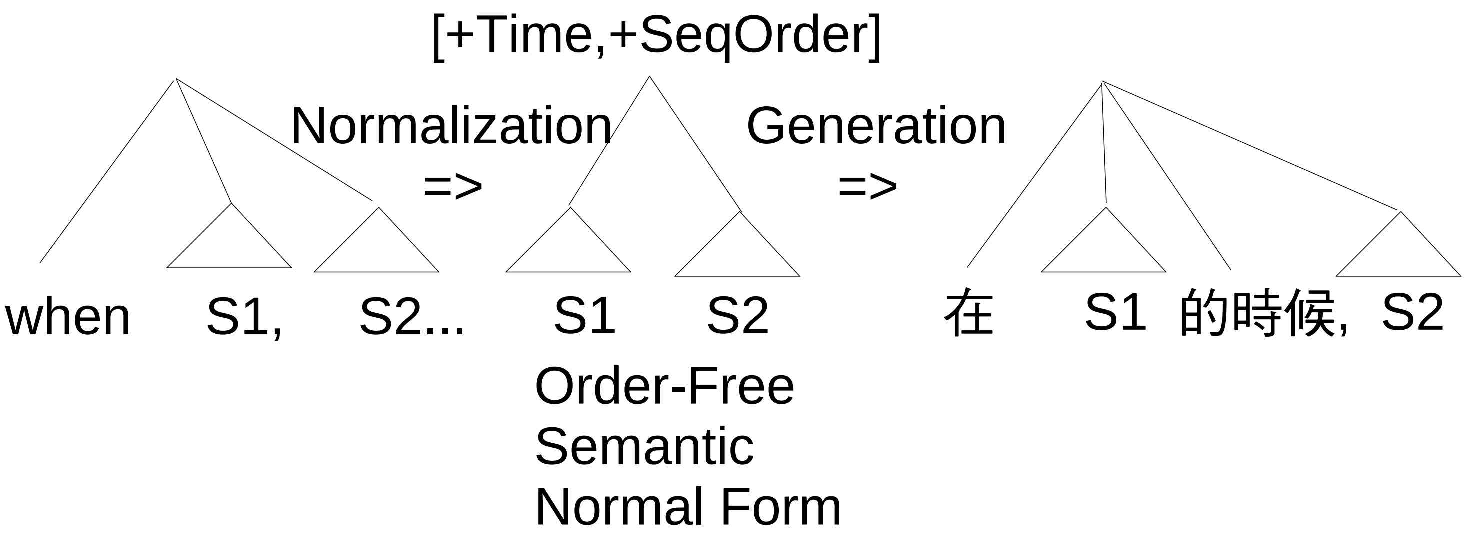
Why Parameterized MT Systems?

1. Easier to obtain the required translation knowledge in a specific domain from a large bilingual corpus.
   - The training mechanism is simply an estimation process to get the parameter sets from a corpus.
   - The search path toward the best parameter set could be found systematically.
2. The training process can be objectively optimized
3. Easier to adapt the system to specific user styles and maintain different parameters for different customers.
   - The parameter sets could be adjusted to reflect user preference.
4. Easier for on-line adapting to user feadback to avoid repeated translation errors.

Why Use Normalized Constructs?

- High level constituents are preferred: Purely statistical approaches applied on low-level constructs (like Markov N-gram approaches or simple bag-translation decoding methods) require a large parameter space to characterize well known structural information in syntactic or semantic levels.
- Fully annotated find-grain constructs are difficult to acquire and maintain.
- Normalized constructs could retain most discriminative features without losing much information; can still significantly reduce the number of parameters required.

- Normalization can be used to reduce the degree of interleaving:



