Why Corpus-Based Statistics-Oriented Machine Translation

Keh-Yih Su and Jing-Shin Chang

Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, TAIWAN 30043, R.O.C.
email: kysu@ee.nthu.edu.tw, shin@ee.nthu.edu.tw

Contents

- Designing Goals for Practical MT
- Methodological Issues in Large-Scale MT Design
- Motivation for Corpus-Based Statistics-Oriented MT
- Necessity, Feasibility of CBSO MT
- CBSO Approaches in ArchTran MT System
- Conclusion

Designing Goals for Practical MT

- Quality for unseen text
  - Coverage: wide coverage for practically encountered phenomena even if they are not theoretically interested
  - Robustness: show good performance both in laboratory and real environment (avoid over-tuning)
- Meeting user requirements: customers want HQMT for publication in their styles
  - Adaptable: can be adapted to user's domain
  - Controllable: can be controlled by feedback from customers
- Capacity of performance tuning: should have a systematic and controllable way to improve the system performance (can be parameterized)
- Cost-effectiveness
  - knowledge base construction
  - maintenance of large knowledge base
  - Need a system which can
    - shift the burden of knowledge acquisition from human to computer
    - handle messy and uncertainty phenomena

Why MT Systems are not Widely Used Today

- Designers' expectation is different from customers' expectation
  - designers: readable, understandable
  - customers: publishable (HQ MT) in the customer's preferred style
- Customer feedback can not be integrated into the system as a designing feature
  - because the system is not parameterized
- One-way designing approach inherited from stratificational paradigm
  - intermediate representations, transfer and generation are tightly bound to the source text
  - designer tends to improve the system from the source side (analysis $\rightarrow$ transfer $\rightarrow$ generation)
  - user feedback could not be fed to the system from the target side

☐ Need a parameterized feedback controllable system

☐ Need the capability of bidirectional training for the translation mechanism

Some Observations in MT Research

- Required knowledge is huge, messy and fine-grained
  - Knowledge acquisition by hand is very expensive and time-consuming
- Human is competent in abstract language modeling, but awkward in dealing with large and fine-grained knowledge consistently
  - It is not easy for people to maintain consistency of large rule bases
  - Tuning rule bases by hand does not guarantee global improvement
- Computers are good at processing large corpora, but do not have modeling capability by themselves
  - appropriate for acquiring and learning parameters of language models
- Most knowledge required in MT is inductive, not deductive
  - Language $\rightarrow$ Linguistics
  - Linguistics $-\times\rightarrow$ Language
- Inductive knowledge can be acquired easily with statistical methods
  - statistical models are appropriate for consistently modeling complex phenomena in large corpora

Rule-Based Approaches

- Have a strict sense of well-formedness in mind
- Impose linguistic constraints on syntactic or semantic constructs to satisfy well-formedness
- Mostly based on linguistic knowledge that is linguistically interesting or *ad hoc* heuristics

- Example: (lexical disambiguation)
  - Heuristics: a “determiner” can not be followed by a “verb”
  - Rule: if $C_{i-1} =$ “det” then $C_i \neq$ “verb”

- Advantages
  - easy to incorporate existing linguistic knowledge
  - have better generalization (deduction) capacity
    - use a small number of concepts to describe linguistic behavior

Purely Statistical Approaches

- No strict sense of well-formedness in mind
- Assume language generation to be a Markov chain
- Treat translation as a decoding process, no traditional linguistic knowledge is used in modeling
- mostly acquire parameters from surface strings

- Example I: (lexical disambiguation)
  - to discover the fact that p(v|det) = 0, one need to find p(y|x) = 0 for all x $\in$ "det" and all y $\in$ "verb"

x	y	p(y|x)
the	ate	0
a	took	0
an	walks	0
	buy	0

- Advantages
  - do not need to establish linguistic models
  - computation is easy

Corpus-Based Statistics-Oriented (CBSO) Approaches

- Establish a language model based on known linguistic knowledge
- Prepare a large corpus (possibly annotated) in order to acquire consistent knowledge implied in the corpus
- Acquire statistical "rules" from the corpus through statistical induction mechanism, namely, to learn model parameters automatically (or semi-automatically)

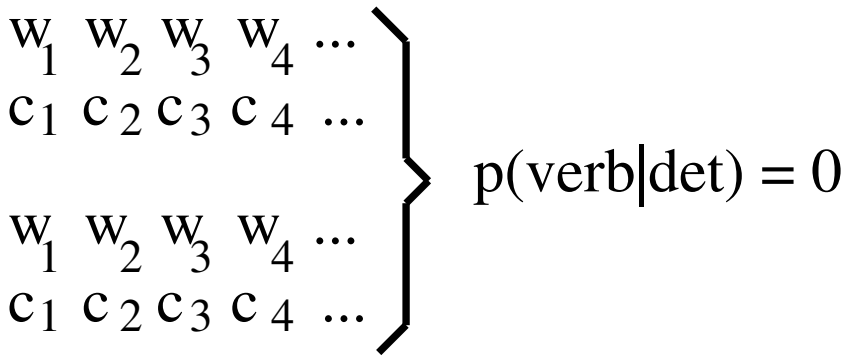
- Example: (Analysis Score or Score Function for Disambiguation)

$$
\begin{aligned}
& P(Sem, Syn, Lex \mid Words) \\
& \quad = P(Sem \mid Syn, Lex, Words) \\
& \quad \times P(Syn \mid Lex, Words) \\
& \quad \times P(Lex \mid Words) \\
& \quad \equiv S_{sem} \times S_{syn} \times S_{lex}
\end{aligned}
$$

CBSO: Characteristics of the Techniques

- CBSO = Statistical Language Model + Language Parameters
- Use corpus as the only information source
  - No heuristic rule is used
  - All knowledge is obtained from corpus
    - Except for modeling
- Use statistical method to acquire knowledge
  - Knowledge is interpreted in statistical sense
  - Parameters are automatically learned from corpus
- No strict sense of well-formedness in mind
- Make use of available linguistic knowledge to construct high-level stochastic language models
- Assume a language generation process has stochastic behavior when a proper language model is used
- A promising paradigm for integrating linguistic and statistic knowledge

