UNIVERSITY OF CAMBRIDGE

Neural Grammatical Error Correction with Finite State Transducers

Felix Stahlberg, Christopher Bryant, and Bill Byrne

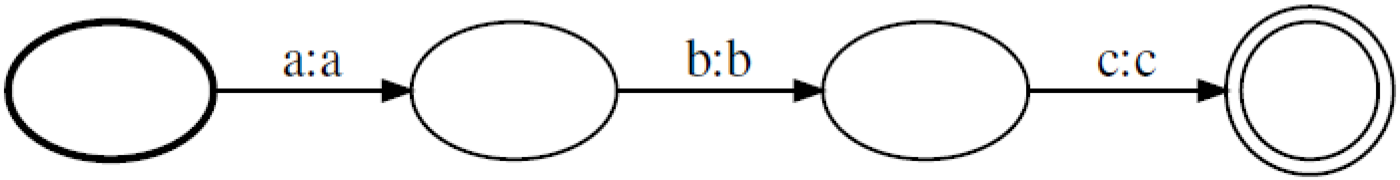
Department of Engineering

Informal introduction to finite state transducers

- FSTs are graph structures with start state and final state
- Arcs are annotated with:
  - An input symbol
  - An output symbol
  - A weight
- The FST transduces an input string s_1 to an output string s_2 iff. there is a path from the start to the final state with:
  - s_1 is the concatenation of all input symbols
  - s_2 is the concatenation of all output symbols
  - The cost of this mapping is the (minimal) sum of arc weights