Parameter Learning in Two-Way Design

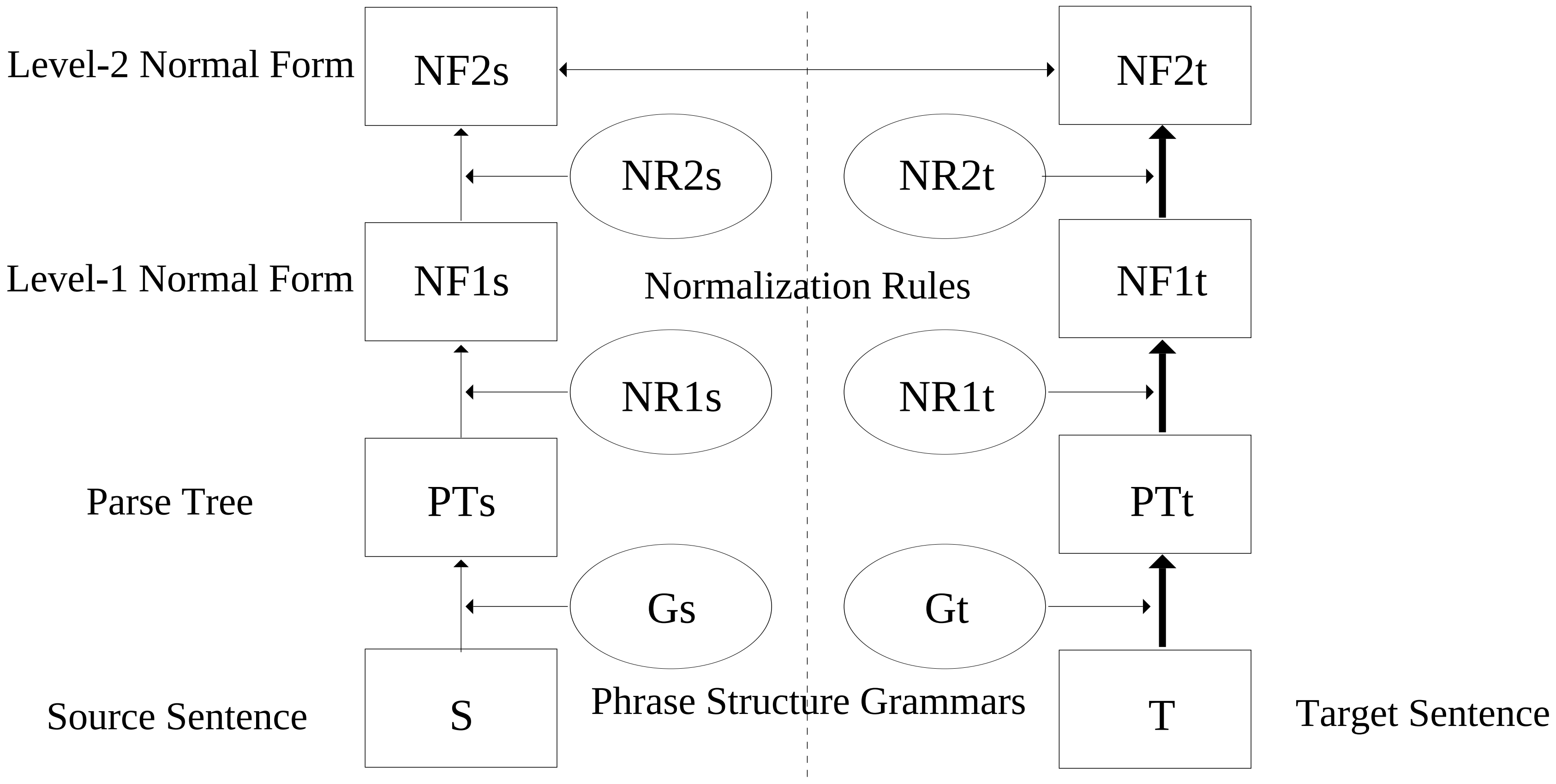


Figure 2 Two-Way Training Flow for a Parameterized MT System

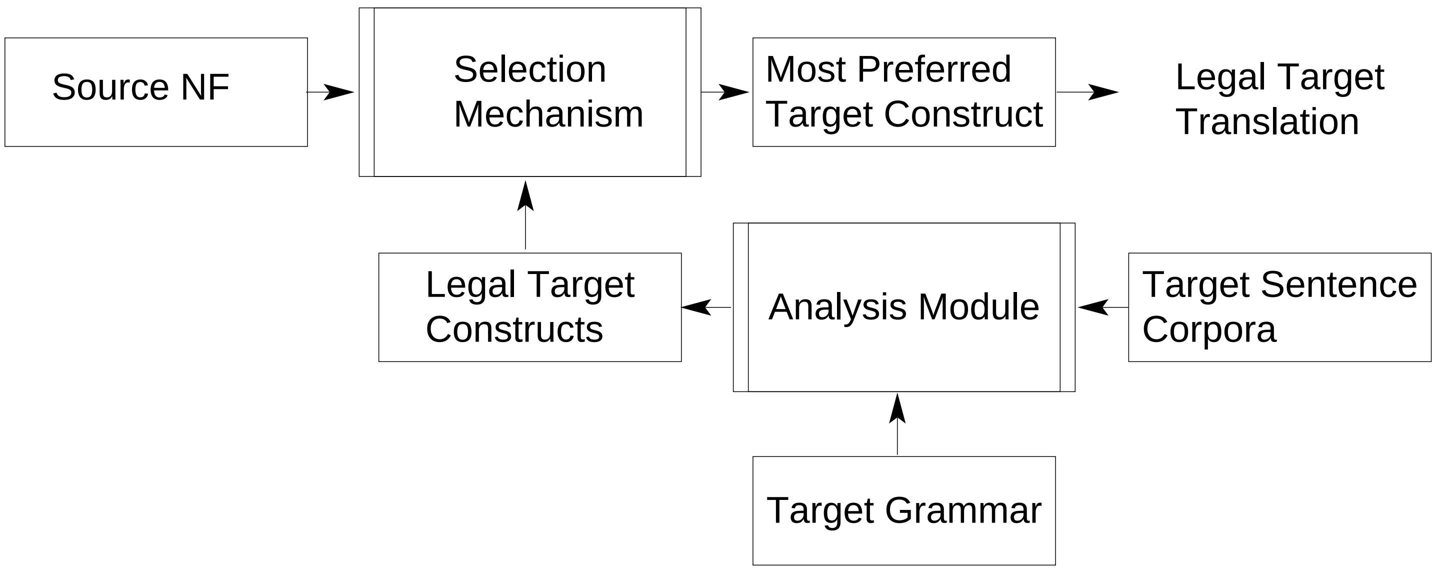
1. The PT's, NF1's and NF2's for *both* the source and target sentences are derived from the source and target sentences respectively, based on their own phrase structure grammars and normalization rules

   - to ensure that all intermediate representations fall within the range of the sentences that will be produced by the native speakers of the source and target languages.

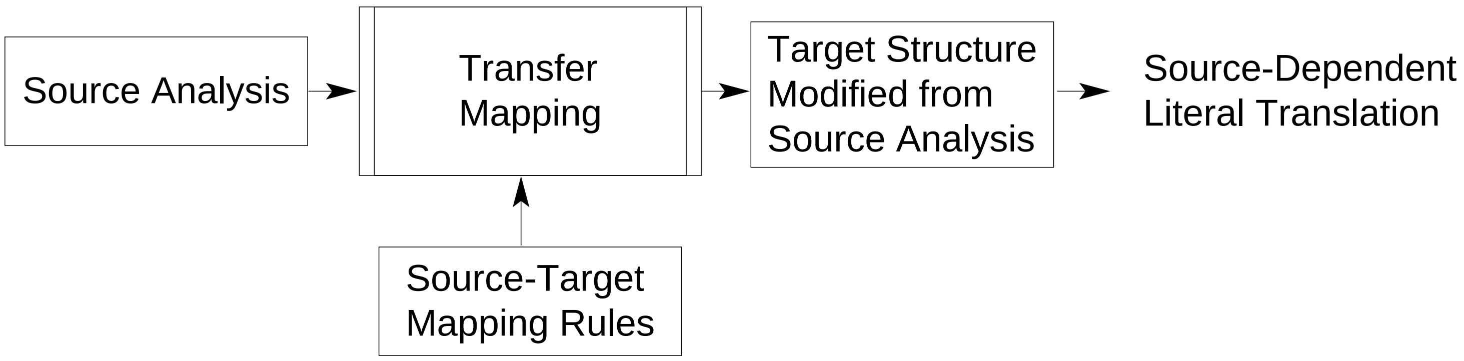
2. The transfer parameters are estimated based on such intermediate representations and the transfer knowledge is derived from both the source and target sentences of an aligned bilingual corpus.