- Example: (lexical disambiguation)
  - use the known lexical knowledge to reduce the parameter space and the required training data drastically



A Comparative Review

- Rule-Based Approaches
  - Heuristic Rule: a “determiner” can not be followed by a “verb”
- Purely Statistical Approach

x	y	p(y|x)
the	ate	0
a	took	0
an	walks	0
	buy	0

  - Conclusion: a word in x can not be followed by a word in y
- CBSO Approach

$$\left.\begin{array}{l} w_1\ w_2\ w_3\ w_4\ \ldots \\ c_1\ c_2\ c_3\ c_4\ \ldots \\ \\ w_1\ w_2\ w_3\ w_4\ \ldots \\ c_1\ c_2\ c_3\ c_4\ \ldots \end{array}\right\} \; p(\text{verb}|\text{det}) = 0$$

Why Not Rule-Based MT

- hard to handle uncertainty and ill-formedness (lack of objective preference measure)
- hard to deal with complex and irregular knowledge (exception)
- hard to maintain consistency among different persons at different time
- poor coverage
  - try to deal with the large complex system by increasing modeling complexity
  - add more and more *ad hoc* heuristics when exception occurs and tend to over-tune the system
  - ☐ usually fail to attain the required coverage due to robustness problem when an unseen text is encountered
- lack of systematic and automatic ways for learning rules in large-scale applications
- knowledge acquisition is costly and time consuming
- local improvement
  - ☐ hard to scale-up and guarantee global improvement
  - ☐ unpredictable global performance improvement or degradation when local improvement is tried

Statistical *vs.* Rule-Based Approaches

- Disadvantages
  - hard to couple with existing linguistic theories
  - generalization (deduction) capability is poor with small database
- Advantages
  - uncertainty or preference is interpreted objectively and consistently
  - consistency can be easily maintained even in large scale systems
  - well-established statistical theories and techniques are available
  - the system parameters can be manipulated in a language-independent manner
  - semi-/automatic training is possible with least human intervention
  - remove the burden of rule induction from linguists to machine

Why Not Purely Statistical Approaches

- the parameter space is too large
  - a very large corpus is required to get reliable statistics due to the large parameter space
- can only deal with local dependency
  - simplified models which consult less contextual information are used to make the parameter space manageable
- suffer from robustness and sparse data problems

Why Corpus-Based Statistics-Oriented MT

- inhere the good properties of objectiveness, consistency, trainability, cost-effectiveness for statistical approaches
- easy to meet the desirable designing goals of wide coverage, robustness, adaptability, controllability, parameterization and cost-effectiveness
- long distance dependency is manageable
  - more contextual information can be used
  - syntactic and semantic information could be consulted
- sparse data problem is less severe
  - intermediate forms are introduced to reduce the parameter space
  - parameter space is small with respect to purely statistical approaches

Necessity and Feasibility of Corpus-Based Approaches

- Necessity
  - The demand of large scale practical MT is becoming increasingly important
  - Uncertainty, preference, and huge amount of fine-grained knowledge must be acquired from real text
  - Human intervention is costly and time consuming
- Feasibility
  - Large machine readable corpora are available
  - Better statistical language models are being constructed
  - Cheap and fast computation power can easily be acquired

Main Tasks in Corpus-Based Statistics-Oriented Approaches

- Language Modeling
  - Feature Selection
  - Linguistic Problem Abstraction
  - Mathematical Formulation
- Corpus Preparation
  - Annotation with required information
- Parameter Estimation
  - Sparse Data Problem
- Adaptive Learning
  - Discrimination Issues
  - Robustness Issues

ArchTran Designing Philosophy

- Cooperative approach
  - Admit the facts that
    - general linguistic knowledge has better generalization for unseen domain
    - uncertain knowledge can be more objectively qualified by statistic models
    - large and fine-grained language parameters can be acquired more cost-effectively by machine
    - human is competent in language modeling while machines are suitable for massive data processing
  - Principles of cooperative approach
    - take advantages of well-recognized linguistic phenomena
    - setup probabilistic language model by humans
    - acquire language parameters from corpora (semi-)automatically
- Integrated approach
  - various levels of linguistic knowledge must be integrated to give best judgement
  - knowledge should be integrated under a uniform formulation & control

- Systematic approach
  - knowledge acquisition & integration should be done with systematic approach
  - least human intervention should be involved via parameter learning

Corpus-Based Research in ArchTran MT Group

- Score Function
  - Lexical Score [Su 91c]
  - Syntactic Score [Su 88, 89a, 90d]
  - Semantic Score [Liu 89, 90, Chang 90, 92]
- Improved Parameter Estimation
  - Smoothing Techniques [Su 89b]
  - Bootstrapping Techniques
  - Robustness Approaches [Su 91b]
- Adaptive Learning Algorithm for Statistics Database Refinement [Su 90b, 91b]
- Scored Truncation [Su 87b, 89a, 90a]
  - use Score Function to cut down searching space
- Probabilistic Transfer and Generation Model [Chang 91]
- Lexicon Acquisition
- Compound Noun Identification
- Grammar Acquisition [Su 91d]
- Performance Evaluation with Distance Measure [Wu 92]
- Parameterized and Feedback-Controlled Model for MT Tuning [Wu 92]
- Spoken Language Processing [Chiang 89, Su 90b, 91a]

- Advanced Topics
  - Language Parameter Re-estimation from Corpora
  - Domain Identification
  - Domain Adaptation
- Goal: design systematic and automatic approaches for acquiring language model parameters
- Achievement: established integrated models for preference measure and ambiguity resolution in various levels of NLP/SLP
  - phonetic, lexical, syntactic & semantic

