Syntactic Annotation

Julia Dobczynska
Nazanin khazaee Farid

Tuebingen University
July 2018
Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts

Benoit Lavoie, Michael White, and Tanya Korelsky (2001)
Overview

1. Lavoie vs Meyers Approaches

2. Data Preparation
   ○ Bi-text
   ○ Baseline Transfer Dictionary

3. Transfer Rule Induction
   ○ Aligning Parse Nodes
   ○ Generating Transfer Rule Candidate
   ○ Ordering Transfer Rule Candidates
   ○ Filtering Transfer Rule Candidates

4. Evaluation
   ○ Tree Accuracy Recall
   ○ Tree Accuracy Precision
Approach

- a novel approach **inducing transfer rules** from syntactic parses of bi-texts and bilingual dictionaries

- Approach based on:
  - lexico-structural transfer (Nasr et. al., 1997)
  - recent work reported in (Han et al., 2000)
  - syntactic approaches to EBMT such as (Nagao, 1984; Sato and Nagao, 1990; Maruyama and Watanabe, 1992).
  - the recent work of (Meyers et al., 1998)
Lavoie vs Meyers

Similarities:

- transfer rules are derived after aligning the source and target nodes of corresponding parses
### Differences:

<table>
<thead>
<tr>
<th><strong>Lavoie</strong></th>
<th><strong>Meyers</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>arses containing lexical labels, syntactic roles, and other syntactic</td>
<td>parses contain only lexical labels and syntactic roles (as arc labels)</td>
</tr>
<tr>
<td>information (tense, number, person, etc.)</td>
<td></td>
</tr>
<tr>
<td>no such restriction (KOREAN-ENG)</td>
<td>the alignment of source and target nodes preserves node dominancy</td>
</tr>
<tr>
<td></td>
<td>(syntactically closely related languages; ENG-SPANISH)</td>
</tr>
<tr>
<td>all source and target tree sub-patterns match a subset of the parse</td>
<td>the exact tree fragments in the source and</td>
</tr>
<tr>
<td>features (must satisfy alignment constraints and attribute constraints)</td>
<td>target parse match one another</td>
</tr>
<tr>
<td>the nodes of transfer rules do not have to be lexicalized</td>
<td>the source and target patterns of each transfer rule are fully lexicalized</td>
</tr>
</tbody>
</table>


Lavoie-like Tree Example

Figure 1: Syntactic analysis of a sentence as a parse tree
Figure 1: A Pair of Aligned Trees

Excel vuelve a calcular valores en libro de trabajo

Excel recalculates values in workbook
Research Methodology

- **data preparation** - creating resources from the bi-texts and bi-dictionaries
- **transfer rule induction** - inducing rules from the training set of corresponding source and target parses
- **evaluation** - producing two sets of transferred parses and calculating tree accuracy recall and precision
Overview

1. Lavoie vs Meyers Approaches

2. **Data Preparation**
   - Bi-text
   - Baseline Transfer Dictionary

3. **Transfer Rule Induction**
   - Aligning Parse Nodes
   - Generating Transfer Rule Candidate
   - Ordering Transfer Rule Candidates
   - Filtering Transfer Rule Candidates

4. **Evaluation**
   - Tree Accuracy Precision
   - Tree Accuracy Recall
Data Preparation

I. Bi-text
   ● creating a training set and a test set of source and target parses
   ● turning them into a syntactic dependency representation

II. Baseline Transfer Dictionary (BTD)
   ● taking the lexico-syntactic transfer dictionary developed by Han et al. (2000)
   ● removing the (more general) rules that were not fully lexicalized
Data Preparation - Parsing the Bi-text

- corpus of a Korean dialog of 4183 sentences and their English human translations

- two off-the-shelf parsers
  - Korean parser (Yoon et al., 1997)
  - English parser (Collins, 1997)

- automatic conversion of phrase structure output of the Collins parser into the syntactic dependency representation
  - realizer - RealPro (Lavoie and Rambow, 1997)

- output of the Yoon parser left as it is
Data Preparation - Parsing the Bi-text

The syntactic dependency representation is based on the deep-syntactic structures (DSyntS) of Meaning-Text Theory (Mel’cˇuk, 1988)

The important features of a DSyntS:

- unordered tree with labeled nodes and labeled arcs
- all the nodes are labeled with lexemes from the target language
- arcs of the tree are labeled with syntactic relations such as SUBJECT, rather than semantic relations such as AGENT
Data Preparation - Parsing the Bi-text