3. In the transfer phase, the system only *select* those preferred candidates among such grammatical constructs.

Selection by Score



Transfer Mapping



Why Two-Way Design?

- to (semi-)automatically acquire the required translation knowledge from a bilingual corpus
- ensure that all intermediate representations are legal constructs to the native speakers, NOT a modified version of their source language counterparts (to avoid being affected by the source style)
- for producing high quality translation that is natural to native speakers
- get bi-directional systems at the same time (e.g., English -> Chinese and Chinese -> English)

Translation Flow in a Parameterized MT System

1. Syntactic Parsing
2. Syntactic Normalization
3. Semantic Normalization
4. Target Semantic Tree Selection
5. Normalized Structure Generation
6. Surface Structure Generation
7. Morphological Generation

Translation Flow and Normal Forms in a Parameterized MT System

1. Syntactic Parsing
   - produce parse trees
     - morphological processing
     - parsing according to source grammar

```
 SJ
  |
SIMP1 --------------------------
  |                             \
 ADTZ ----------------------      SIMP -----------------------
  |                         \     |                          \
 ADT                          \    V1                           ADTC
  |                            |   |  \                          |
 SBJ                           |   VJ   VN-T                    SBJ
  |                            |   |     |                       |
  SB                           |   |    N3J                      SB  --
  |                            |   |     |                       |    \
 SIJ                           |   |   N3-AJ                     |      V2-AJ
  |                            |   |     |                       |        |
  SI                           |   |    N3-A                     |       V2-A
  |  \                         |   |     |                       |        |
  |    V2J                     |   |    N2J                      |        V1
  |     |                      |   |     |                       |        |  \
  |     V2                     |   |     N2                      |        VJ  VSI-T
  |     |                      |   |     |  \                    |        |     |
  |     V1  -                  |   |    DET    N1                |        |    SIJ
  |     |    \                 |   |     |     |                 |        |     |
  |     VJ     VN-T            |   |     |     N*  ---           |        |     SI
  |     |       |              |   |     |     |      \          |        |     |  \
  |     |      N3J             |   |     |     N*       \        |        |     |   V2J
  |     |       |              |   |     |     |         |       |        |     |    |
  |     |     N3-AJ            |   |     |     |         |       |        |     |    V2
  |     |       |              |   |     |     |         |       |        |     |    |
  |     |      N3-A            |   |     |     |         |       |        |     |    V1
  |     |       |              |   |     |     |         |       |        |     |    |  \
  |     |      N2J             |   |     |     |         |       |        |     |    VJ  VN-T
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N2             |   |     |     |         |       |        |     |    |   N3J
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N1             |   |     |     |         |       |        |     |    |  N3-AJ
  |     |       |              |   |     |     |         |       |        |     |    |    |
  |     |       N*  ---        |   |     |     |         |       |        |     |    |   N3-A
  |     |       |      \       |   |     |     |         |       |        |     |    |    |
  |     |       N*      \     |   |     |     |         |       |        |     |    |   N2J
  |     |       |         |    |   |     |     |         |       |        |     |    |    |
  |     |       |         |    |   |     |     |         |       |        |     |    |    N2 ---
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     \
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM*   N1
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM*   N*
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   NLM    A1
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
  |     |       |         |    |   |     |     |         |       |        |     |    |   QUAN   |
  |     |       |         |    |   |     |     |         |       |        |     |    |    |     |
 comp   v       n         n    ,   v    art    n         n      cjsb      v    comp  v   quan   n

  To  meet  spectrum specific  , allow   a  30-minute warm-up before attempting to  make any calibrated
            -analyzer  -ations                                                           -measurements
```

Figure 3 Example: a Syntax Tree for a Source Sentence

2. Syntactic Normalization

- normalize syntactic variants and reduce required parameters:
    - reduce unnecessary bar levels

      e.g., N3J -> N3-AJ -> N3-A -> N2J -> N2 -> N1 ... => NP
    - reduce unnecessary recursive constructs

      e.g., N* -> N* n   => NP
    - relabelling major constructs with representative node symbols

      e.g., VN-T, N3J, N3-AJ, N3-A, N2J, N2, N1, N* => NP

```
SIMP -----------------------------------------------------------
  |                               \     \                        \
  SB -----------                   V    NP ---                   SB ---------------
  |     \       \                  |     |    \                  |        \        \
  |     V       NP ------          |    DET   NP -------         |        V        VP ----------
  |     |       |        \         |     |     |        \        |        |        |    \       \
  |     |       |         |        |     |     |         |       |        |        |    NLM      NP
  |     |       |         |        |     |     |         |       |        |        |     |       |
 comp   v       n         n        v    art    n         n      cjsb      v        v    quan     n

  To   meet  spectrum  specific  allow   a 30-minute  warm-up  before attempting  make  any calibrated
           -analyzer  -ations                                                         -measurements
```

Figure 4 Example: an NF1 Tree (Normalized Syntax Tree)

3. Semantic Normalization to further reduce the parameter space

· normalize semantic variants by:

- extract discriminative features for transfer
- normalize voice, tense, modal and type information as a feature vector
- construct a hierarchical feature structure, such as a case frame
  - annotate cases and word senses