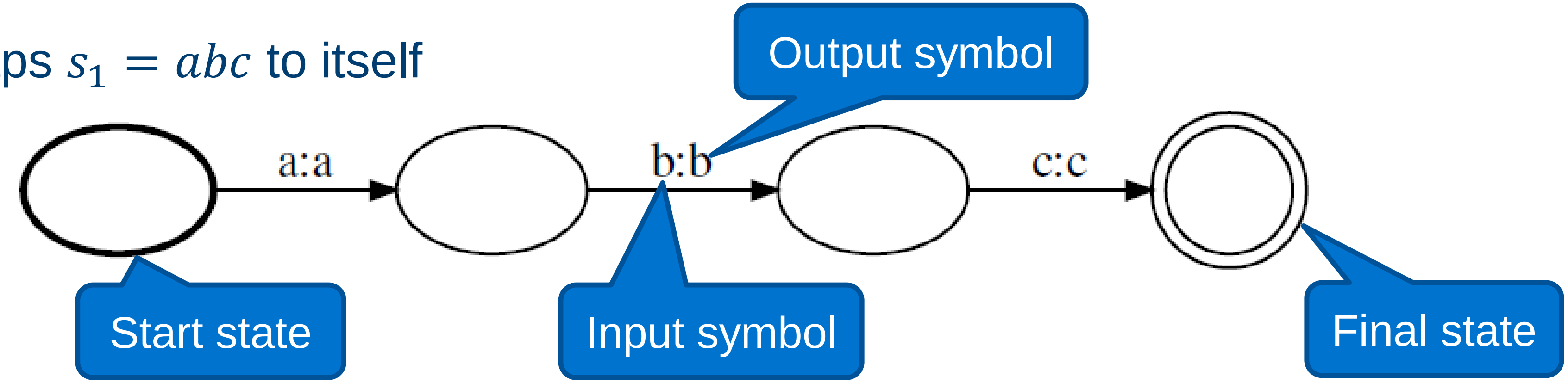
Example FSTs

- Maps $s_1 = abc$ to itself



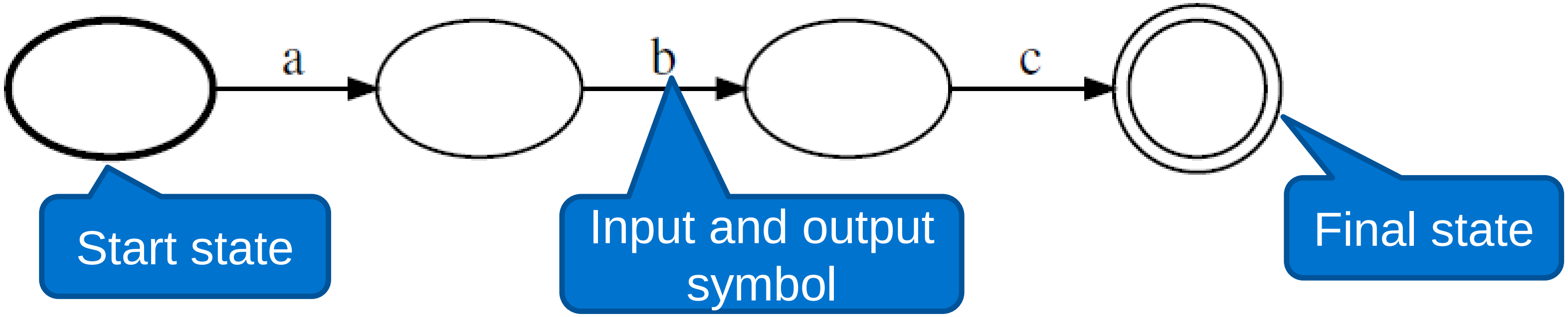
Example FSTs

- Maps $s_1 = abc$ to itself



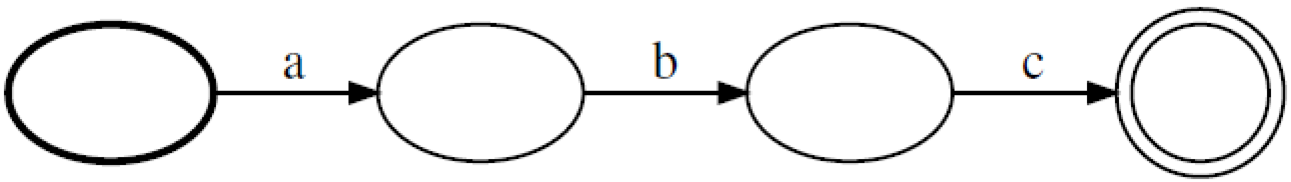
Example FSTs

- Maps $s_1 = abc$ to itself

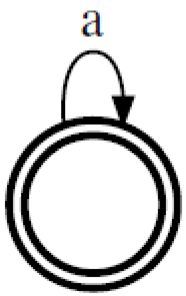


Example FSTs

- Maps $s_1 = abc$ to itself



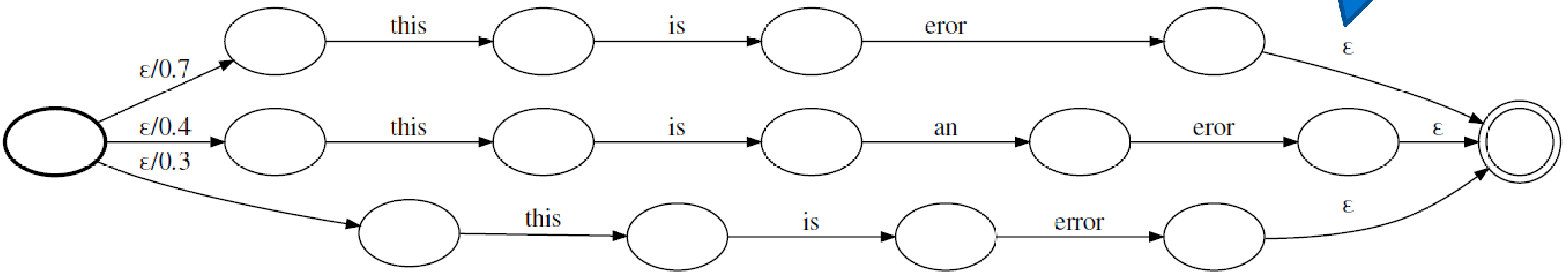
- Maps any string consisting only of a symbols to itself