(S1) {i} {Ci-To-Reul} {Ta-Si} {Po-Ra}.
    this + map-accusative + again + look-imp
(D1) {po} [class=vbma ente={ra}] (    
    s1 {ci-to} [class=nnin2 pPCA={reul}] (    
        s1 {i} [class=ande]      
    )
    s1 {ta-si} [class=adco2]
)
(S2) Look at the map again.
(D2) look [class=verb mood=imp] (    
    attr at [class=preposition] (    
        ii map [class=common noun article=def]      
    )
    attr again [class=adverb]
)
Data Preparation - Parsing the Bi-text

Output: approx. half of the parse pairs contained incorrect dependency assignments, incomplete lemmatization or incomplete parses.

- selecting a higher quality subset of 1763 sentence pairs
  - source or target parse contained more than 10 nodes were rejected
  - parse pairs where the source or target parse contained non-final punctuation

- dividing this subset into training and test sets by randomly choosing 50% of the 1763 higher quality parse pairs
Data Preparation - Creating a BTD

- any available bilingual dictionaries can be combined to create the Baseline Transfer Dictionary (BTD)
  - lexical transfer dictionaries extracted from the bi-texts using statistical methods
  - existing bilingual dictionaries
  - handcrafted lexico-structural transfer dictionaries

Lavoie’s BTD:

- taking the lexico-syntactic transfer dictionary developed by Han et al. (2000)
- removing the (more general) rules that were not fully lexicalized
Overview

1. Lavoie vs Meyers Approaches

2. Data Preparation
   - Bi-text
   - Baseline Transfer Dictionary

3. Transfer Rule Induction
   - Aligning Parse Nodes
   - Generating Transfer Rule Candidate
   - Ordering Transfer Rule Candidates
   - Filtering Transfer Rule Candidates

4. Evaluation
   - Tree Accuracy Recall
   - Tree Accuracy Precision
Transfer Rule Induction (TRI)

Inducing lexico-structural transfer rules:

I. **aligning nodes of the corresponding source and target parses**
II. generating transfer rule candidate
III. ordering transfer rule candidates (likelihood ratios)
IV. filtering transfer rule candidates
TRI - Aligning the Parse Nodes

- a new dynamic algorithm performing a **top-down, bidirectional beam search** for the least cost mapping between the nodes

- The algorithm is parameterized by the costs of:
  - aligning two nodes whose lexemes are not found in the BTD
  - aligning two nodes with differing parts of speech
  - deleting or inserting a node in the source or target tree
  - aligning two nodes whose relative locations differ

- Result: alignment id attributes (aid) are added to the nodes of the parse pairs
Figure 2: Transfer rule for English lexicalization and preposition insertion
@KOREAN:  
$X \ [\text{class=vbma ente=\{ra\}}]$

@ENGLISH:  
$X \ [\text{class=verb mood=imp}]$

$-2 \times \text{LOG}_\text{LIKENESS}: 33.37$

Figure 3: Transfer rule for imperative forms
Transfer Rule Induction (TRI)

Inducing lexico-structural transfer rules:

1. aligning nodes of the corresponding source and target parses
2. generating transfer rule candidate
3. ordering transfer rule candidates (likelihood ratios)
4. filtering transfer rule candidates
Candidate transfer rules are generated using three data sources:

A. the training set of aligned source and target parses
B. the alignment constraints
C. the attribute constraints
The **alignment** constraints:

- used to define most of the possible syntactic divergences between languages (Dorr, 1994)
- only a handful of them are necessary for two given languages
Figure 4: Alignment constraint

@KOREAN:
$X_1 [\text{aid}=$1] (  
   $R_1 \ \bar{X}_2 [\text{aid}=$2]  
)

@ENGLISH:
$Y_1 [\text{aid}=$1] (  
   $R_2 \ \bar{Y}_2 (  
      $R_3 \ \bar{Y}_3 [\text{aid}=$2]  
   )  
)

$X_1$ and $Y_1$ have the same attribute aid values and $X_2$ and $Y_3$ are aligned

Figure 2: Transfer rule for English lexicalization and preposition insertion

@KOREAN:
\{po\} \ [\text{class= vbma}] (  
   s1 \ $X [\text{ppca=} \{\text{reul}\}]  
)

@ENGLISH:
\text{look} \ [\text{class=verb}] (  
   \text{attr at} \ [\text{class=preposition}] (  
      ii \ $X  
   )  
)