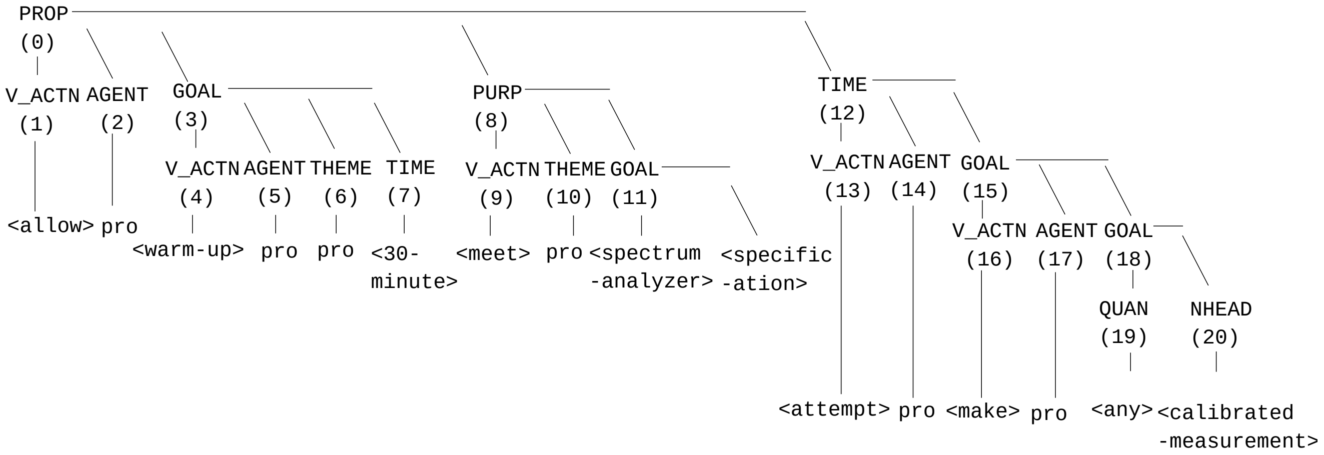


Figure 5 Example: an NF2 Tree (Normalized Semantic Tree)

4. Target Semantic Tree Selection

- select a preferred target normal form among legal target NF2's according to the trained parameters based on source normal form
  - optionally including discourse information and user feedback
- select the correct word senses in the target language

❇ the candidate target normal forms are derived from the *target* grammar and normalization rules, NOT modified from the source normal forms

❇ normal forms are order-free and thus avoid problems introduced in transferring interleaving constructs at the syntactic level (e.g., "when S" => "在 S 的時候")

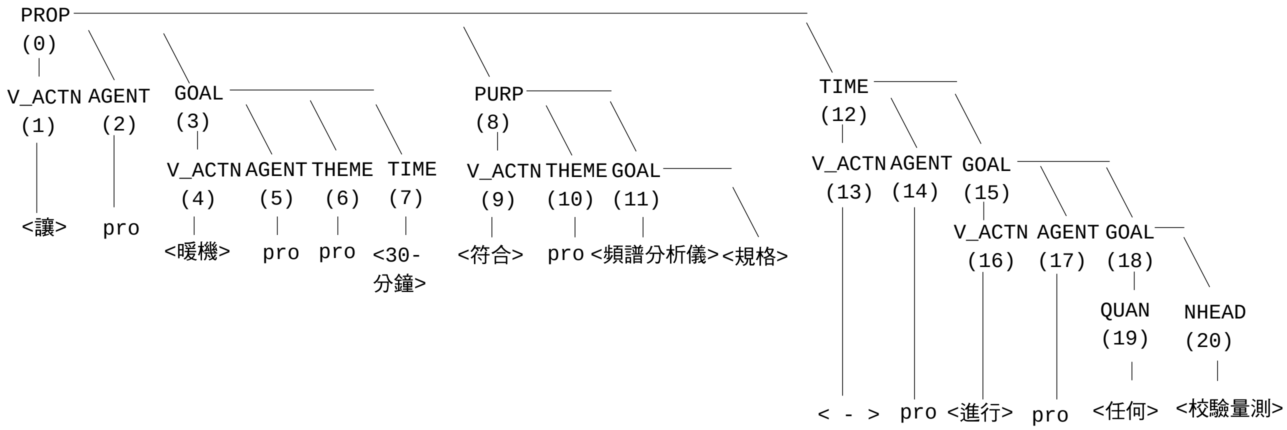


Figure 6 A Preferred Target Semantic Tree Selected from Legal Target NF2 Trees.

5. Normalized Structure Generation

- choose an appropriate normalized syntax tree for generation
- generate the major skeletal structure and major lexical items
- optionally include discourse information and user feedback in the selection process

❇ selection is made on *legal* normalized syntax trees

```
S -------------------------------------------------------------------------
 |                            \                           \    \      \     \
VADT                        VADT                        VADT    |     NP    VP ---
 |                            |                           |     |     |      |    \
 PP ----                     PP -------------------       |     |     |      |   BAK --
 |      \                     |     \              \      |     |     |      |    |    \
 |      VP ----               |     VP ---          |     |     |     |      |    |     |
 |      |      \              |      |     \        |     |     |     |      |    |     |
 |      |      NP -----       |      |      |       |     |     |     |      |    |     |
 |      |       |      \      |      |      |       |     |     |     |      |    |     |
 |      |      NLM      |     |      |      |       |     |     |     |      |    |     |
 |      |       |       |     |      |      |       |     |     |     |      |    |     |
 pv    vn       n       n    pls    vn      n      loc   advv  vnv    n     vi    q    cltm
 爲了  符合  頻譜分析儀  規格  在   進行   校驗量測   之前   請    讓   儀器  暖機  30  分鐘
```

Figure 7 Normalized Syntax Tree for the Target Sentence

6. Surface Structure Generation

- generate a complete syntax tree

```
S -------------------------------------------------------------------------------
  |                         \    \                           \   \     \     \     \
 VADT                         | VADT                          | VADT    |   NP    VP ---
  |                           |    |                          |   |     |    |      |    \
  PP ----                     |  PP --------------------      |   |     |    |      |  BAK --
  |      \                    |    |     \              \     |   |     |    |      |    |    \
  |      VP ----              |    |    VP ----          |    |   |     |    |      |    |     |
  |      |      \             |    |      |     \        |    |   |     |    |      |    |     |
  |      |      NP -----      |    |      |      |       |    |   |     |    |      |    |     |
  |      |       |      \     |    |      |      |       |    |   |     |    |      |    |     |
  |      |      NLM      |    |    |      |      |       |    |   |     |    |      |    |     |
  |      |       |       |    |    |      |      |       |    |   |     |    |      |    |     |
  pv    vn       n       n   cj  pls    vn       n     loc  cj advv  vnv     n    vi     q   cltm
  爲了 符合  頻譜分析儀 規格  ， 在    進行  校驗量測  之前  ，  請    讓  儀器   暖機  30   分鐘
```

Figure 8 Target Syntax Tree

7. Morphological Generation

- generate target specific morphemes, like Chinese『們』(plural form)、『了』、『過』(past tense)

Example:

- Reset the scale after the calibrated measurements.
- 進行『過』校驗量測之後，請重新設定刻度。

Two-Way Design Issues

1. Optimization:
   What's the optimization criteria for the translation task and how to design a feasible system according to the optimization function?