Example FSTs

ϵ: empty input/output symbol

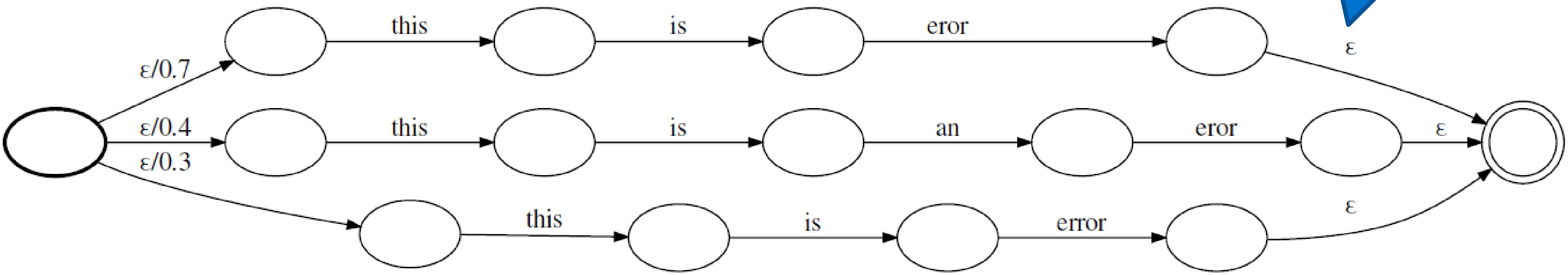
- Represents an n-best list



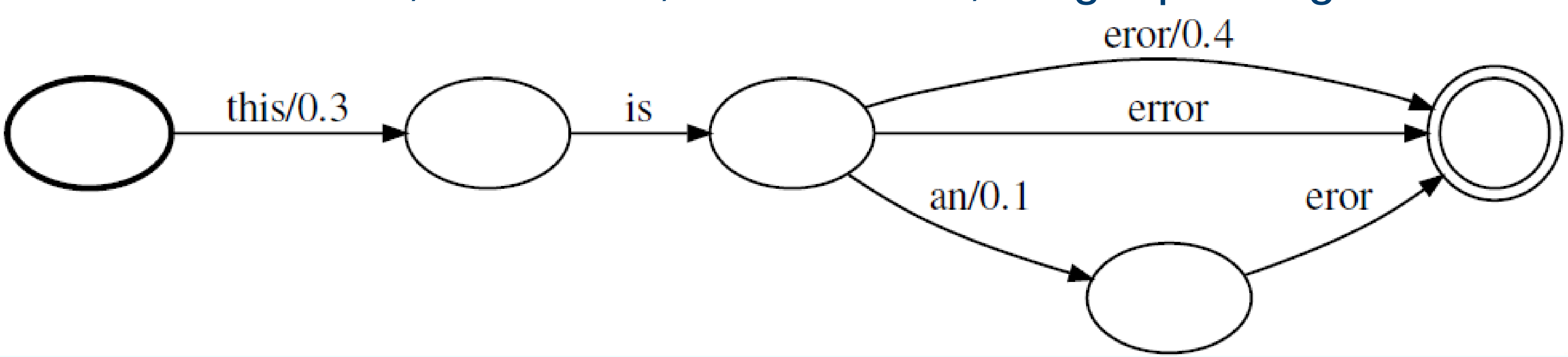
Example FSTs

ϵ: empty input/output symbol

- Represents an n-best list



- After determinization, ϵ-removal, minimization, weight pushing

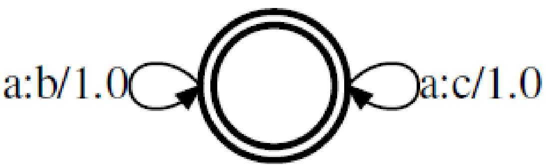


FST composition

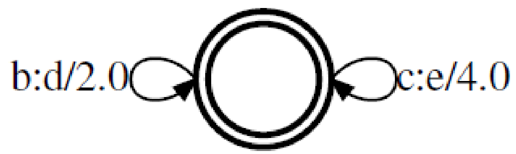
- Composition: Combines two FSTs T_1 and T_2 to a new FST $T_1 \circ T_2$
- If T_1 maps s_1 to s_2 and T_2 maps s_2 to s_3, then $T_1 \circ T_2$ maps s_1 to s_3.
- The cost is the (minimum) sum of the path costs in T_1 and T_2.