Probabilistic Translation Model

- Translation problem is an optimization problem
- Let
  - $CW_1^{n_C}$: Chinese sentence of length n_C
  - $EW_1^{n_E}$: English sentence of length n_E
  - *CLM*: Chinese Language Model
  - *TM*: Transfer Model from English to Chinese
  - *ELM*: English Language Model
  - I_C: Intermediate Representation of Chinese (i.e., normalized Chinese annotated syntax tree)
  - I_E: Intermediate Representation of English (i.e., normalized English annotated syntax tree)

- Find the mapping from English to Chinese is equivalent to finding $CW_1^{n_C}$ that maximizes the *translation score*:

$$
\begin{aligned}
& P\left(CW_1^{n_C} \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&= \sum_{I_E} \sum_{I_C} P\left(CW_1^{n_C}, I_C, I_E \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&= \sum \sum P\left(CW_1^{n_C} \mid I_C, I_E, CLM, \cdots\right) \\
&\quad \times P\left(I_C \mid I_E, TM, EW_1^{n_E}, \cdots\right) \\
&\quad \times P\left(I_E \mid EW_1^{n_E}, ELM, TM, CLM\right) \\
&\approx \sum \sum P\left(CW_1^{n_C} \mid I_C, CLM\right) && (generation\ score : P_G) \\
&\quad \times P\left(I_C \mid I_E, TM\right) && (transfer\ score : P_T) \\
&\quad \times P\left(I_E \mid EW_1^{n_E}, ELM\right) && (analysis\ score : P_A) \\
&= \sum \sum P_G(.) \times P_T(.) \times P_A(.)
\end{aligned}
$$

- Searching Strategy: to find the desirable candidates of $CW_1^{n_C}$, we can achieve our goal in three separate phases.
  - Analysis: find the intermediate representation of English sentence that maximizes the *analysis score*: $P_A(\cdot)$
  - $I_{Emax} \stackrel{\Delta}{=} \arg\max_{I_E} P\left(I_E \mid EW_1^{n_E}, ELM\right)$
  - Transfer: find the intermediate representation of Chinese sentence that maximizes the *transfer score*: $P_T(\cdot)$
  - $I_{Cmax} \stackrel{\Delta}{=} \arg\max_{I_C} P\left(I_C \mid I_{E_{max}}, TM\right)$
  - Generation: find the Chinese sentence that maximizes the *generation score*: $P_G(\cdot)$
  - $CW_{max} \stackrel{\Delta}{=} \arg\max_{CW_1^{n_C}} P\left(CW_1^{n_C} \mid I_{C_{max}}, CLM\right)$
- In each phase, the Best-N candidates can be selected for further processing.

☐ The introduction of intermediate forms is a general technique to make a purely statistical approach become CBSO.

☐ The probabilistic translation model provides a standard and systematic method to improve the system in a predictable way.

Score Function

- A realization of the analysis score for general ambiguity resolution problem.
- Score Function as preference measure:

$$Score\,(Sem, Syn, Lex, Words) \equiv P\,(Sem, Syn, Lex \mid Words)$$

  - Words: Input Tokens
  - Lex: Lexical Category
  - Syn: Syntactic Structure
  - Sem: Semantic Annotation

- Decomposed form for stratified analysis

$$\begin{aligned} &P\,(Sem, Syn, Lex \mid Words) \\ &= P\,(Sem \mid Syn, Lex, Words) \\ &\times P\,(Syn \mid Lex, Words) \\ &\times P\,(Lex \mid Words) \\ &\equiv S_{sem} \times S_{syn} \times S_{lex} \end{aligned}$$

  - S_{sem}: Semantic Score
  - S_{syn}: Syntactic Score
  - S_{lex}: Lexical Score

- Why score function ?
  - Provide objective preference measure for
    - Scored Parsing
    - Ambiguity Resolution
    - Ill-formedness Recovery
  - Optimal in Bayesian sense
  - Minimal cost of misjudgement under certain cost function
  - Integration of various level of knowledge in a uniform statistical framework
  - Easy to extend to other levels in a stratified analysis hierarchy
  - Easy to couple with spoken language processing

Lexical Score & Lexicon Tagging

- Lexical Score:

$$S_{lex} \equiv P\left(Lex \mid Words\right) \equiv P\left(c_1^n \mid w_1^n\right)$$
$$= \prod P\left(c_i \mid c_1^{i-1}, w_1^n\right)$$

- Various reduced forms
  - [Garside, et. al, 87]:

$$P\left(c_i \mid c_1^{i-1}, w_1^n\right) \approx P\left(c_i \mid c_{i-1}\right) \times P\left(c_i \mid w_i\right)$$

  - [Church 88]:

$$P\left(c_i \mid c_1^{i-1}, w_1^n\right) \approx P\left(c_i \mid c_{i-2}^{i-1}\right) \times P\left(c_i \mid w_i\right)$$

- ArchTran Formulation [Su, et. al, 91c]:
  - Generalized nonlinear smooth form

$$g\left(P\left(c_i \mid c_1^{i-1}, w_1^n\right)\right)$$
$$\equiv \lambda g\left(P\left(c_i \mid w_i\right)\right) + \left(1-\lambda\right) g\left(P\left(c_i \mid c_{i-k}^{i-1}\right)\right)$$

  - λ: lexical weight for lexical probability database
  - *k*: context window size
  - g: transform function (Log function)

