Speech Recognition and Machine Translation: A Comparative Overview

Xiaodong He

Natural Language Processing group,
Microsoft Research

Redmond, WA, USA

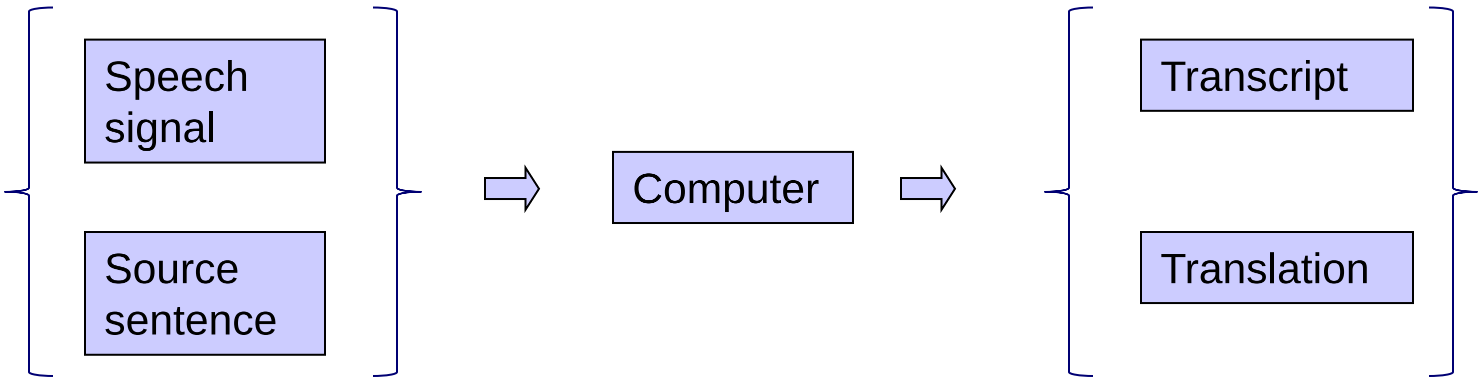
BIRS Multimedia, Mathematics and Machine Learning workshop II

Outline

- Introduction of ASR and SMT
- HMM: ASR vs. MT
- System Combination
- Summary

ASR & MT: Sequential PR Problems

Sequential Pattern Recognition:



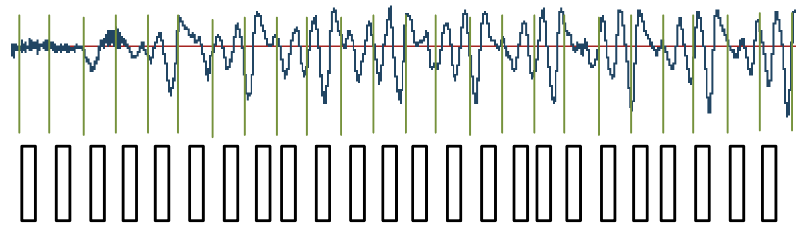
Input signal:
a sequence of
input samples

Output result:
a sequence of
output symbols

Simple Illustration of ASR and SMT

ASR

Speech signal



Feature seq.
(*via feature extraction*)

phonemes: *h* *au* *ar* *ju*

compose phones to word

Transcript: *how* *are* *you*

SMT

Source lang. sentence: 你 过得 怎样 ?