FST composition examples

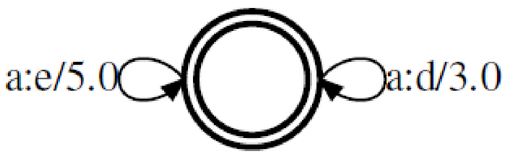
- Composition and weights



T_1



T_2

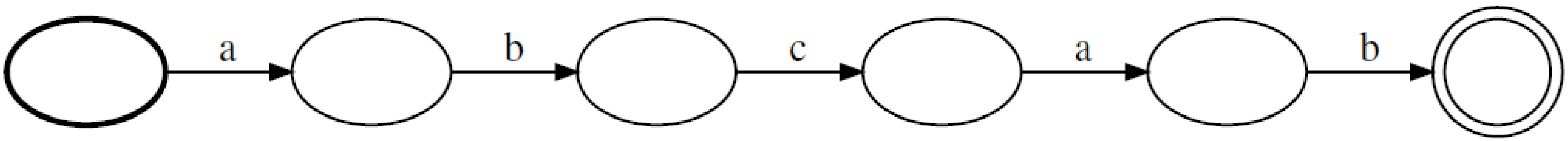


$T_1 \circ T_2$

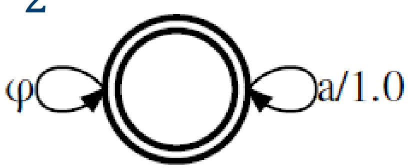
FST composition examples

- Counting transducers

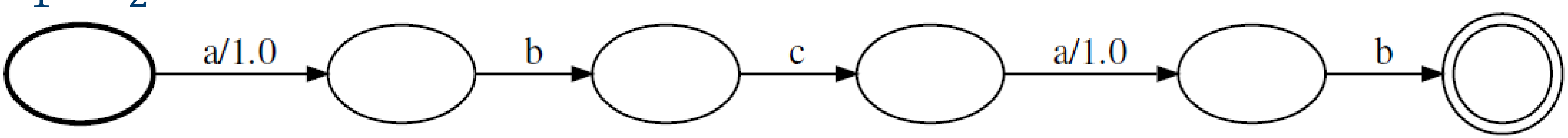
T_1



T_2



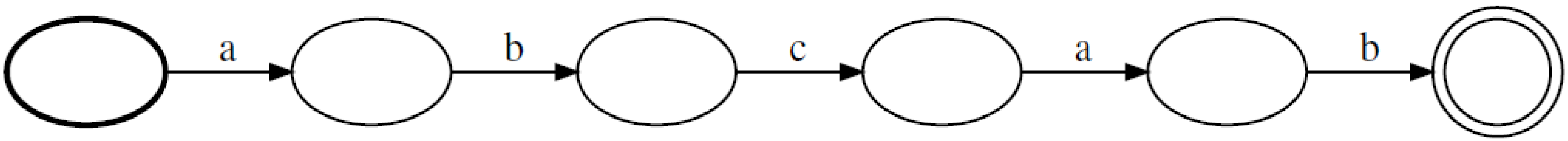
$T_1 \circ T_2$



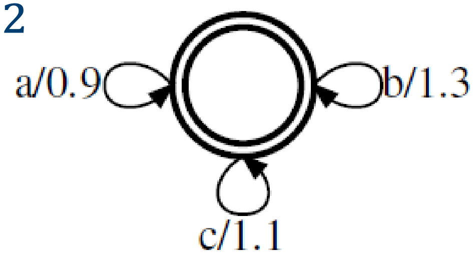
FST composition examples

- Language models

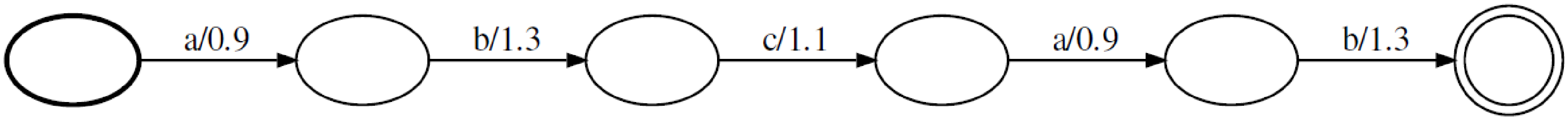