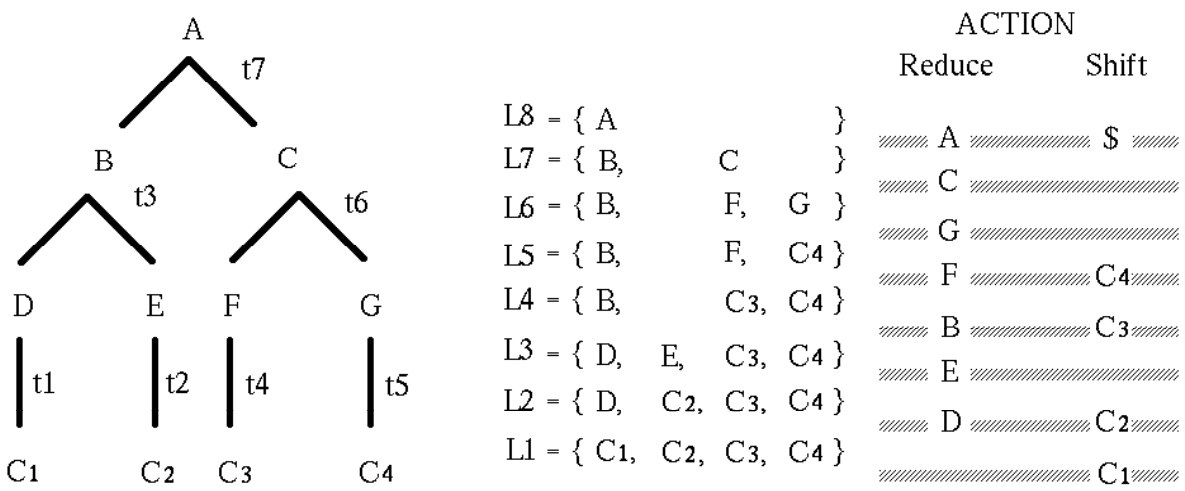
- Why generalized smooth form
  - too many parameters to estimate in unreduced form
  - $P\left(c_i \mid c_1^{i-1}, w_1^n\right)$ can not be approximated with the product $P\left(c_i \mid c_{i-k}^{i-1}\right) \times P\left(c_i \mid w_i\right)$ unless a number of strong independency assumptions are satisfied
  - instead, we can regard the two factors as two parameters estimated from two databases of different degree of reliability
  - λ gives the weigh to the two databases
  - the nonlinear formulation reduces to [Garside 87] or [Church 88] when λ=1/2
- Experimental Results [Su, et. al, 91c]
  - Nonlinearly smooth form with λ = 0.6 is the best for Brown Corpus (0.7 for BTC corpus), not 1/2.

Syntactic Score & Structure Disambiguation

- Syntactic Score [Su 88]

$$S_{syn} \equiv P\,(Syn \mid Lex, Words)$$

- Why not stochastic context-free grammar?
  - context-sensitivity is not encoded
  - correlation between the use of rules is not considered
  - normalization problem will be introduced (applying a large number of rules tends to reduce score)
- Formulation of syntactic score
  - decompose a syntax tree into phrase levels
    - for LR parsers, a phrase level corresponds to a sentential form acquired through rightmost derivation
  - formulate the transition between phrase levels as *context-sensitive* rewriting process
  - approximate the transition probabilities by considering finite length window of left/right context
  - compact highly correlated phrase levels when evaluating syntactic score



Decomposition of a syntax tree into phrase levels
for score computation in a bottom–up GLR parser.

– Basic Context-Sensitive Formulation

$$
\begin{aligned}
& P\left(L_j | L_1^{j-1}, c_1^n, w_1^n\right) \\
& \quad \approx P\left(L_j | L_{j-1}\right) \\
& \quad \approx P\left(\{l_A, A, r_A\} \,|\, \{l_A, X_1, X_2 \ldots X_m, r_A\}\right)
\end{aligned}
$$

- L_j: the jth phrase level
- $L_j : \{l_A A\, r_A\} \leftarrow L_{j-1} : \{l_A X_1 X_2 \cdots X_m r_A\}$
- encode context-sensitivity within context-free framework
- evaluated after each **reduce** action

– Example:

$$
\begin{aligned}
& SCORE_{syn}\left(Syn_A\right) \\
& \quad = P\left(L_{8,} L_{7 \ldots} L_2 | L_1\right) \\
& \quad = P\left(L_8 | L_{7 \ldots} L_2, L_1\right) \times P\left(L_7 | L_{6 \ldots} L_1\right) \times \ldots \times P\left(L_2 | L_1\right) \\
& \quad \approx P\left(L_8 | L_7\right) \times P\left(L_7 | L_6\right) \times \ldots \times P\left(L_2 | L_1\right) \\
& \quad \approx P\left(\{A\} | \{l_7, B, C, r_7\}\right) \times P\left(\{C\} | \{l_6, F, G, r_6\}\right) \times \ldots \times P\left(\{D\} | \{l_{1,} c_1, r_1\}\right)
\end{aligned}
$$

– Run-Time Formulation (Normalization Form)
  - compact multiple highly correlated phrase levels when evaluating score
  - evaluated after each **shift** action
  - avoid “normalization problem” (same input with different number of transition probabilities)

– Example:

$$
\begin{aligned}
&SCORE_{syn}(Syn_A) \\
&\quad = P\left(L_{8,}L_{7\ldots}L_2|L_1\right) \\
&\quad = P\left(L_{8,}L_{7,}L_6|L_{5,}L_{4\ldots}L_1\right) \times P\left(L_5|L_{4,}L_{3\ldots}L_1\right) \times P\left(L_{4,}L_3|L_{2,}L_1\right) \times P\left(L_2|L_1\right) \\
&\quad \approx P\left(L_{8,}L_{7,}L_6|L_5\right) \times P\left(L_5|L_4\right) \times P\left(L_{4,}L_3|L_2\right) \times P\left(L_2|L_1\right) \\
&\quad \approx P(L_8|L_5) \times P(L_5|L_4) \times P(L_4|L_2) \times P(L_2|L_1)
\end{aligned}
$$