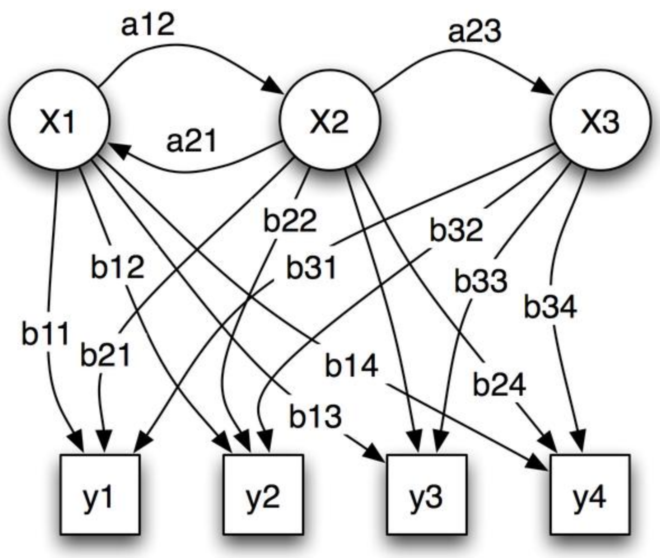
Lexical translation

Target lang. words: you are how ?

Word reordering

Translation: how are you ?

HMM for Sequential PR Problem



States of HMM Λ

Observation sample seq. Y

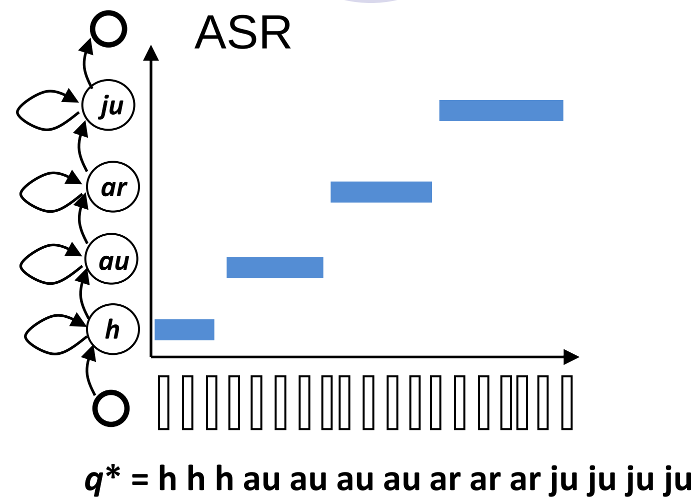
(From Wikipedia.org)

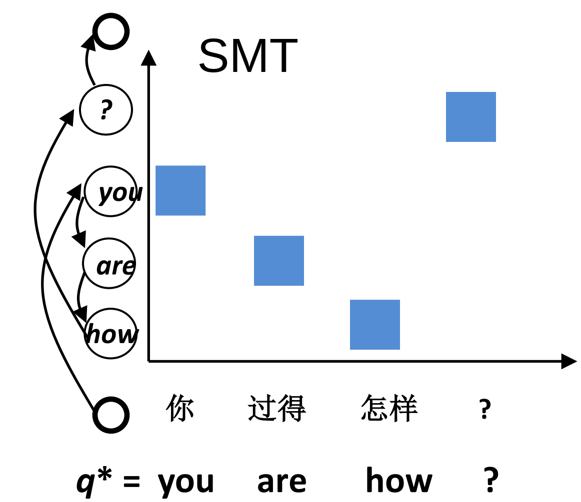
$$P(Y, q \mid \Lambda) = \Pi_t \{a_{qt-1,qt}\, b_{qt}(y_t)\}$$

- Training Problem: $\Lambda^* = \text{argmax}_{\Lambda}\{P(Y \mid \Lambda)\}$ [EM]
- Evaluation Problem: $P(Y \mid \Lambda) = \Sigma_q \{P(Y, q \mid \Lambda)\}$ [Forward/Backward]
- Decoding Problem: $q^* = \text{argmax}_q\{P(Y, q \mid \Lambda)\}$ [Viterbi]

HMM for ASR and MT: Alignment

- Align the input sample seq. to the reference symbol seq.
- HMM is used. each symbol in the reference is treated as a HMM state.
- ASR vs. MT:
  - *ASR*: Input speech samples and HMM states are in *monotonic* order.
  - *SMT*: Input source words and HMM states are in *non-monotonic* order.
- Viterbi decoding works for both ASR and MT (in polynomial time).



