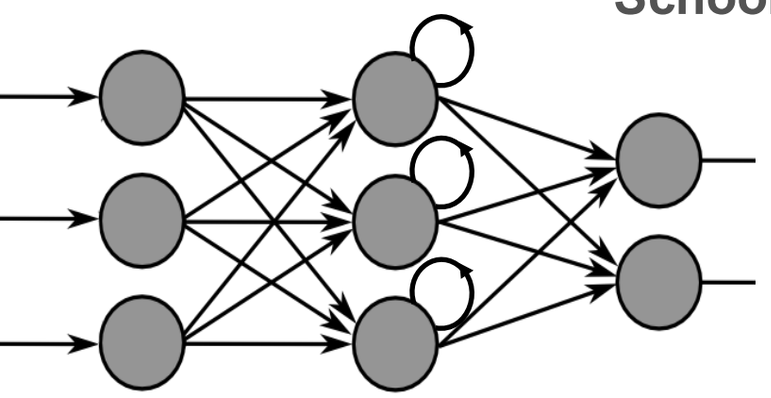
DEEP LEARNING DRIVEN INCREMENTAL DISFLUENCY DETECTION: METHODS AND APPLICATIONS

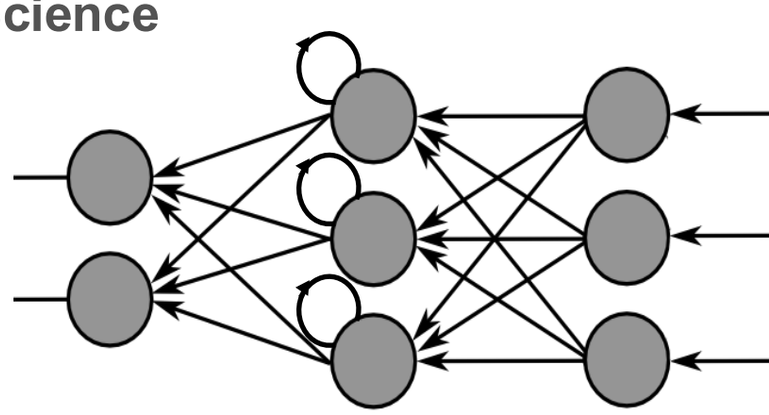
Julian Hough[1]

with Morteza Rohanian[1], Tom Gurion[1], Shamila Nasreen[1], Matthew Purver[1], Pat Healey[1] and David Schlangen[2]

1. **Cognitive Science Group**
School of Electronic Engineering and Computer Science
Queen Mary University of London

2. Computational Linguistics Lab
Potsdam University





CONTENTS

CONTENTS

THE NEED FOR LIVE DISFLUENCY DETECTION

- Say you wanted to call the White House from Europe on 001 202-456-1111

User: Zero Zero One
Sys: Zero Zero One?
User: Yes
User: Two Zero Two
Sys: Two Zero Two?
User: Yes
User: Four Five Six
Sys: …

- As dialogue system designers, we might presuppose the user says exactly what they intended and never changes their mind nor indicates hesitation…

THE NEED FOR LIVE DISFLUENCY DETECTION

- However, they may have ‘misdialled’ vocally or indicate uncertainty:

User: Zero One, uh, no, Zero Zero One
Sys: Zero Zero One?
User: Yes
User: Two Zero Three, uh, I mean, Two
Sys: Two Zero Two?
User: Um , yes
User: Four Five Six
Sys: …

- We need a way for the system to deal with user **disfluency and self-repair,** which happens in natural conversation every 25-30 words.
- A system needs to recover the **repaired meaning**.
- **Robots** need to recognize this too and react **quickly:** ‘Cut the red, uh no, blue wire’.