@-2xLOG\_LIKELIHOOD: 12.77
The attribute constraints:

- limit the number of possible transfer rule candidates that can be generated from the sub-trees satisfying the alignment constraints

2 types of attribute constraints:

- independent - covers only one part of a candidate transfer rule
- concurrent - covers both parts of a candidate transfer rule
Independent Attribute Constraint

\[ \emptyset \, \text{x1} \, ( \, \text{R} \, \text{x2} \, ) \]
Concurrent Attribute Constraint

\[\begin{align*}
x & \quad [\text{aid} = \$1] \\
\text{and} & \\
y & \quad [\text{aid} = \$1] \\
\emptyset & \quad [\text{aid} = \$1] \\
\text{and} & \\
\emptyset & \quad [\text{aid} = \$1]
\end{align*}\]
Transfer Rule Induction (TRI)

Inducing lexico-structural transfer rules:

1. aligning nodes of the corresponding source and target parses using mostly the BTD
2. generating transfer rule candidate
3. **ordering transfer rule candidates (likelihood ratios)**
4. filtering transfer rule candidates
TRI - Ordering Transfer Rule Candidates

- by their log likelihood ratios (Manning and Schutze, 1999: 172-175)
- then by their specificity
  - the sum of: the number of attributes found in the source and target patterns, plus 1 for each lexeme attribute and for each dependency relationship
  - applied when two or more candidate transfer rules have the same log likelihood ratio
\[ \log \lambda = \log L(C_{12}, C_1, p) + \log L(C_2 - C_{12}, N - C_1, p) \\
- \log L(C_{12}, C_1, p_1) - \log L(C_2 - C_{12}, N - C_1, p_2) \]

where, not counting attributes \texttt{aid},

- \( C_1 \) = number of source parses containing at least one occurrence of C’s source pattern
- \( C_2 \) = number of target parses containing at least one occurrence of C’s target pattern
- \( C_{12} \) = number of source and target parse pairs containing at least one co-occurrence of C’s source pattern and C’s target pattern satisfying the alignment constraints
- \( N \) = number of source and target parse pairs
- \( P = C_2/N \)
- \( P_1 = C_{12}/C_1 \)
- \( P_2 = (C_2 - C_{12})/(N - C_1) \)
- \( L(k, n, x) = x^k(1 - x)^{n-k} \)
Transfer Rule Induction (TRI)

Inducing lexico-structural transfer rules:

1. aligning nodes of the corresponding source and target parses using mostly the BTD
2. generating transfer rule candidate
3. ordering transfer rule candidates (likelihood ratios)
4. filtering transfer rule candidates
TRI - Filtering Rule Candidates

A. selecting those rules that improve the baseline transfer dictionary
   a. applying these transfer rules to all the source structures
   b. calculating the difference between the resulting transferred structures and the target parses

B. testing each transfer rule candidate to see if it reduces the error rate
   a. adding each of the rules to the current set of accepted rules
   b. applying the updated set is to all the source structures
   c. the candidate is accepted and the error rate updated if

The difference between the transferred structures and the target parses < the current error rate
Figure 2: Transfer rule for English lexicalization and preposition insertion

Figure 3: Transfer rule for imperative forms

Figure 8: Transfer rule for Korean imperative with lexicalization and preposition insertion
TRI in Practice

- training set of 882 parse pairs (Data Preparation - Bi-text)
- produced 12467 source and target sub-tree pairs (TRI-generating)
- 20569 transfer rules candidate were generated (TRI-generating)
- 7565 rules were accepted after filtering (TRI-filtering)
Overview

1. Lavoie vs Meyers Approaches

2. Data Preparation
   - Bi-text
   - Baseline Transfer Dictionary

3. Transfer Rule Induction
   - Aligning Parse Nodes
   - Generating Transfer Rule Candidate
   - Ordering Transfer Rule Candidates
   - Filtering Transfer Rule Candidates

4. Evaluation
   - Tree Accuracy Recall
   - Tree Accuracy Precision
Evaluation

Task 1: applying both the baseline transfer dictionary and the induced transfer dictionary (BTD + rules from the training set) to the test half of the 1763 higher quality parse pairs

Result: producing two sets of transferred parses, the baseline set and the induced set

Task 2: calculating tree accuracy recall and precision measures
Evaluation - Tree Accuracy Recall

C/Rq

- C - the total number of features (attributes, lexemes and dependency relationships) in the nodes of target and transferred parses
- Rq - total number of features found in the nodes of the target parse
Evaluation - Tree Accuracy Precision