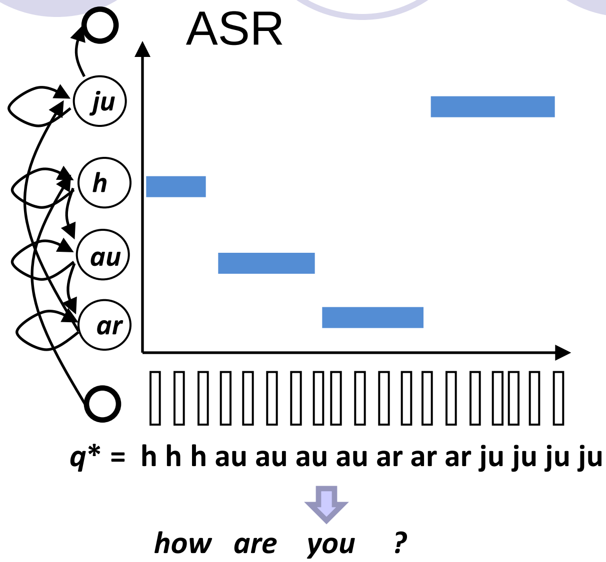


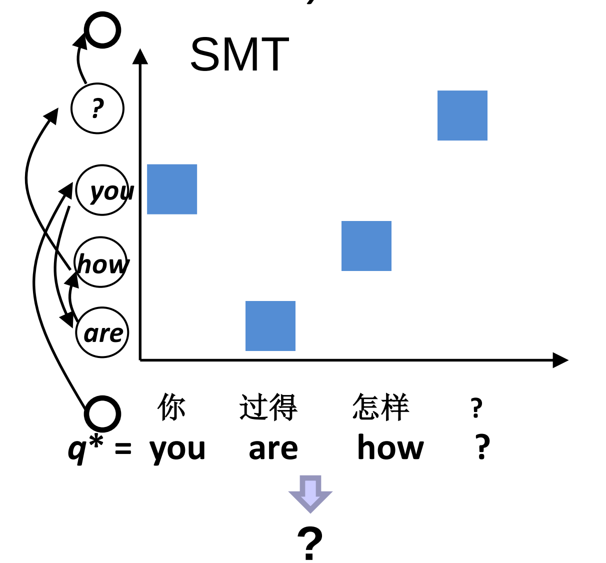
HMM for ASR and MT: Decoding

- Search for the optimal output symbol sequence given the input.
- HMM is used. Each symbol in the vocabulary is treated as a HMM state.
- ASR vs. MT:
  - *ASR*: Input speech and HMM states are *non-monotonic* (since need to explore all possible phone seq). But input is still monotonic to output.

    Viterbi works. (but harder)
  - *SMT*: The order of the output words can not be determined even if we find the best state sequence.

    Viterbi doesn't work.