THE NEED FOR LIVE DISFLUENCY DETECTION

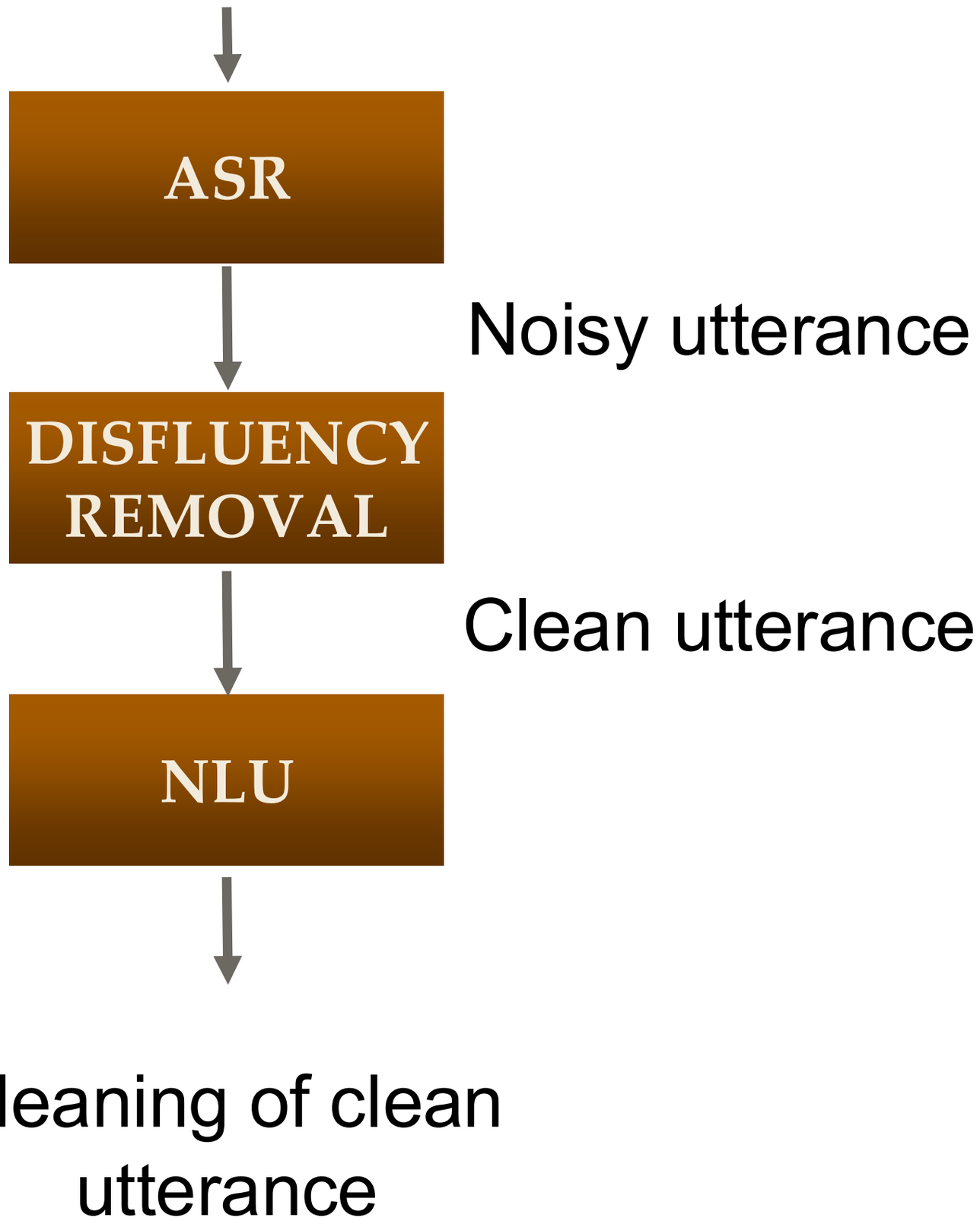
- Also, for clinical dialogue systems, there is a need to recognize disfluencies.

- In clinical situations **repair rates** (per utterance) can predict **adherence to treatment** for Schizophrenia patients (Howes et al., 2012).

- Disfluency presence can be useful in **depression** and **Alzheimer's Disease** detection. (Nasreen et al 2021).

DISFLUENCY IN DIALOGUE SYSTEMS

The standard picture:

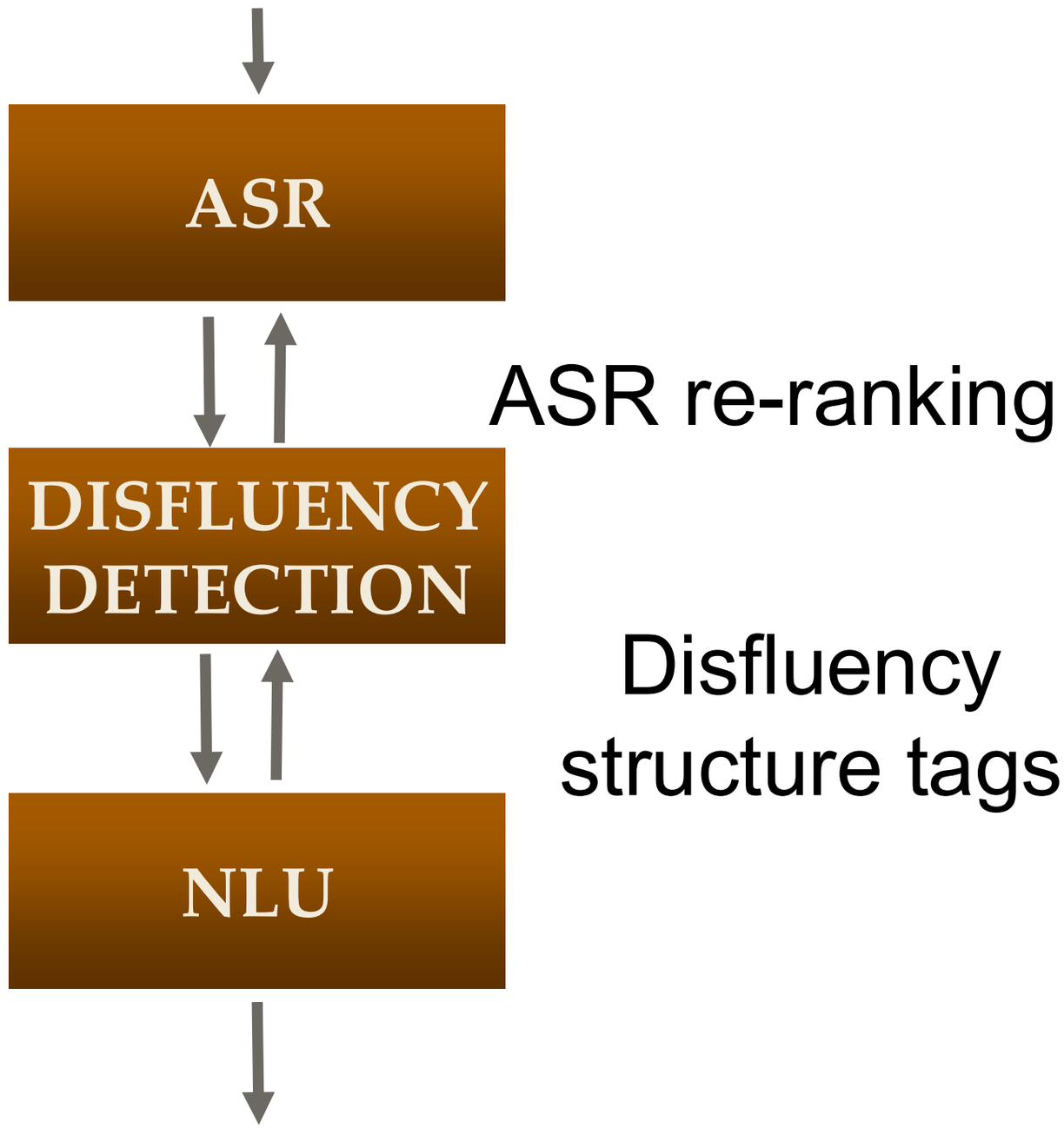


DISFLUENCY IN DIALOGUE SYSTEMS

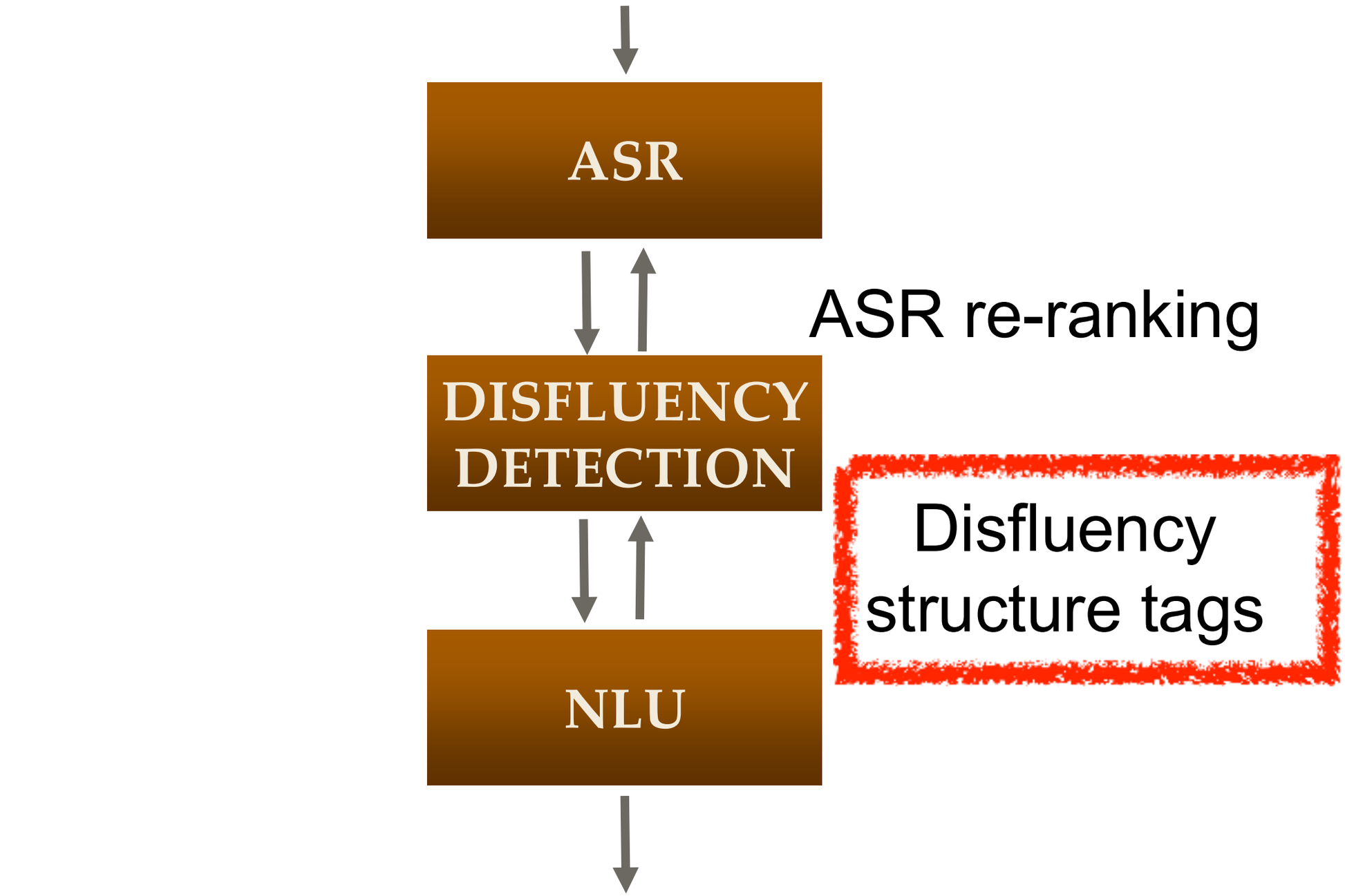
The standard picture:

FIGHT FOR DISFLUENCY RIGHTS!

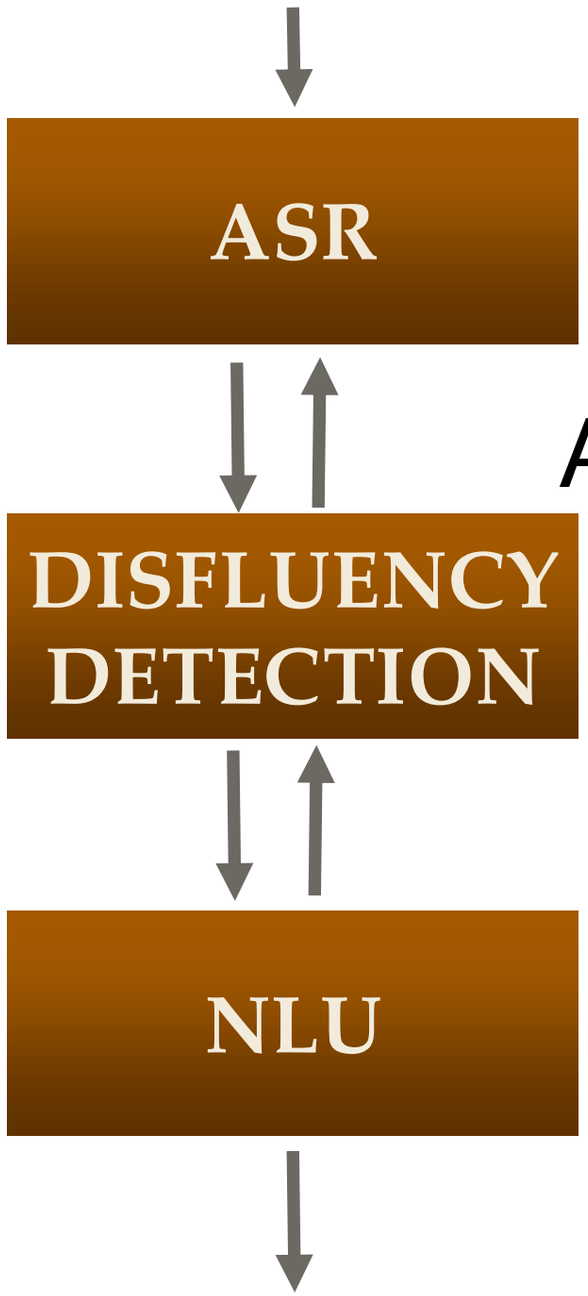
DISFLUENCY IN DIALOGUE SYSTEMS



DISFLUENCY IN DIALOGUE SYSTEMS



DISFLUENCY IN DIALOGUE SYSTEMS



ASR re-ranking

Disfluency structure tags

Incrementally:
Zwarts et al. (2010)
Hough and Purver (2014)
Heavily engineered LM features

Meaning of utterance with disfluency

A NAÏVE APPROACH

- A first stab at this might be to list all the **edit terms** in the language, such as ‘uh’, ‘um’ ‘no’, ‘I mean’ and treat those as repair indicators.
- However, **the data don’t lie**. From the most predictive up-coming repair signal **‘uh’,** there is a repair only **15.5%** of the time (in Switchboard dialogues)…

form	p(repair|form)	p(form|repair)
(fluent word)	0.039	0.842
“uh”	0.155	0.071
“you know”	0.100	0.037
“um”	0.061	0.011
“I mean”	0.074	0.005
“well”	0.080	0.005
“or”	0.017	0.003
“like”	0.014	0.003
“yeah”	0.038	0.002
“oh”	0.005	0.002
“actually”	0.025	0.001

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

"But one of **the, the** two things that I'm really. . ."