C/Rt

- C - the total number of features (attributes, lexemes and dependency relationships) in the nodes of target and transferred parses
- Rt - total number of features found in the nodes of the transferred parse
Almost 15% difference between Baseline TD and Inducted TD

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Precision</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>37.77</td>
<td>46.81</td>
<td>41.18</td>
</tr>
<tr>
<td>Induction</td>
<td>55.35</td>
<td>58.20</td>
<td>55.82</td>
</tr>
</tbody>
</table>

Table 1: Tree accuracy results
Learning Domain-Specific Transfer Rules: An Experiment with Korean to English Translation

Benoit Lavoie, Michael White, and Tanya Korelsky (2002)

Julia Dobczynska
Nazanin Khazaee Farid
Hybrid Machine Translation
University of Tuebingen
Overview

1. Approach

2. Data Preparation
   - Bi-text
   - Training and Test Sets of Parse Pairs
   - Baseline Transfer Dictionary

3. Transfer Rule Induction
   - Aligning Parse Nodes
   - Generating Transfer Rule Candidate
   - Ordering Transfer Rule Candidates
   - Filtering Transfer Rule Candidates

4. Evaluation
   - System Compared
   - Automatic Evaluation Results
   - Human Evaluation Results
Approach

Lavoie (2001): system that employs transfer rules induced from parsed bitexts

Lavoie (2002): system learns lexico-structural transfer rules for Korean to English translation
Approach

- Approach based on:
  - lexico-structural transfer (Nasr et. al., 1997), extends recent work reported in (Han et al., 2000)
  - syntactic approaches to EBMT such as (Nagao, 1984; Sato and Nagao, 1990; Maruyama and Watanabe, 1992).
  - the recent work of (Meyers et al., 1998)
Data Preparation - Parsing the Bi-text

- a parallel corpus **derived from bilingual training manuals** provided by the U.S
- corpus of a Korean dialog of 4183 sentences and their English human translations
- Parses obtained from:
  - Korean parser (Yoon et al., 1997)
  - **English Tree Bank developed in** (Han et al., 2000)
- realizer - RealPro (Lavoie and Rambow, 1997)
- output of the Yoon parser left as it is
Data Preparation - Training and Test Sets

Output: approx. half of the parse pairs contained incorrect dependency assignments, incomplete lemmatization or incomplete parses.

- selecting a higher quality subset of 1763 sentence pairs
  - source or target parse contained more than 11 nodes were rejected
  - parse pairs where the source or target parse contained non-final punctuation

- dividing this subset into training and test sets by randomly choosing 50% of the 1763 parse pairs
<table>
<thead>
<tr>
<th></th>
<th>Korean</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. sentence size</td>
<td>6.43</td>
<td>9.36</td>
</tr>
<tr>
<td>Avg. parse size</td>
<td>6.43</td>
<td>7.34</td>
</tr>
</tbody>
</table>

Table 2: Average sizes for sentences and parses in training set

<table>
<thead>
<tr>
<th></th>
<th>Korean</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. sentence size</td>
<td>7.04</td>
<td>10.58</td>
</tr>
<tr>
<td>Avg. parse size</td>
<td>7.02</td>
<td>8.56</td>
</tr>
</tbody>
</table>

Table 3: Average sizes for sentences and parses in test set
Data Preparation - Baseline Transfer Dictionary

Creating by combining three bilingual dictionaries:

1. **A corpus-based handcrafted dictionary** manually assembled by (Han et al., 2000)

2. **A corpus-based extracted dictionary** automatically created from Levoie’s corpus by the RALI group from the University of Montreal

3. **A wide coverage dictionary** (70,300 entries) created by Systran, without regard to our corpus
Data Preparation - Baseline Transfer Dictionary

1. replacing the inflected words with uninflected lexemes
2. merging all morphologically analyzed entries
3. matching the resulting transfer dictionary entries with the training parse set
4. creating a baseline dictionary using the instantiated rules
Transfer Rule Induction

1) Aligning the Parse Nodes
2) Generating Rule Candidates
   a) Alignment constraints
   b) Attribute constraints
3) Ordering Rule Candidates
4) Filtering Rule Candidates
5) Discussion of Induced Rules
Transfer Rule Induction

Discussion of Induced Rules

<table>
<thead>
<tr>
<th>Total number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source and target sub tree</td>
</tr>
<tr>
<td>pairs</td>
</tr>
<tr>
<td>Training set</td>
</tr>
<tr>
<td>Transfer candidate rules</td>
</tr>
<tr>
<td>Reduced candidate rules</td>
</tr>
<tr>
<td>Accepted transfer rules</td>
</tr>
<tr>
<td>candidates</td>
</tr>
<tr>
<td>22,881</td>
</tr>
<tr>
<td>1,433</td>
</tr>
<tr>
<td>801,674</td>
</tr>
<tr>
<td>32,877</td>
</tr>
<tr>
<td>2,133</td>
</tr>
</tbody>
</table>
Evaluation

- **Babelfish**
  - A commercial large coverage MT system supporting Korean to English translation.
  - This system was not trained on the corpus.