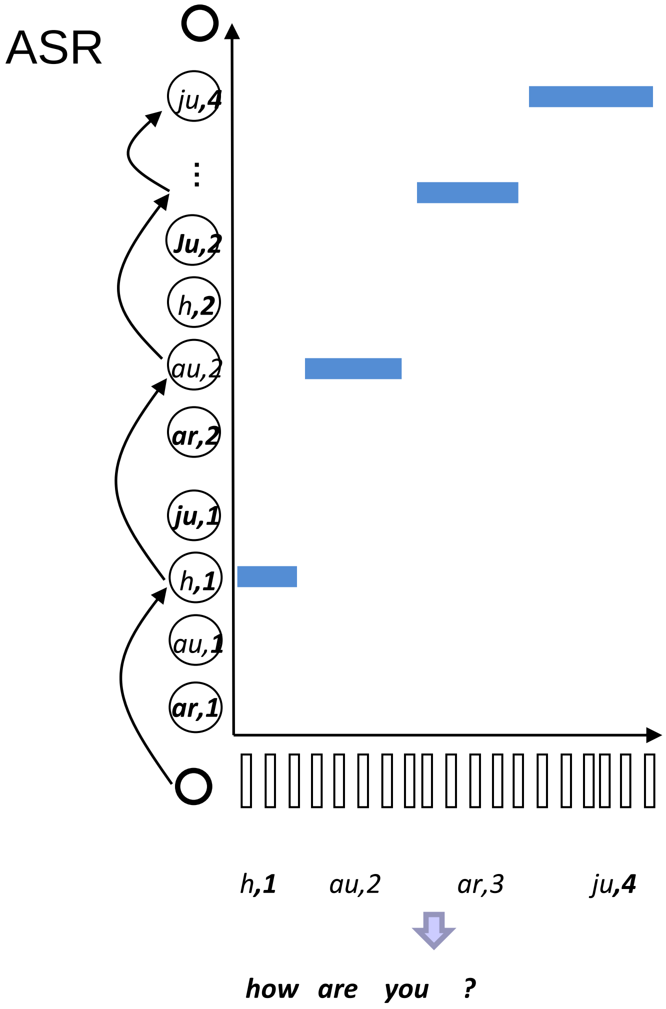


Extended HMM State

- To make the comparison clearer, we extend the previous HMM. I.e., each state is not only word/phone dependent, but also position dependent.
  - i.e., each state is a <*phone, pos*> or <*word, pos*> pair for ASR and MT, respectively.
    - *pos* is the position of the phone/word in the output phone/word sequence
  - Then, the state sequence determines both the output phones/words and their ordering.

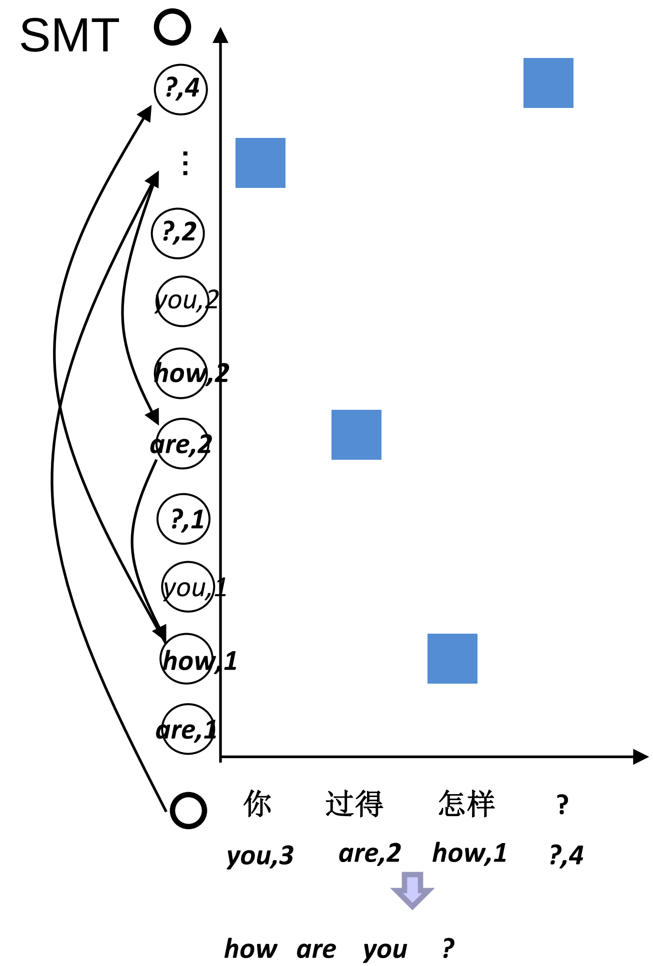
ASR after State Extension

- After state extension, decoding of ASR becomes *monotonic* .
- *Position constraint*: each position should be taken by one and only one phone.
  - This is out of the capability of a general HMM (bc. short memory).
  - *But* we can design the topology of the HMM such that
    - backward jump is not allowed
    - position skipping is not allowed
- Viterbi still works.
  - Given this topology, any valid state sequence meets the position constraint.