"And **they have a, they don't have any kind of** pension plan..."

"and you know it's like **you're, I mean,** employments are contractual by nature anyway"

[Switchboard examples]

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

Repairs:

John [likes + {uh} loves] Mary

reparandum interregnum repair

(Shriberg 1994) onwards

Edit terms:

‘uh’, ‘um’, ’I mean’, ‘like’, ‘you know’

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

"But one of [the, + the] two things that I'm really. . ."
[repeat, 51.3% of repairs]

"And [they have a + they don't have any kind of] pension plan..."
[substitution, 41.5% of repairs]

"and you know it's like [you're + {I mean}] employments are contractual by nature anyway"
[delete, 7.2% of repairs]

[Switchboard examples]

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

"{Um} [the interview was, + it was] alright"
[substitution with anaphora, <5%]

"Peter went [swimming with Susan + {or rather} surfing] "
[substitution with verb phrase ellipsis, <5%]

DISFLUENCY IN SPOKEN HUMAN DIALOGUE

- Listeners **use the reparandum** to compute meaning in substitutions. Psycholinguistic experiments suggest repairs ***aid*** comprehension (Brennan and Schober, 2001):

  "The yel-, purple square"

- **Repeats and isolated edit terms** mean something like a hesitation (**forward-looking disfluency**), while substitutions and deletes are revisions/**backwards-looking** (Ginzburg et al 2014, Hough 2015)

- Deletes more likely to have an interregnum than substitutions, which are both more likely to have one than repeats. **The more 'destructive' the repair, the more likely there's a signal of upcoming trouble.**

DISFLUENCY IN DIALOGUE SYSTEMS

RESEARCH GOAL:

Method for live **disfluency detection (not removal) on continuous speech**, which, at minimum:

- **Works live** on ASR results.
- Identifies the **different types** of disfluency**.**
- Exhibits **good, fast incremental** performance.
- Has good **disfluency rate correlation** to human-annotated data.
- Using **deep learning** methods**.**

CONTENTS

LEFT-RIGHT DISFLUENCY TAGGING

Joint work with **David Schlangen**
at Bielefeld on the DUEL project 2014-17

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { uh, I mean } to Denver]

```
f   e   f       f   f       e   e   e   rpS-5   rpESub
```

Joint work with **David Schlangen**
at Bielefeld on the DUEL project 2014-17

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { uh, I mean } to Denver]

`f e f f f e e e rpS-5 rpESub`

Tag	Meaning
`f`	Fluent word
`e`	Edit term
`rpS-[1-8]`	Repair Start with Reparandum start 1-8 words back
`rpMid`	Mid repair word
`rpESub`	End of repair (substitution or repetition)
`rpEDel`	End of repair (delete)

- 27 possible tags, some very sparse in training data

LEFT-RIGHT DISFLUENCY TAGGING

SPARSE!

	% of words
f	88.35
e	7.10
rpS-1_rpESub	1.64
rpESub	1.08
rpS-2_rpMid	0.46
rpS-2_rpESub	0.34
rpS-3_rpMid	0.21
rpS-1_rpMid	0.19
rpS-1_rpEDel	0.12
rpS-3_rpESub	0.11
rpS-4_rpMid	0.11
rpS-5_rpMid	0.05

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [t(

f e f

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { ul

f e f f f

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { uh, I mean } to I

f e f f f e e e

LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { uh, I mean } to Denver]

```
f  e  f     f  f        e  e  e    rpS-5
```