2. Parameterization:
   How to map the hierarchical normal form structures into statistical events and express the optimization function in terms of such events ([Su 91a, Chang 93])?

3. Parameter Estimation:
   How to estimate the parameters from annotated corpora with MLE (Maximum Likelihood Estimation), adaptive training and robust estimation (Amari 67, Juang 92, Chiang 92, Su 91b, 94a)?

4. Unsupervised Training:
   How to estimate the parameters from raw translation pairs with a Viterbi Training or EM method as the solution to the Two-End constraint optimization problem?

Translation Model and Optimization Criteria

- If discourse context is not considered, the best translation would be the one which has the highest translation score as follows:

$$\hat{T} = \operatorname*{argmax}_{T_i} P(T_i | S_i, EM, CM, TM)$$

where

S_i: i^{th} source sentence T_i: i^{th} target sentence

EM: English (source) language Model

CM: Chinese (target) language Model

TM: Translation/Transfer Model

□ If discourse information (in the form of normal forms of the previous sentences) is considered, the one with the highest discourse-dependent translation score is preferred:

$$P\left(T_i | S_i, I_1^{i-1}\right)$$

where

I_1^{i-1}: source & target normal forms $I_1, I_2, \ldots, I_{i-1}$ for previous sentences, and

$$I_i = \left\{PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i)\right\}$$

Approximation of Translation Score for a Single Sentence

$$
\begin{aligned}
P(T_i \mid S_i) &= \sum_{I_i} P(T_i, I_i | S_i) \\
&= \sum_{I_i} P(T_i, PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i) | S_i) \\
&= \sum_{I_i} \{ P(T_i | PT_t(i), NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i) \\
&\times P(PT_t(i) | NF1_t(i), NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i) \\
&\times P(NF1_t(i) | NF2_t(i), NF2_s(i), NF1_s(i), PT_s(i), S_i) \\
&\times P(NF2_t(i) | NF2_s(i), NF1_s(i), PT_s(i), S_i) \\
&\times P(NF2_s(i) | NF1_s(i), PT_s(i), S_i) \\
&\times P(NF1_s(i) | PT_s(i), S_i) \\
&\times P(PT_s(i) | S_i) \} \\
&\cong \sum_{I_i} \{ [P(T_i | PT_t(i)) \times P(PT_t(i) | NF1_t(i)) \times P(NF1_t(i) | NF2_t(i))] \quad \ldots(1) \\
&\times [P(NF2_t(i) | NF2_s(i))] \quad \ldots(2) \\
&\times [P(NF2_s(i) | NF1_s(i)) \times P(NF1_s(i) | PT_s(i)) \times P(PT_s(i) | S_i)] \} \quad \ldots(3)
\end{aligned}
$$

where

(1) = generation score (2) = transfer score (3) = analysis score

Selection of the Best Target Translation

$$T_i^* = \operatorname*{argmax}_{T_i} \left\{ \max_{I_i} \left[P(T_i, I_i | S_i) \right] \right\}$$

$$= \operatorname*{argmax}_{T_i} \{ \max_{I_i} \left[P\big(T_i | PT_t(i)\big) \times P\big(PT_t(i) | NF1_t(i)\big) \times P\big(NF1_t(i) | NF2_t(i)\big) \right.$$
$$\times P\big(NF2_t(i) | NF2_s(i)\big)$$
$$\left. \times P\big(NF2_s(i) | NF1_s(i)\big) \times P\big(NF1_s(i) | PT_s(i)\big) \times P\big(PT_s(i) | S_i\big) \right] \}$$

- To make it feasible, the target translation with the highest maximum value of path scores is selected as the preferred translation.
- Including Discourse Information from the normal forms of the previous target sentences (for example, by using the translation score: $P\big(T_i | S_i,\ NF2_t(i-2),\ NF2_t(i-1)\big)$) could enhance the target style generation (i.e., suppress the source style dependency), since target normal forms are emphasized in the selection process.

Two-End Constraint Optimization for Parameter Learning in Two-Way Design

- The required parameters could be learned from bilingual corpora by considering it as a two-end constraint optimization problem.
- Formulation: Find the set of parameters Λ_{MAX} which jointly maximizes the following likelihood function for all translation pairs-

$$\begin{aligned}\Lambda_{MAX} &= \underset{\Lambda}{\operatorname{argmax}} P\left(T_1^N \mid S_1^N, \Lambda\right) \\ &= \underset{\Lambda}{\operatorname{argmax}} \sum_{I_1^N} P\left(T_1^N, I_1^N \mid S_1^N, \Lambda\right)\end{aligned}$$

where

S_1^N : all source sentences $S_1, S_2, \ldots, S_N$

T_1^N : all target sentences $T_1, T_2, \ldots T_N$

I_1^N : intermediate normal forms $I_1, \ldots, I_N$

- It's reasonable to estimate the parameters by considering only the best intermediate normal forms with a Viterbi Training method (Rabiner 86) to optimize the maximum value (instead of the sum) of the path scores:

$$\Lambda_{MAX} = \operatorname*{argmax}_{\Lambda}\left[\max_{I_1^N} P\left(T_1^N, I_1^N \mid S_1^N, \Lambda\right)\right].$$