Semantic Score & Generalized Approach for Ambiguity Resolution

- Semantic Score

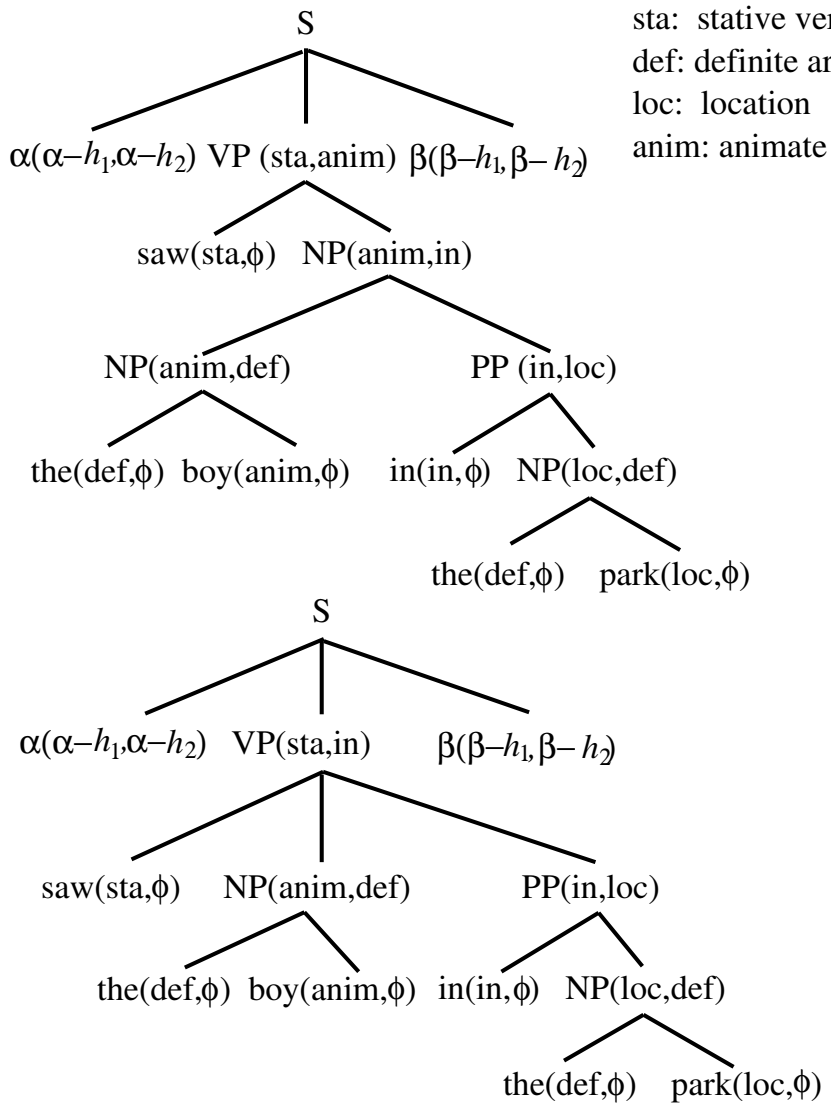
$$
\begin{aligned}
S_{sem} &\equiv P\left(Sem \mid Syn, Lex, Words\right) \\
&\equiv P\left(\Gamma_1^m \mid L_1^m, c_1^n, w_1^n\right) \\
&= \prod_i P\left(\Gamma_i | \Gamma_1^{i-1}, L_1^m, c_1^n, w_1^n\right) \\
&\approx \prod_i P\left(\Gamma_i | \Gamma_{i-1}\right) \\
&\approx \prod_i P\left(\left\{\bar{l}_A, A\left(S_1^N\right), \bar{r}_A\right\} \mid \left\{\bar{l}_A, \bar{X}_1, \bar{X}_2 \ldots \bar{X}_M, \bar{r}_A\right\}\right)
\end{aligned}
$$

- Formulation
  - Γ_i (annotated phrase level): the semantic information is encoded as a form of annotation to the phrase levels
  - the annotation is the feature structure associated with each constituent in the phrase level.
  - a semantic N-tuple (S_1, S_2, ..., S_N) is used as the feature structure of constituent A.
  - S_j is the "head feature" (the 1st feature) of the "jth head" of the current node A.

- Major categories
  - constituents which carry little semantic information (e.g., function words) are not used for disambiguation in order to reduce the parameter space
  - only verbs (V), nouns (N), prepositions (P), adjectives (A) and adverbs (ADV) are used to encode the semantic information of a node
- Encoding Scheme
  - each phrase structure rule is associated with a "head list" which ranks the children nodes in decreasing degree of importance to the compositional semantics of the mother node
  - a permutation vector can be used to specify the list of heads of the mother node
- Specialized model for PP attachment disambiguation [Liu 89, 90]

$$\max_{X \in \{V,N\}} P\left(X - PP \mid [v, n_1, p, n_2]\right)$$

- Generalized Probabilistic Semantic Model for general problems of ambiguity resolution [Chang 90]



sta: stative verb
def: definite article
loc: location
anim: animate

The VP subtrees show the discriminative characteristics of the semantic 2–tuple encoding scheme for PP attachment problem.

Probabilistic Transfer and Generation Models

- Simplified translation model

$$\begin{aligned} S_{SYS} &\equiv P(t, I_t, I_s \mid s) \\ &= P(t \mid I_t, I_s, s) \times P(I_t \mid I_s, s) \times P(I_s \mid s) \\ &\equiv S_{gen} \times S_{txf} \times S_{ana} \end{aligned}$$

- Transfer Score

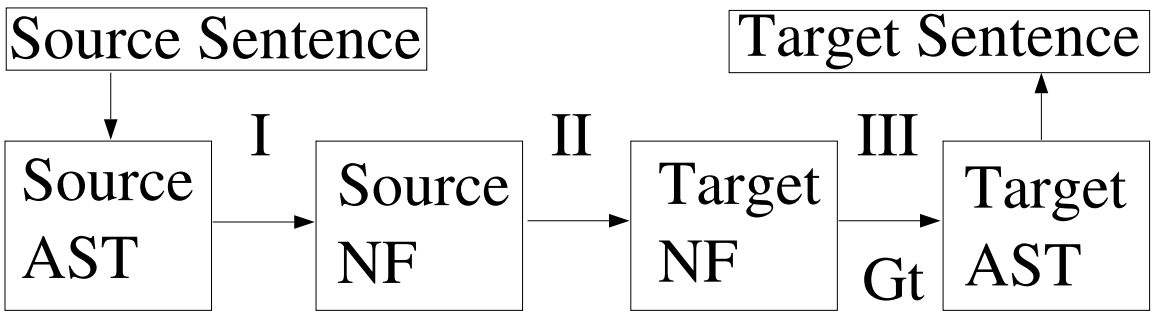
$$S_{txf} \equiv P(I_t \mid I_s, s) \approx P(T_t \mid T_s) \approx P\left(\hat{T}_t \mid \hat{T}_s\right)$$
$$= \prod_i P\left(p_j\left([s_s, h_s, a_s, m_s]_i\right) \mid [s_s, h_s, a_s, m_s]_i\right)$$