MT after State Extension

- After state extension, decoding of MT is *non-monotonic* .
- Note, now both the output words and their order can be determined if we can find the optimal state sequence.
- But not easy: *Position constraint*.
  - Unfortunately, no workaround as the ASR case.
    Viterbi doesn't work.
- The decoding problem is NP-complete since it needs to remember the past state history. (Traveling Sales Man problem.)



Highlights

- Word ordering is a major challenge distinguishing MT from ASR.
  - For training, since both input and output are known, don't need to "decide" the order of the output.
    - So HMM/Viterbi work for both ASR and MT
    - Still, MT is harder due to non-monotonic order
  - For decoding, HMM/Viterbi doesn't work for MT due to the non-monotonic-order problem.
    - It is more clear if we cast both ASR and MT into HMM with state extension:
      - MT decoding is a NP problem
      - ASR, instead, can survive after applying some tricks

System Combination for ASR

- ROVER (Fiscus, 97)
  - Recognizer Output Voting Error Reduction
  - Other works (Byrne et al.)
  - 10% to 20% error rate reduction.
- *Averaging* gives a result better than the best.

N-best from ASR systems
E_1: how you
E_2: how and you
E_3: who are you
E_4: how are oil

Combination

how	ε	you
how	and	you
who	are	you
how	are	oil
$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$

Theory Behind: MBR

- Given the observation F and a hypothesis E', Bayes-risk of classifying F to E'
  - $R(E') = \sum_{E \in \mathbf{E}_e} P(E \mid F) L(E', E)$
- MBR classification

$$E^* = \arg\min_{E' \in \mathbf{E}_h} \sum_{E \in \mathbf{E}_e} P(E \mid F) L(E', E)$$

  - $P(E \mid F)$: posterior probability
  - $L(E', E)$: loss function, application specific
  - $\mathbf{E}_h$: hypothesis space, for selecting classification candidate
  - $\mathbf{E}_e$: evidence space, for computing Bayes-risk

Segmental - MBR

- The global risk can be decomposed

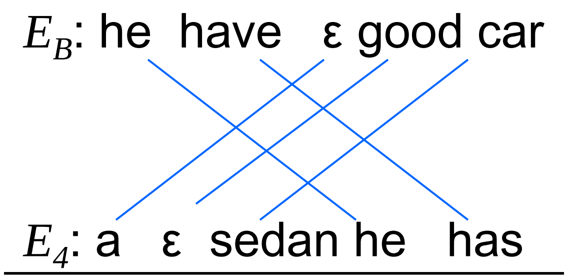
$$
\begin{aligned}
R(E') &= \sum_{E \in \mathbf{E}_e} P(E \mid F) L(E', E) \\
&= \sum_{E \in \mathbf{E}_e} P(E \mid F) \sum_{l=1}^{L} L(e'_l, e_l) \\
&= \sum_{l=1}^{L} \underbrace{\sum_{e_l \in \mathbf{e}_l} L(e'_l, e_l) \underbrace{\sum_{\substack{E : E \in \mathbf{E}_e \\ \& e_l \in E}} P(E \mid F)}_{\text{local posterior: } P(e_l|F)}}_{\text{local risk: } R(e'_l)}
\end{aligned}
$$

Minimizing global risk can be done by minimizing local risks

System Combination for SMT

N-best from MT systems
E_1: he have good car
E_2: he has nice sedan
E_3: it a nice car
E_4: a sedan he has

1) Hypothesis alignment



- Similar to ROVER of ASR.
- But alignment is challenging
  - Non-monotonic word ordering
  - Synonyms / Semantic similarity measurement