- Simplification:

$$
\begin{aligned}
& P\left(T_1^N, I_1^N \mid S_1^N, \Lambda\right) \\
&= P\left(T_1^N | I_1^N, S_1^N, \Lambda\right) \cdot P\left(I_1^N | S_1^N, \Lambda\right) \\
&= \prod_{i=1}^{N} [P\left(T_i | T_1^{i-1}, I_1^N, S_1^N, \Lambda\right) \cdot P\left(I_i | I_1^{i-1}, S_1^N, \Lambda\right)] \\
&\cong \prod_{i=1}^{N} [P\left(T_i | I_{i-m}^{i}, \Lambda\right) \cdot P\left(I_i | I_{i-m}^{i-1}, S_i, \Lambda\right)] \\
&\cong \prod_{i=1}^{N} [P\left(T_i | I_i, \Lambda\right) \cdot P\left(I_i | S_i, \Lambda\right)] \qquad (m = 0)
\end{aligned}
$$

where

m: discourse window (m=0 <=> discourse independent model)

$$\prod_{i=1}^{N}[P(T_i|I_i,\Lambda)\cdot P(I_i|S_i,\Lambda)]$$

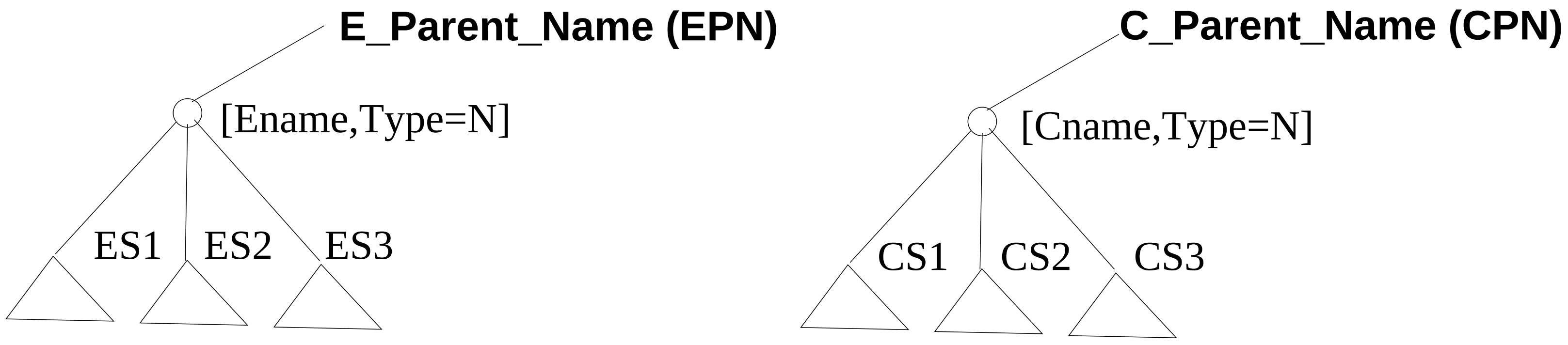
$$\cong \prod_{i=1}^{N}\{[P(T_i|\ PT_t(i),\Lambda)\times P(PT_t(i)|NF1_t(i),\Lambda)\times P(NF1_t(i)|NF2_t(i),\Lambda)]$$
$$\times\ [P(NF2_t(i)|NF2_s(i),\Lambda)]$$
$$\times\ P(NF2_s(i)|NF1_s(i),\Lambda)\times P(NF1_s(i)|PT_s(i),\Lambda)\times P(PT_s(i)|S_i,\Lambda)]\}$$

- Optimization with discourse information:

  When the discourse context is considered, the following likelihood function is used to estimate the best parameters: (m ≠ 0)

$$P(T_1^N,I_1^N\,|\,S_1^N,\Lambda)$$
$$\cong \prod_{i=1}^{N}[P(T_i|I_{i-m}^{i},\Lambda)\cdot P(I_i|I_{i-m}^{i-1},S_i,\Lambda)]$$

Formulation for Normal Form Mapping



Mapping Score:

$$
\begin{aligned}
& P(\,CTree \mid ETree, EPN, CPN) \\
&= \sum_{mapping} P(CTree, Mapping \mid ETree,\ EPN,\ CPN) \\
&= \sum_{mapping} P(CTree \mid Mapping, ETree,\ EPN,\ CPN) \times P(Mapping \mid ETree,\ EPN,\ CPN) \\
&= \sum_{mapping} P(Mapping \mid ETree,\ EPN,\ CPN)
\end{aligned}
$$

Use Viterbi Decoding:

$$\underset{CTree}{argmax} P(CTree | ETree, EPN, CPN)$$

$$\Rightarrow \underset{CTree, Mapping}{argmax} P(CTree, Mapping | ETree, EPN, CPN)$$

$$= \underset{CTree, Mapping}{argmax} P(Mapping | ETree, EPN, CPN)$$

Example:

$$
\begin{aligned}
& P(Mapping | ETree, EPN, CPN) \\
&= P(Ename \rightarrow Cname, ES1 \rightarrow CS1, ES2 \rightarrow CS4, ES3 \rightarrow \Phi, \Phi \rightarrow CS3 | \ldots) \\
&= P(\Phi \rightarrow CS3 | \ldots, Ename \rightarrow Cname, ETree) \\
&\times P(ES3 \rightarrow \Phi | \ldots, Ename \rightarrow Cname, ETree) \\
&\times P(ES2 \rightarrow CS4 | \ldots, Ename \rightarrow Cname, ETree) \\
&\times P(ES1 \rightarrow CS1 | Ename \rightarrow Cname, ETree) \\
&\times P(Ename \rightarrow Cname | ETree) \\
&\approx P(\Phi \rightarrow CS3 | Ename, Cname) \\
&\times P(ES3 \rightarrow \Phi | Ename, Cname) \\
&\times P(ES2 \rightarrow CS4 | ES2, Ename, Cname) \\
&\times P(ES1 \rightarrow CS1 | ES1, Ename, Cname) \\
&\times P(Cname, N | Ename, N, EPN, CPN)
\end{aligned}
$$

□ Recursive Expansion for Mapping Score:

The mapping between nodes are in the forms:

$$P\left([Node_T, Type] | [Node_S, Type], Ename, Cname\right)$$
$$\equiv P\left([N_T, T_T] | [N_S, T_S], Ename, Cname\right)$$
$$= P\left(N_T | T_T, [N_S, T_S], Ename, Cname\right) \times P\left(T_T | [N_S, T_S], Ename, Cname\right)$$
$$where$$
$$Type = N(onterminal),\ T(erminal)\ or\ \Phi\ (Null)$$