  - *s (t)*: source (target) sentence
  - I_s *(*I_t*)*: Intermediate representation of source (target)
  - T: annotated syntax tree (AST)
  - $\hat{T}$: normalized AST (NAST)
  - p_j: *j*th permutation

- Any localized transfer units can be used to encode the normalized AST
- e.g., $\hat{T} \approx$ {(s, h, a, m) : (specifier, head, argument, modifier) 4–tuple}
- Generation Score

$$S_{gen} \equiv P(t \mid T_t, T_s, s) \approx P\left(t \mid \hat{T}_t, s\right)$$
$$= \prod_i P\left([s_t, h_t, a_t, m_t]_i \mid p_j\left([\cdot]_i\right), [s_s, h_s, a_s, m_s]_i\right)$$

Transfer Scheme



I : Source Normalization
II : Transfer
III : Target Generation

Gt : Target Grammar
AST : Annotated Syntax Tree
NF : Normal Form

A conceptual model for the transfer and generation phases

- AST: syntax tree with semantic annotations
- Source Normalization: reduce analysis results to a normal form (NF) which consists of only atomic transfer units
  - only local operations are needed to accomplish transfer/generation phase
  - make transfer more easy and robust
  - map various syntactic variants into the same deep structure to reduce parameter space
- Normal Form Transfer: map source NF to target NF
  - bridge the conceptual gaps and stylistic gaps between languages

- Target Generation: use target NF to construct the target AST and the target sentence according to the generation grammar (Gt)

Transfer Score

$$S_{txf} = P\left(T_t \mid T_s\right)$$

- T_t, T_s are the target and source AST's. (equivalent to I_C, I_E in "Translation Score")
- the target AST that maximizes the transfer score, given the source AST, is preferred

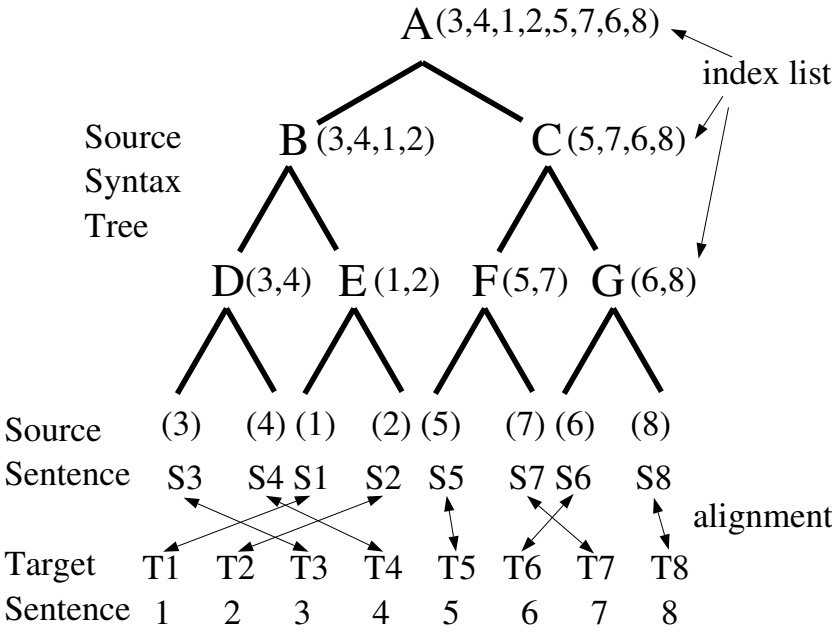
- Decomposition: if each AST can be reduced to a normalized version that can be decomposed into a set of locally transferable units (LTU's) ...
  - transfer and generation process can be accomplished by using only local permutation for reordering, insertion or deletion of target-specific morphemes to the LTU's
- Decomposition of Transfer Score in terms of LTU's:

$$\begin{aligned} S_{txf} &= P\left(T_t \mid T_s\right) \\ &\approx P\left(\hat{T}_t \mid \hat{T}_s\right) \\ &= P\left(X'_1, X'_2, \cdots, X'_n \mid X_1, X_2, \cdots, X_n\right) \\ &= \prod_{i=1}^{n} P\left(X'_i\left(h', a', s', m'\right) \mid X_i\left(h, a, s, m\right)\right) \end{aligned}$$

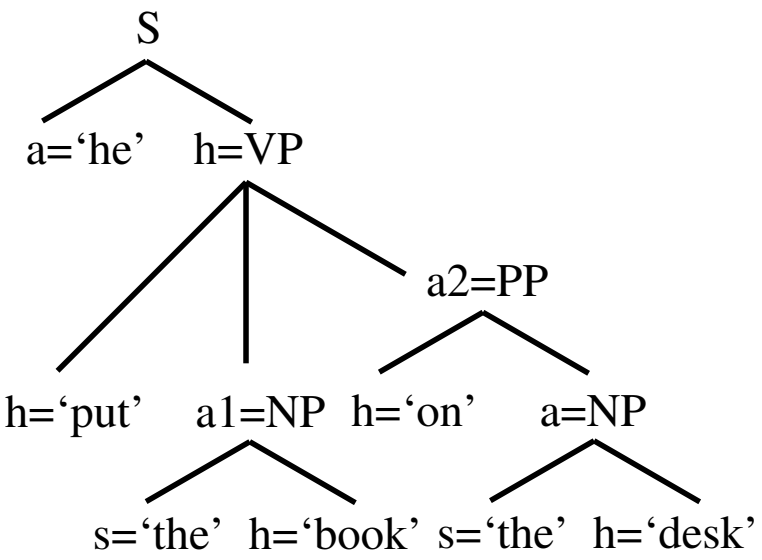
  - $\hat{T}_t$, $\hat{T}_s$: source and target normal forms
  - X_i, X'_i: source and target LTU's
  - assume that X'_i depends on X_i only