- **GIZA++/RW**
  - A statistical MT system, consisting of
    - ISI ReWrite Decoder (Germann et al., 2001)
    - Translation model produced by GIZA++ (Och and Ney, 2000)
    - Language model produced by the CMU Statistical Language Modeling Toolkit (Clarkson and Rosenfeld, 1997)
  - This system was trained on our corpus only

- **Lex Only**
- **Lex+Induced**
Automatic Evaluation Results

- used the Bleu metric from IBM (Papineni et al., 2001).

<table>
<thead>
<tr>
<th>System</th>
<th>1-g Prec</th>
<th>2-g Prec</th>
<th>3-g Prec</th>
<th>4-g Prec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babelfish</td>
<td>0.3814</td>
<td>0.1207</td>
<td>0.0467</td>
<td>0.0193</td>
</tr>
<tr>
<td>GIZA++/RW</td>
<td>0.1894</td>
<td>0.0173</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Lex Only</td>
<td>0.4234</td>
<td>0.1252</td>
<td>0.0450</td>
<td>0.0145</td>
</tr>
<tr>
<td>Lex+Induced</td>
<td>0.4725</td>
<td>0.1618</td>
<td>0.0577</td>
<td>0.0185</td>
</tr>
</tbody>
</table>

Table 5: Bleu N-gram precision scores
Automatic Evaluation Results

<table>
<thead>
<tr>
<th>System</th>
<th>Bleu Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babelfish</td>
<td>0.0802</td>
</tr>
<tr>
<td>GIZA++/RW</td>
<td>NA</td>
</tr>
<tr>
<td>Lex Only</td>
<td>0.0767</td>
</tr>
<tr>
<td>Lex+Induced</td>
<td>0.0950</td>
</tr>
</tbody>
</table>

Table 6: Bleu overall precision scores
## Human Evaluation Results

<table>
<thead>
<tr>
<th>System Pair Comparison</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Babelfish <em>better than</em> Lex Only</td>
<td>37%</td>
</tr>
<tr>
<td>Lex Only <em>better than</em> Babelfish</td>
<td>36%</td>
</tr>
<tr>
<td>Babelfish <em>same as</em> Lex Only</td>
<td>27%</td>
</tr>
<tr>
<td>Babelfish <em>better than</em> Lex+Induced</td>
<td>27%</td>
</tr>
<tr>
<td>Lex+Induced <em>better than</em> Babelfish</td>
<td>46%</td>
</tr>
<tr>
<td>Babelfish <em>same as</em> Lex+Induced</td>
<td>27%</td>
</tr>
<tr>
<td>Lex Only <em>better than</em> Lex+Induced</td>
<td>18%</td>
</tr>
<tr>
<td>Lex+Induced <em>better than</em> Lex Only</td>
<td>41%</td>
</tr>
<tr>
<td>Lex Only <em>same as</em> Lex+Induced</td>
<td>41%</td>
</tr>
</tbody>
</table>

Table 7: Human evaluation results
An Integrated Architecture for Example-Based Machine Translation

Alexander Franz, Keiko Horiguchi, Lei Duan, Doris Ecker, Eugene Koontz, and Kazami Uchida 2000
Overview

1. Domain
2. NLP Infrastructure
   a. The Grammar Programming Language
   b. The GPL Compiler
   c. The GPL Run-time Environment
3. Source Language Analysis
4. Transfer
5. Target-language Generation
6. Evaluation and Conclusions
"example-based translation" Approach was first proposed by (Nagao 1984).