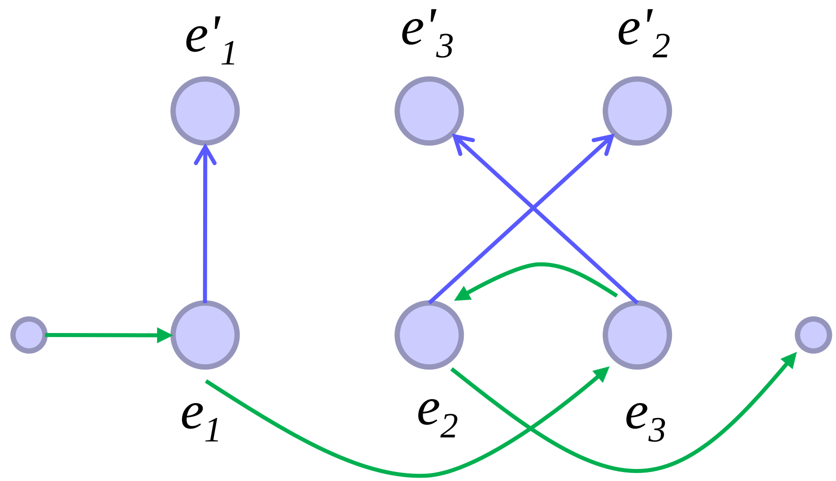
2) Confusion network

he	have	ε	good	car
he	has	ε	nice	sedan
it	ε	a	nice	car
he	has	a	ε	sedan
$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$

- Previous works: Matusov et al, Sim et al, Rosti et al., He et al.