- LTU:
  - any appropriate substructure that satisfies the local transfer properties can be used as an LTU
  - an LTU is assumed to consist of the syntactic category (Xi) for the unit and its head (h), arguments (a), specifiers (s) and modifiers (m) 4-tuples (and the associated features attached to these slots) in the above
- Expressive power of the formulation
  - the syntactic category of a source LTU (X_i) can be changed after transformation to the target LTU (X_i')
  - the elements of the 4-tuple are allowed to change their syntactic or semantic roles after transfer
  - multiple arguments, specifiers and modifiers are possible for a given head

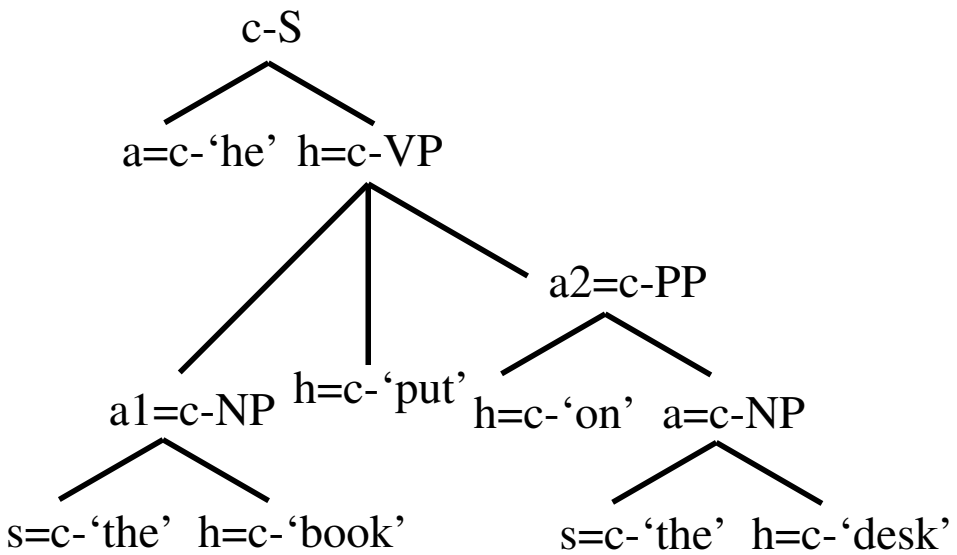
Finding LTU from syntax tree and word index list.



Examples of Normal Form



The normal form in terms of the (head, arguments, specifiers, modifiers) 4–tuple for the sentence ‘he put the book on the desk’.



The equivalent normal form for target (Chinese). (c-X: the Chinese equivalent of ’X’)

Generation Score

$$
\begin{aligned}
S_{gen} &\approx P\left(t \mid \hat{T}_t\right) = P\left(t_1, t_2, \cdots, t_n \mid X'_1, X'_2, \cdots, X'_n\right) \\
&= \prod_i P\left(f\left(X'_i\right), p\left(X'_i\right) \mid X'_i\left(h', a', s', m'\right)\right) \\
&= \prod_i P\left(p\left(X'_i\right) \mid X'_i\left(h', a', s', m'\right)\right) \quad (struct.\ transfer/generation) \\
&\times P\left(f\left(X'_i\right) \mid p\left(X'_i\right), X'_i\left(h', a', s', m'\right)\right) \quad (morph.\ generation)
\end{aligned}
$$

- *t*: target sentence ($Cw_1^{n_C}$)
- t_i: i-th sentence segment generated from X'_i
- $p\left(X'_i\right)$: permutation function
  - a permuted list of the 4–tuple
- $f\left(X'_i\right)$: (morphological) generation frame for the LTU 4–tuple of X'_i
  - specifies which target-specific morphemes are to be added to the particular positions in between the elements of the 4–tuple

- Example (generation):

VP(h=put,a1=the-book,a2=on-the-desk,s=NIL,m=NIL)

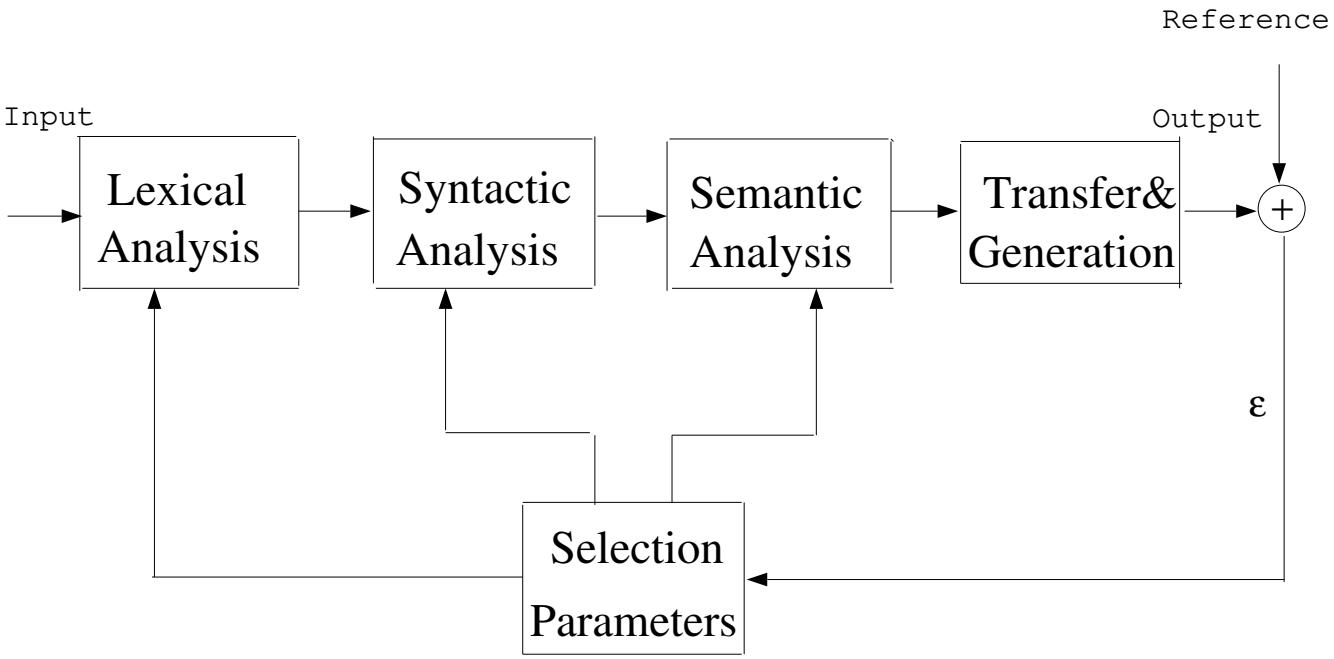
p(h,a1,a2,s,m)=(a1,h,a2,s,m)