For the various types of nodes:

1. $T_T = N,\ T_S = N$

$$\approx P\left(N_T | N_S, Ename, Cname\right)$$

2. $T_T = T,\ T_S = N$

$$\approx P\left(N_T\_word\_sense | N_S\_name, Ename, Cname\right)$$

3. $T_T = N,\ T_S = T$

$$\approx P\left(N_T\_name \mid N_S\_word\_sense, Ename, Cname\right)$$

4. $T_T = T,\ T_S = T$

$$\approx P\left(N_T\_word\_sense \mid N_S\_word\_sense, T_T = T, Ename, Cname\right)$$

5. $T_T = N,\ T_S = \Phi$

$$\begin{aligned} & P(Mapping \mid \Phi, EPN, CPN) \\ & = P(\Phi \to Cname,\ \Phi \to CS1, \Phi \to CS2 \mid \Phi, EPN, CPN) \\ & \approx P(CS2 \mid \Phi, Cname, EPN) \\ & \times P(CS1 \mid \Phi, Cname, EPN) \\ & \times P(Cname \mid \Phi, EPN, CPN) \end{aligned}$$

6. $T_T = T,\ T_S = \Phi$

$$\approx P\left(N_T\_word\_sense \mid T_S = \Phi, T_T = T, Ename, Cname\right)$$

7. $T_T = \Phi,\ T_S = N$

$$
\begin{aligned}
& P(Mapping|ETree,EPN,CPN) \\
& = P(Ename \rightarrow \Phi,\ ES1 \rightarrow \Phi, ES2 \rightarrow \Phi | ETree, EPN, CPN) \\
& \approx P(\Phi|ES2,Ename,CPN) \\
& \times P(\Phi|ES1,Ename,CPN) \\
& \times P\left(\Phi|Ename,T_S = N,EPN,CPN\right)
\end{aligned}
$$

8. $T_T = \Phi,\ T_S = T$

$$
\approx P\left(\Phi|N_S\_word\_sense,Ename,Cname\right)
$$

A Viterbi Training Approach for Solving the Two-End Constraint Problem

- Two-way constraint optimization: Learn the translation parameters given only a parallel corpus of the source sentence and preferred target translation, the functional forms of the normalized constructs.
- Unsupervised Training is preferred due to the high costs in fully annotating the corpora.

□ A Viterbi Training Approach:

1. Initial Selection: For each (S_i, T_i) pair, derive all possible parse trees, NF1 trees and NF2 trees for S_i and T_i, respectively. And randomly choose a path from S_i and T_i toward the NF2 trees. The NF2 trees selected in this way are considered a transfer pair for the translation.
2. Estimation: Estimate the parameters with the Maximum Likelihood Estimator (MLE) based on the corresponding transfer pairs, parse trees and normal forms along the selected translation path.
3. Re-selection: Compute the translation scores for the various possible paths, which specify the combination of parse tree, NF1 and NF2, with the new parameters acquired in step 2, and select the path with the largest translation score.
4. Re-estimation: go to step 2, unless the parameters converge to a particular stopping criterion.