T_1



T_2



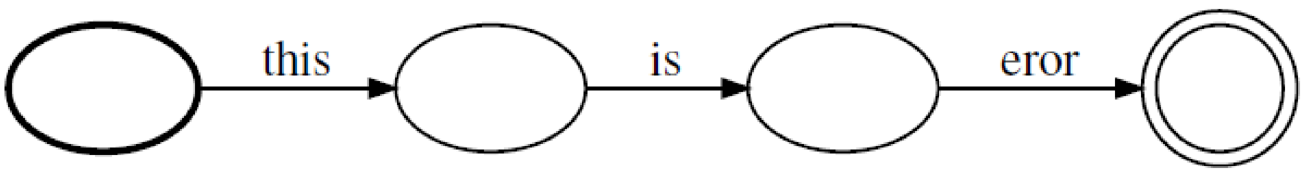
$T_1 \circ T_2$



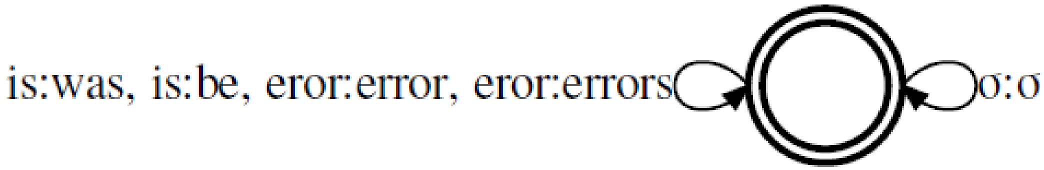
FST composition examples

- 1:1 corrections

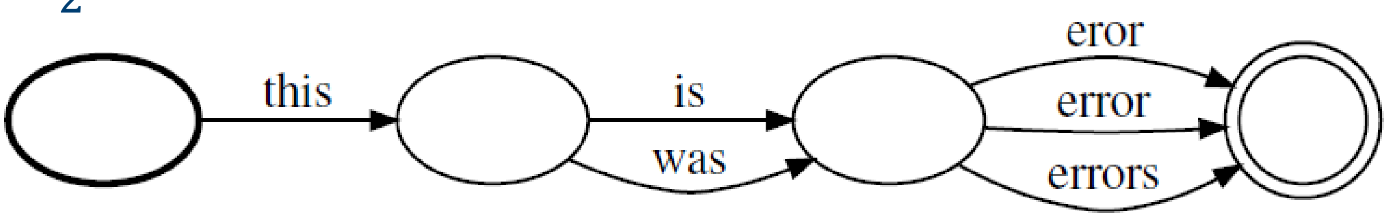
T_1



T_2

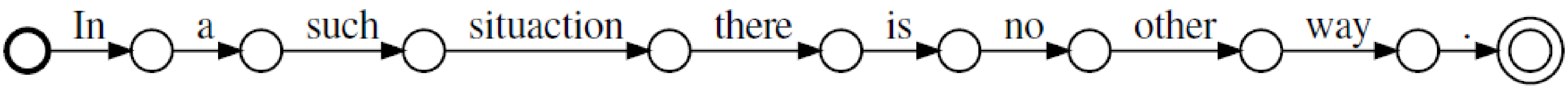


$T_1 \circ T_2$

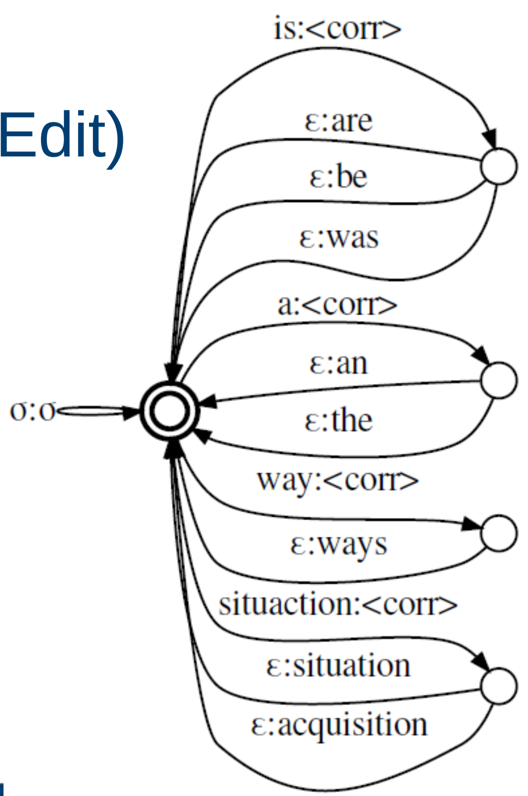


FST-based unsupervised grammatical error correction

I (Input)



E (Edit)



P (Penalization)

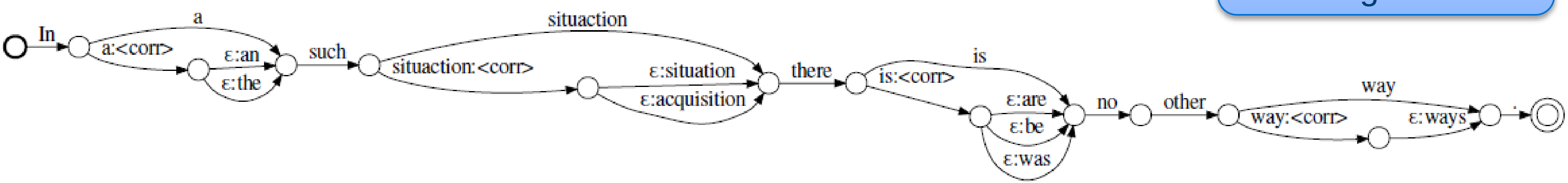
L (5-gram LM)

...