VP(a1=the-book,h=put,a2=on-the-desk,s=NIL,m=NIL)

f(a1,h,a2,s,m)
=(BA,a1,0,h,0,a2,0,s,0,m,0)

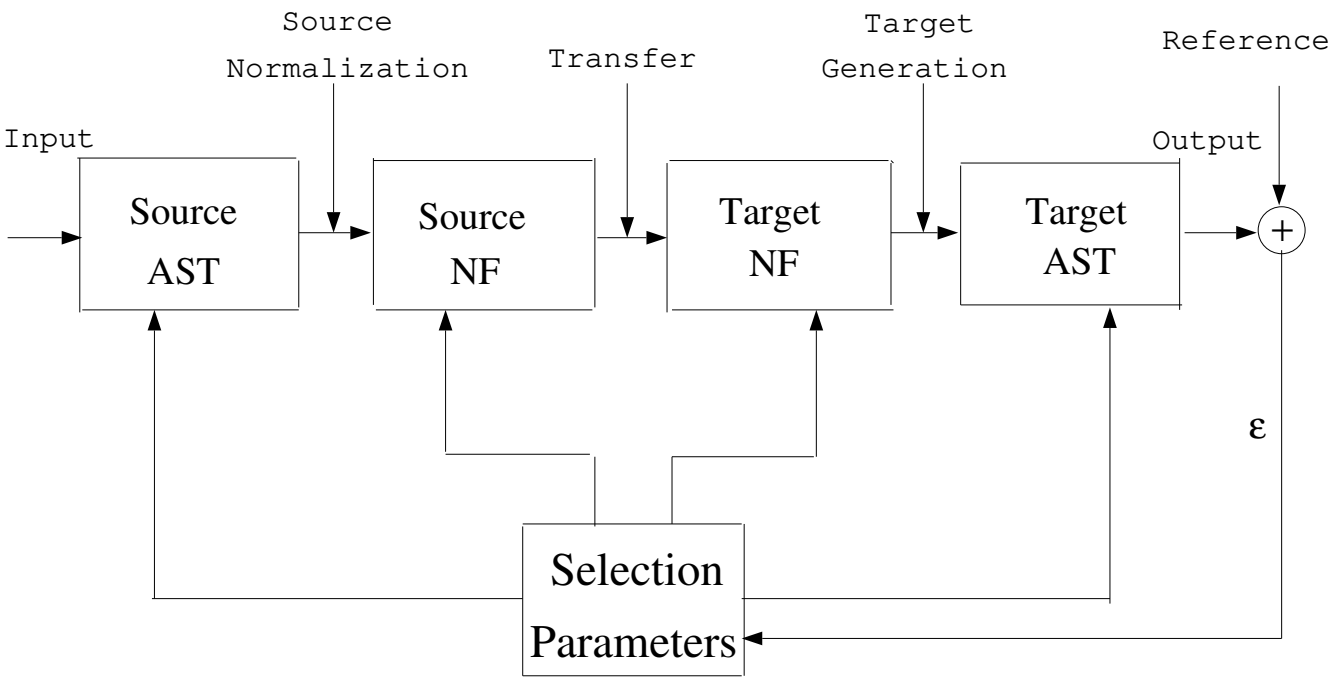
VP(BA,a1=BOOK,h=PUT,a2=ON-DESK)

Ambiguity Resolution and Lexicon Selection in a Feedback System



Parameter tuning from feedback

Bi-lingual Transfer Model



AST : Annotated Syntax Tree

NF : Normal Form

An adaptive learning conceptual model for the transfer and generation phases

New Trends in Language Processing

- Focus: theoretical formalisms toward real text processing and practical applications
- Domain: general interesting toward domain adaptation and specific sublanguage
- Analysis: syntactic toward semantic
- Processing Units: full parse toward partial parse
- Robustness: strictly well-formed formalisms toward ill-formedness, error recovery and robust parsing
- Knowledge Acquisition: human constructed rule/knowledge bases toward corpus-based statistics-oriented parameter training

☐ "Good Language Model + Good Learning Mechanism + Large Corpus" are the key factors to a successful large-scale language processing system

Conclusion

- Human is competent in establishing high level linguistic models, but awkward in dealing with a large amount of fine-grained knowledge consistently and cost-effectively.
- Machines are good at processing massive data.
- Statistics theories provide well established techniques for systematically processing corpus.
- Hence, the cooperative approach is recommended:
  - Setup probabilistic language models by human.
  - Estimate the parameters by machines.
- Combining the deduction capability of linguists and induction capability of the machine is probably the best way to make large scale MT systems possible.
- Corpus-Based Statistics-Oriented approaches will play an important role in knowledge acquisition for large MT systems.