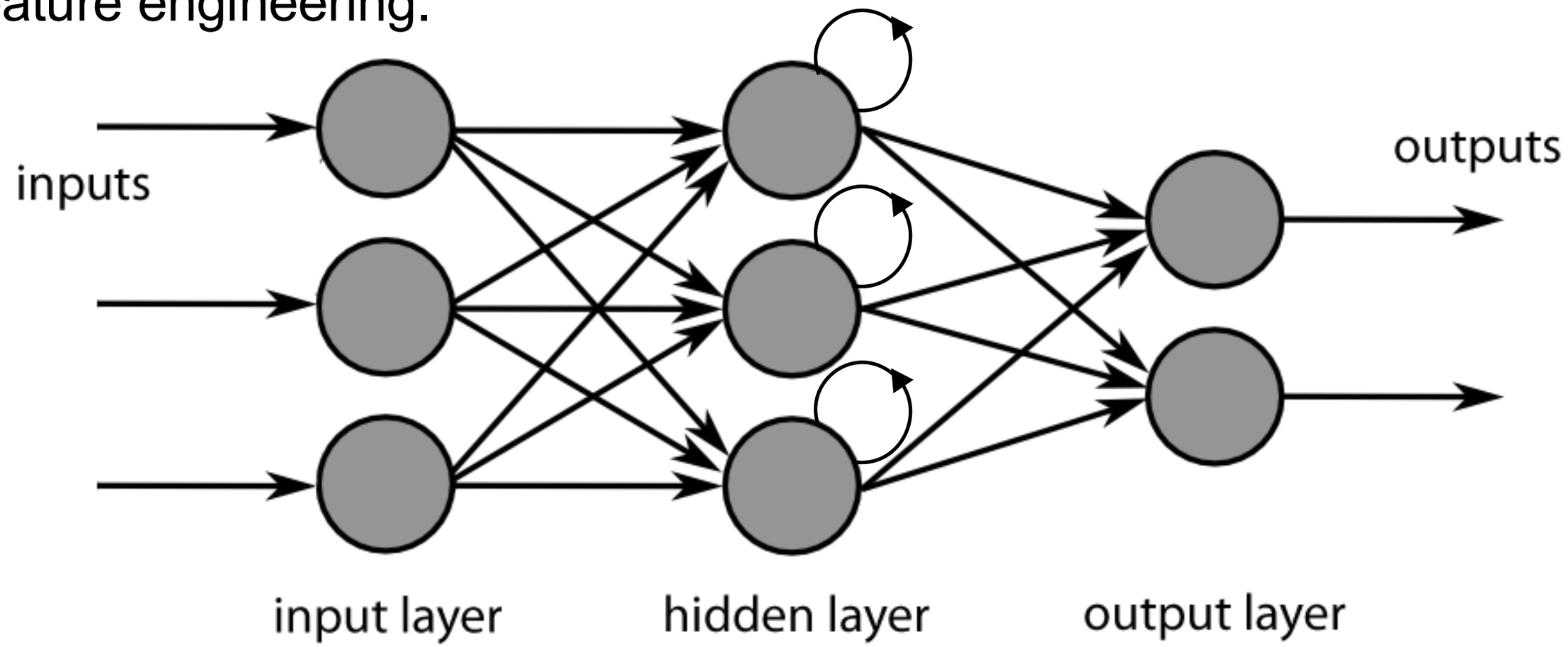
LEFT-RIGHT DISFLUENCY TAGGING

A { uh } flight [to Boston, + { uh, I mean } to Denver]

```
f  e  f     f  f     e  e  e   rpS-5  rpESub
```

RNNS FOR DISFLUENCY TAGGING

- Recurrent Neural Networks (RNNs) successful in other NLP tasks e.g. SLM (Mikolov et al. 2011) and NLU (Mensil et al. 2013)- reduces feature engineering.



- Use ‘working memory’ through using copy of hidden layer from previous time step- memory required for detecting repairs
- For left-right sequence tagging one of the best classifiers (in 2015!)…

RNNS FOR DISFLUENCY TAGGING

RESEARCH QUESTIONS:

Q1 How well can a vanilla RNN perform on standard disfluency detection on pre-segmented data?

Q2 How suitable is it for *incremental left-right* processing in this task?

Q3 How do different input types affect performance?

(Hough and Schlangen, 2015 InterSpeech)