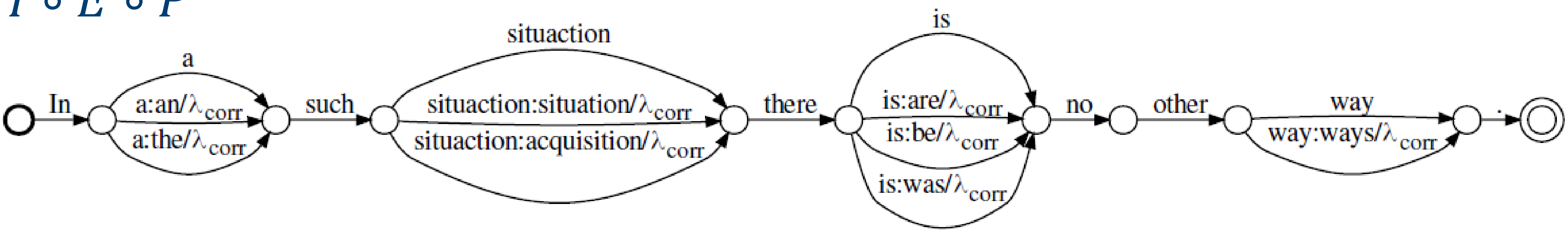
FST-based unsupervised grammatical error correction

- I: Input
- E: Edit
- P: Penalization
- L: 5-gram LM

$I \circ E$



$I \circ E \circ P$



$I \circ E \circ P \circ L$: Non-neural unsupervised GEC with 5-gram LM scores

FST-based neural unsupervised GEC

- I: Input
- E: Edit
- P: Penalization
- L: 5-gram LM
- T: Tokenization (word → BPE)

- Idea: Use the constructed FSTs to constrain the output of a neural LM
- Neural sequence models normally use subwords or characters rather than words.
- Build transducer T that maps full words to subwords (byte-pair encoding, BPE)
- Constrain neural LM with $I \circ E \circ P \circ L \circ T$
- For constrained neural decoding we use our SGNMT decoder http://ucam-smt.github.io/sgnmt/html/

Results (unsupervised)

	Uses E	5-gram FST-LM	NLM (BPE)	CoNLL-2014 P	CoNLL-2014 R	CoNLL-2014 M2	CoNLL-2014 GLEU	JFLEG Test P	JFLEG Test R	JFLEG Test M2	JFLEG Test GLEU
1	Best published (B&B, 2018)			40.56	20.81	34.09	59.35	76.23	28.48	57.08	48.75
2	✓	✓		40.62	20.72	34.08	64.03	81.08	28.69	59.38	48.95
3	✓	✓	✓	54.43	25.21	44.19	66.75	79.88	32.99	62.20	50.93
4	✓	✓	✓	53.64	26.34	44.43	66.89	70.24	38.94	60.51	52.61

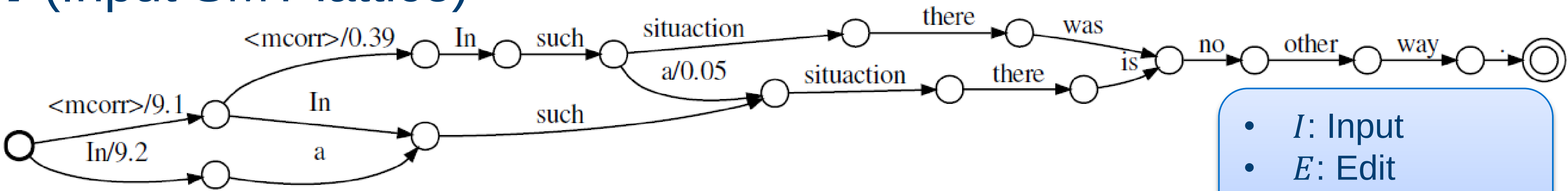
Systems are tuned with respect to metric highlighted in grey.