Recent work includes:

- memory-based translation (Sate & Nagao 1990)
- similarity-driven translation (Watanabe 1992)
- transfer-driven machine translation (Furusc & Iida 1996)
- pattern-based machine translation (Watanabe & Takeda 1998)
Benefits:

Example-based approach

- Promises easy translation knowledge acquisition
- More flexible transfer than brittle rule-based approaches

Rule-based approach

- Detailed linguistic analysis can allow an example-based machine translation system to handle a wide variety of input
- Rules can be used to factor out all linguistic variations that do not influence the exampled based transfer
- Rule-based language generation from detailed linguistic representations can lead to higher grammatical output quality
- Modular system architecture that uses domain-independent linguistic regularities in separate linguistic modules allows extending the system to much broader domains.
Domain

Prototype implementation architecture was designed to cover the "travel domain".

Principles guided definition of the translation domain

- The translation domain should not be limited to a narrow sub-domain
- The expressions considered in the domain should reflect the fact that people quickly adapt to limitations in human-machine or machine-mediated communication by simplifying the input
Domain

Vocabulary division

- General words
- Extensible word groups
- Area-specific word groups

Situation division

- General situation
- Transportation
- Accommodation
- Sightseeing
- Shopping
- Wining
- Dining, and nightlife
- Banking, postal, doctor and pharmacy
Domain

Corpus

- Development set of 7,000 expressions
  - used for creation and refinement of the translation knowledge sources
- Test set of 5,000 expressions
  - used for evaluations

The average length of the expressions in the corpus is 6.5 words
Domain

Corpus examples

- Can I have your last name, please?
- Is this the bus for Shinagawa station?
- I would like to make a reservation for two people for eight nights.
- Can you tell us where we can see some Buddhist temples?
- Most supermarkets sell liquor.
- Can you recommend a good Chinese restaurant in this area?
- I'd like to change 500 Dollars' in traveler’s checks into Yen.
- Are there any English-speaking doctors at the hospital?
Infrastructure

Main aspects

- Grammar Programming Language (GPL)
- GPL compiler
- GPL runtime environment
The Grammar Programming Language

```
WH_SENT → NP YN_SENT {
    exist[$m VP SUBJ WH];
    local-variable WH_VP = [$m VP];
    local-variable WH PHRASE;
    $WH PHRASE = find-substructure in $WH VP
    where (?exist[$x WH]);
    $d1 = [$WH PHRASE SLOT-VALUE];
    [$WH PHRASE SLOT-VALUE TRACE] = ‘+’;
    $d2 = $m;
}
```

Figure 1  Example of a GPL Generation Rule
The GPL Compiler

- Compile GPL grammars into C code
- Handles disjunctive feature structures
- Tracks variable references
- Generates and tracks separate test functions for nested test expressions.
GPL Run-time Environment

Figure 2  GPL Run-time Environment
Source Language Analysis

Translation steps:

- Analysis
- Transfer
- Generation

Figure 3 Architecture of the Analysis Module
Transfer

Figure 4  Architecture of the Transfer Module
Target-language Generation

Figure 5  Architecture of the Generation Module
Evaluation and Conclusions

Translation quality

- **Failure**
  - Complete translation failure, due to lack of coverage of a rule-based component.
- **Wrong**
  - A translation that is completely wrong, or that has major errors in an important part, such as in the main clause.
- **Major Problem**
  - A translation that has a missing, extra, or incorrect constituent.
- **Minor Problem**
  - A translation that has a missing, extra, or incorrect minor part.
- **Stylistic Problem**
  - Stylistic Problem include awkward but tolerable word order.
- **Flawless**
  - A translation that does not exhibit any of the above problems is considered flawless.
Evaluation and Conclusions

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flawless</td>
<td>60%</td>
</tr>
<tr>
<td>Stylistic Problem</td>
<td>9%</td>
</tr>
<tr>
<td>Minor Problem</td>
<td>14%</td>
</tr>
<tr>
<td>Acceptable with OOV</td>
<td>1%</td>
</tr>
<tr>
<td>Major Problem</td>
<td>9%</td>
</tr>
<tr>
<td>Wrong Translation</td>
<td>5%</td>
</tr>
<tr>
<td>Translation Failure</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 1 Translation Quality

<table>
<thead>
<tr>
<th>Memory Requirement</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read-only Memory for Code and Data</td>
<td>6MB</td>
</tr>
<tr>
<td>Read-only Memory for Dictionary, Examples, Fast Match Index, etc.</td>
<td>23MB</td>
</tr>
<tr>
<td>Read/Write Memory for Feature Structures</td>
<td>14MB</td>
</tr>
<tr>
<td>Read/Write Memory for Software Engines</td>
<td>4MB</td>
</tr>
</tbody>
</table>

Table 2 Memory Requirements