HMM based Hypothesis Alignment

E_B :	e_1	e_2	e_3
E_{hyp}:	e'_1	e'_3	e'_2

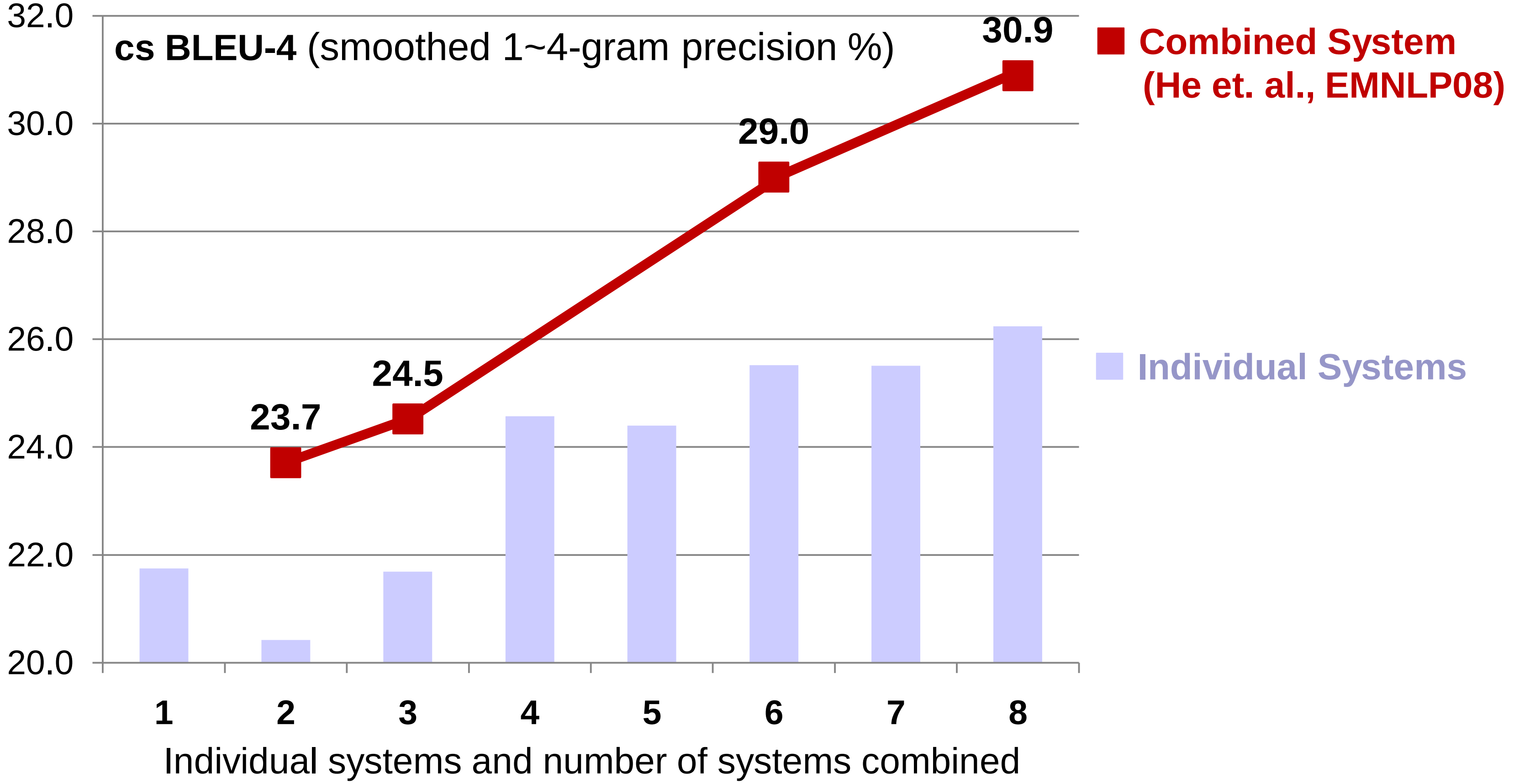


e_1	e_2	e_3
e'_1	e'_2	e'_3

- HMM is built on the backbone side
- HMM aligns the hypothesis to the backbone
- After alignment, a CN is built

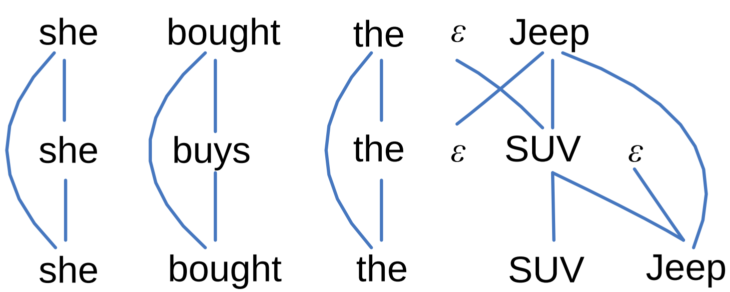
Results on 2008 NIST Open MT Eval

- The MSR-NRC-SRI entry for Chinese-to-English



Problems of ROVER

- Alignment, word ordering and lexical choice are decided independently.
- Lots of heuristics and local decisions



MT system hypotheses w/ pair-wise alignments.

she	bought	the	Jeep	ε
she	buys	the	SUV	ε
she	bought	the	SUV	Jeep

Conventional Confusion Network

Beyond ROVER: Direct Decoding

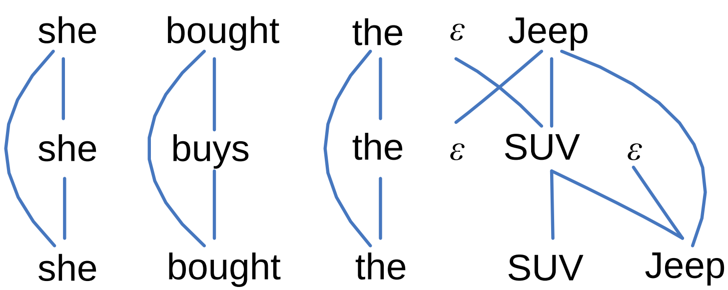
- A joint optimization framework via a max entropy model:

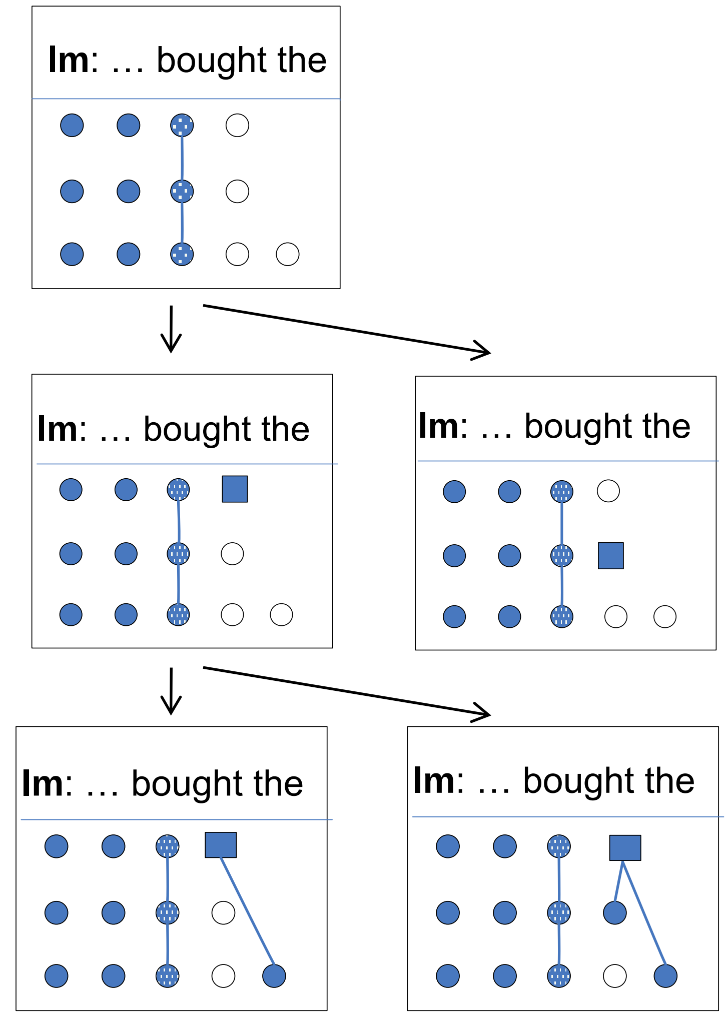
$$w^* = \operatorname*{argmax}_{w \in \boldsymbol{W}, O \in \boldsymbol{O}, C \in \boldsymbol{C}} exp\left\{\sum_{i=1}^{F} \alpha_i \cdot f_i(w, O, C, \boldsymbol{H})\right\}$$

- Features
  - Word posterior, bi-gram posterior, order distortion to input hyp, alignment score, word count, LM, alignment entropy
- Search Space
  - A product of the alignment, ordering, and lexical selection spaces.
- Decoding Algorithm
  - Beam search

(He and Toutanova, EMNLP09)

Decoding Algorithm





- A finite state machine
- Each state records:
  - Decoding cost, back-trace history, output words
- State expansion
- Beam pruning

Experimental Results

- Database: 2008 NIST MT Open Eval Chinese-to-English track
- Single systems: the top five C2E entries of NIST MT08
- Training and testing data: divide the data into dev set and test set.
- Evaluation metric: ci BLEU

System ID	dev	test
System A	32.88	31.81
System B	32.82	32.03
System C	32.16	31.87
System D	31.40	31.32
System E	27.44	27.67
IHMM baseline	36.91	35.85
Incremental HMM	37.32	36.38
Direct Decoding	37.94	37.20

Summary

- Both ASR and MT are sequential pattern recognition problem.
- Techniques in ASR and MT can be cross-fertilized.
- However, the difference between ASR and MT raises special challenges (or opportunities)
  - Word ordering
  - Semantic features
  - Context dependency

Thank you!

Two online machine translation services:

Microsoft MT
http://www.microsofttranslator.com/

Google MT
http://translate.google.com/