FST-based neural supervised GEC

- If annotated training data is available:
  - Input I is a (Moses) SMT lattice rather than a single sentence
  - In addition to the <corr> token, we use an <mcorr> token to count the edits by the SMT system.
  - We use an ensemble of a neural language model and a neural machine translation model.

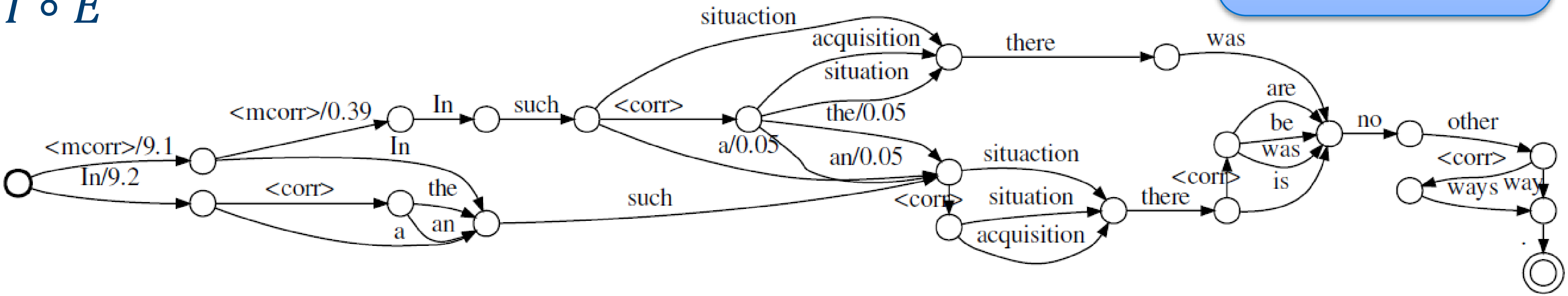
FST-based supervised grammatical error correction

I (Input SMT lattice)



- I: Input
- E: Edit
- P: Penalization
- L: 5-gram LM
- T: Tokenization (word → BPE)

$I \circ E$



$I \circ E \circ P \circ L \circ T$: Constraint for neural ensembles

Results (supervised)

	Uses E	5-gram FST-LM	NMT (BPE)	NLM (BPE)	CoNLL-2014 P	CoNLL-2014 R	CoNLL-2014 M2	CoNLL-2014 GLEU	JFLEG Test P	JFLEG Test R	JFLEG Test M2	JFLEG Test GLEU
1	Best published (G&J-D, 2018)				66.77	34.49	56.25	n/a	n/a	n/a	n/a	61.50
2					60.95	26.21	48.18	68.30	66.64	40.68	59.09	50.86
3	✓	✓			57.58	32.39	49.83	68.82	71.60	42.45	62.95	53.20
4			✓	✓	65.26	33.03	54.61	69.92	76.35	40.55	64.89	51.75
5	✓		✓	✓	64.55	37.33	56.33	70.30	78.85	47.72	69.75	55.39
6	✓		✓(4x)	✓	66.71	38.97	58.40	70.60	82.15	47.82	71.84	55.60
7	✓		✓(4x)	✓	66.96	38.62	58.39	70.60	74.19	56.41	69.79	58.63

Systems are tuned with respect to metric highlighted in grey.

Results (supervised)

	G&J-D (2018)		This work	
	CoNLL (M2)	JFLEG (GLEU)	CoNLL (M2)	JFLEG (GLEU)
SMT	50.27	55.79	48.18	50.86
Hybrid	56.25	61.50	58.40	58.63
Rel. gain	**11.90%**	**10.23%**	**21.21%**	**15.28%**

Thanks

BACKUP

$j = 1$

NMT posteriors:

y_1	$P(y_1)$
A	0.4
B	0.2
C	0.3
UNK	0.1
</s>	0.0

Combined:

y_1	$P(y_1)$
A	0.0
B	0.1
C	0.15
UNK	0.0
</s>	0.0

$j = 2$

NMT posteriors:

y_2	$P(y_2)$
A	0.05
B	0.2
C	0.1
UNK	0.6
</s>	0.05

Combined:

y_2	$P(y_2)$
A	0.0
B	0.0
C	0.06
UNK	0.24
</s>	0.0

$j = 3$

NMT posteriors:

y_3	$P(y_3)$
A	0.1
B	0.1
C	0.4
UNK	0.1
</s>	0.3

Combined:

y_3	$P(y_3)$
A	0.0
B	0.0
C	0.0
UNK	0.0
</s>	0.3