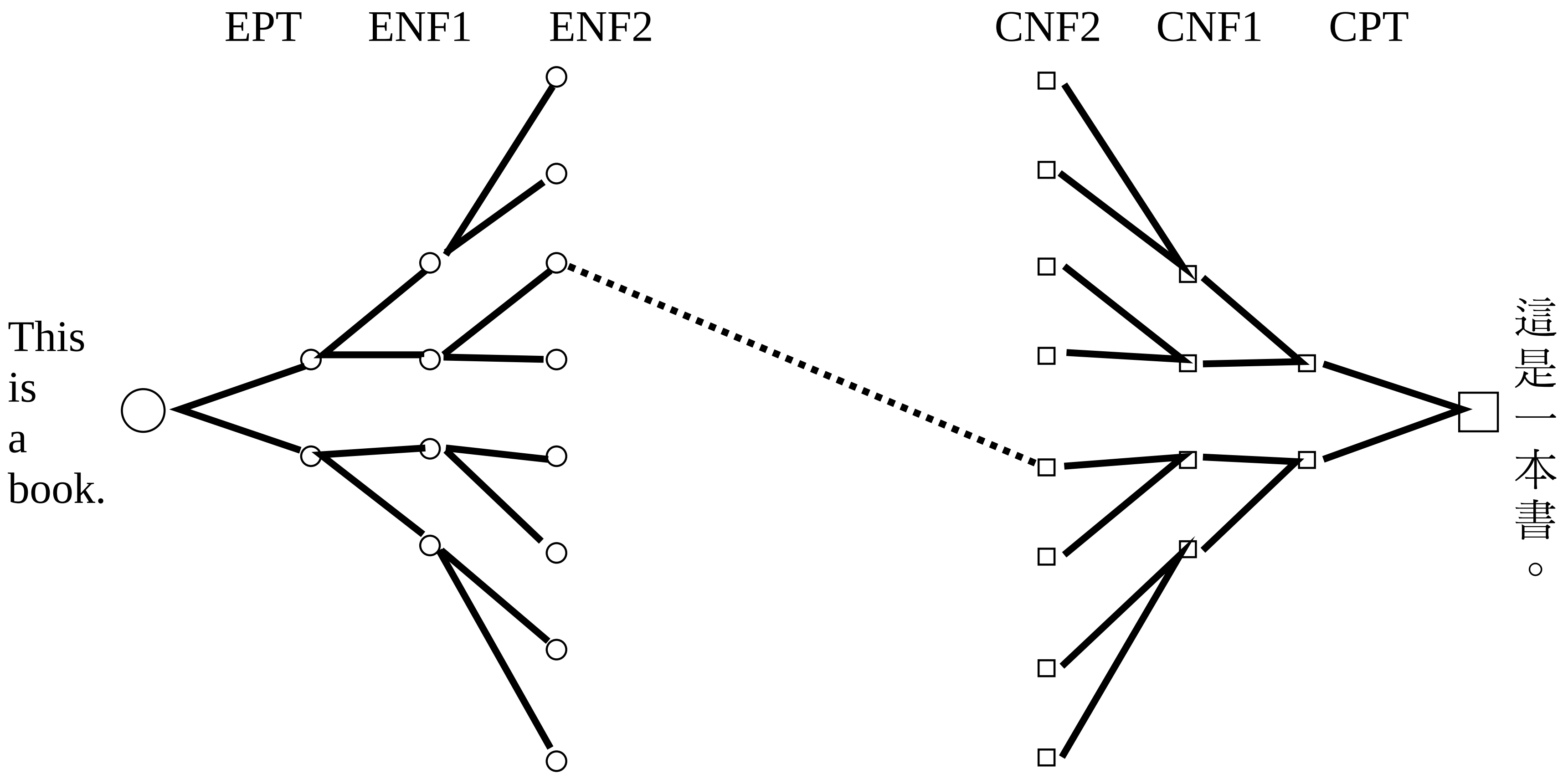


Figure 9 A Two-Way Training Method for Learning the Translation Knowledge.

Including User Preference and Feedback

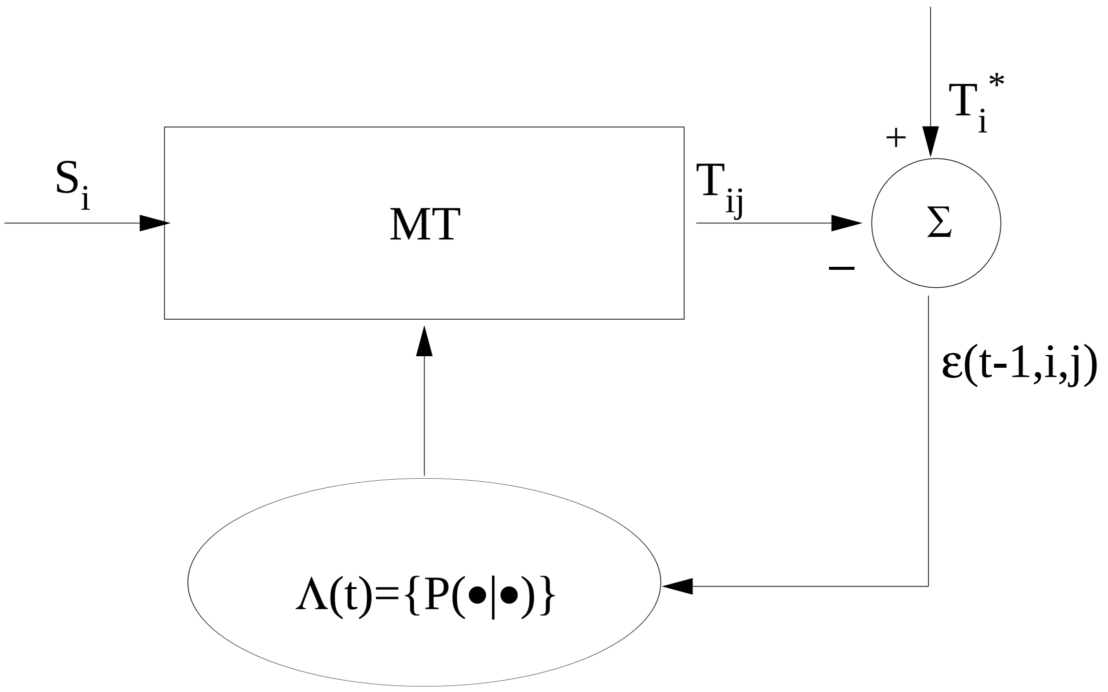
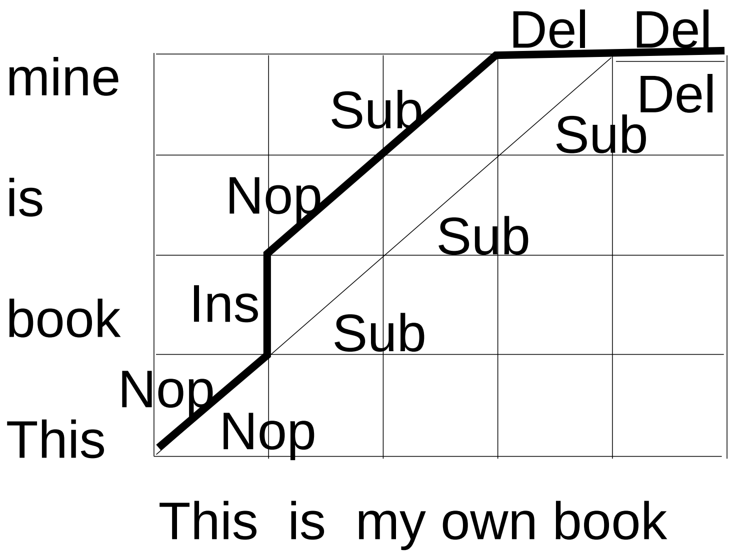


Figure 10 Feedback Controlled Parameterized MT Architecture

- The parameters could be further modified to respect special user styles or particular cost/distance metrics.

- Metrics for error distance

  (i) Levison Distance



Ins: Insertion
Del: Deletion
Sub: Substitution
Nop: No Operation

  (ii) $\varepsilon(t-1, i, j) = log P(T_i^* | S_i) - log P(T_{ij} | S_i)$

- Two-modes for feedback control:
  - Batch mode: for users with large volumes of translation
    - adjust parameters with training by epoch method [Schalkoff 92]
  - On-line mode: for on-line post-editor to avoid repeated errors
    - adjust parameters with training by sample method [Schalkoff 92]

Concluding Remarks

1. The best mode for MT operations is to have an architecture capable of
   - operating in several subdomains
   - producing publishable translation in each domain
   - respecting user styles
   - adapting to post-editor feedback
2. A parameterized architecture will make it possible to configure the MT system adaptively to the various domains and users.
3. High quality translation is possible with a two-way design method to avoid source dependency.
4. Optimization in the two-way design could be achieved by applying two-end constraint optimization on properly normalized constructs based on parallel bilingual corpora.