RNNS FOR DISFLUENCY TAGGING

RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

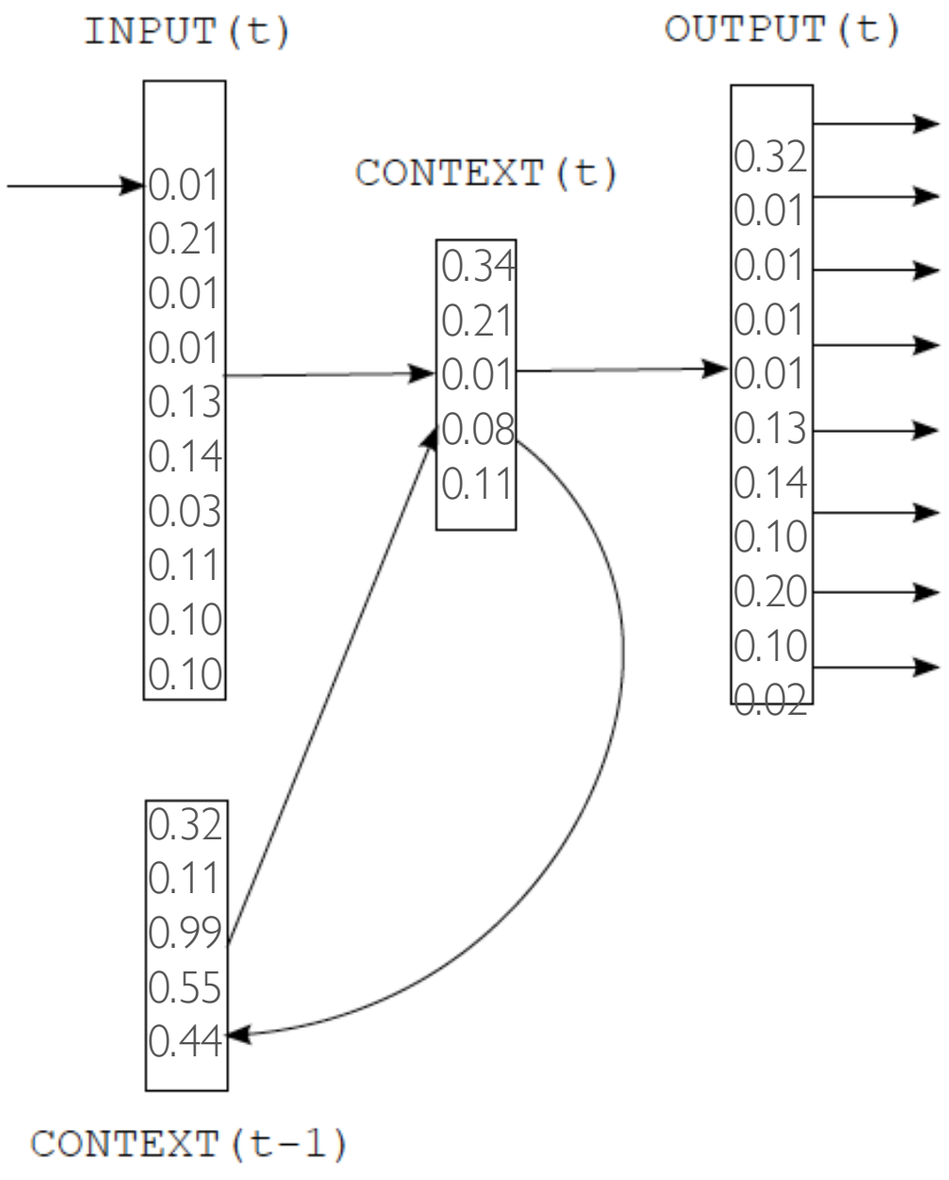
f e rpS-2 f f

rpESub

RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

f e rpS-2 f f
rpESub

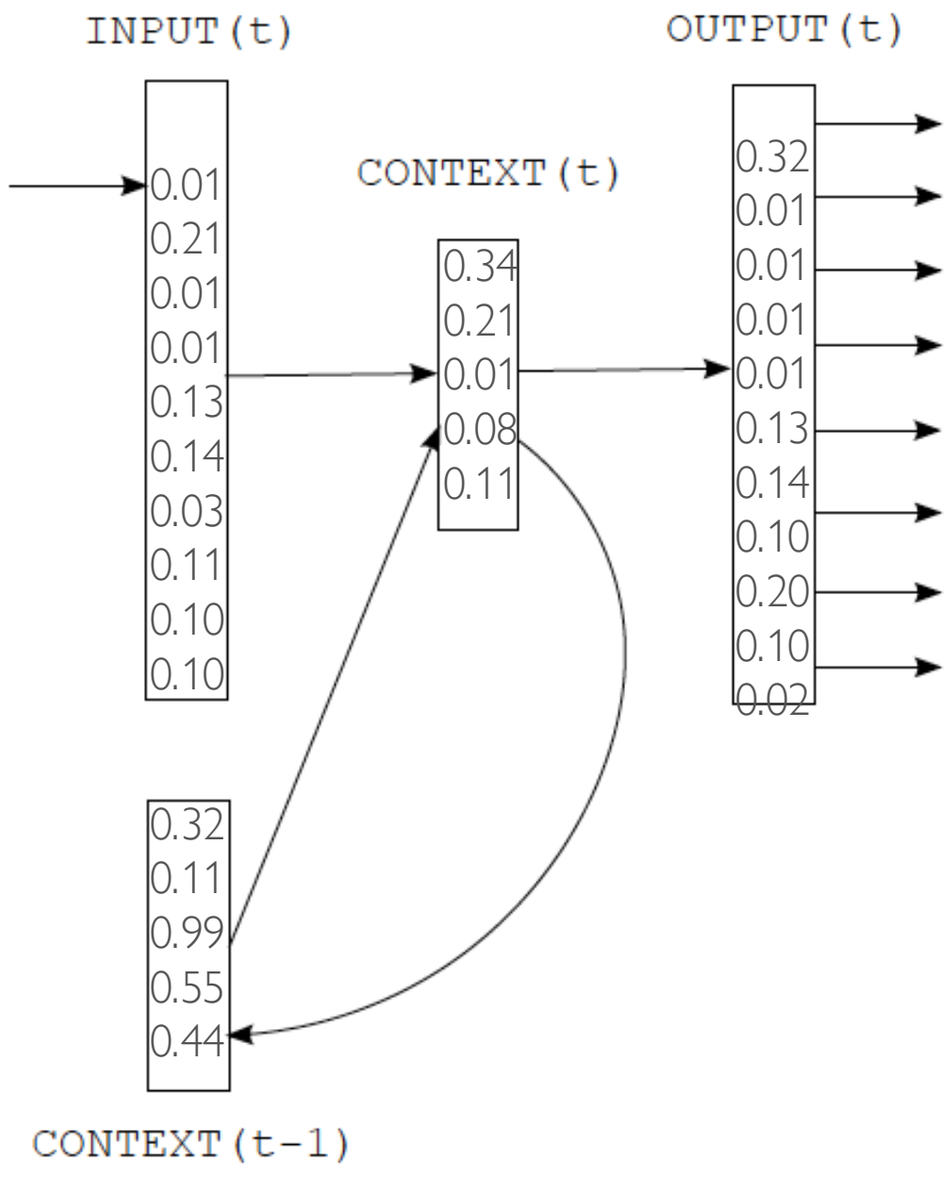


RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

f e rpS-2 f f

rpESub



RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

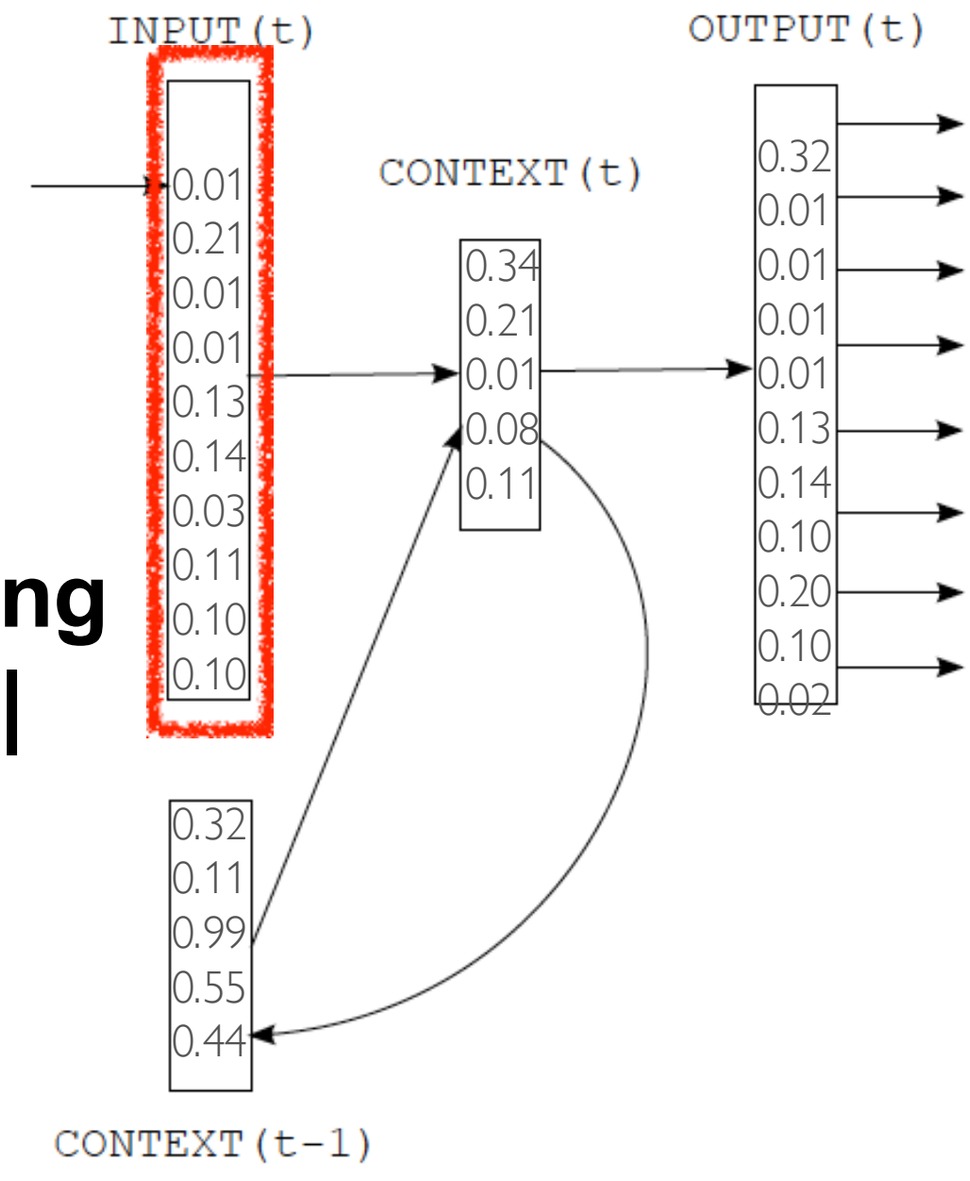
f e rpS-2 f f

rpESub

john

Wt

Indices map to **word embedding** of dimension **|e|**



RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

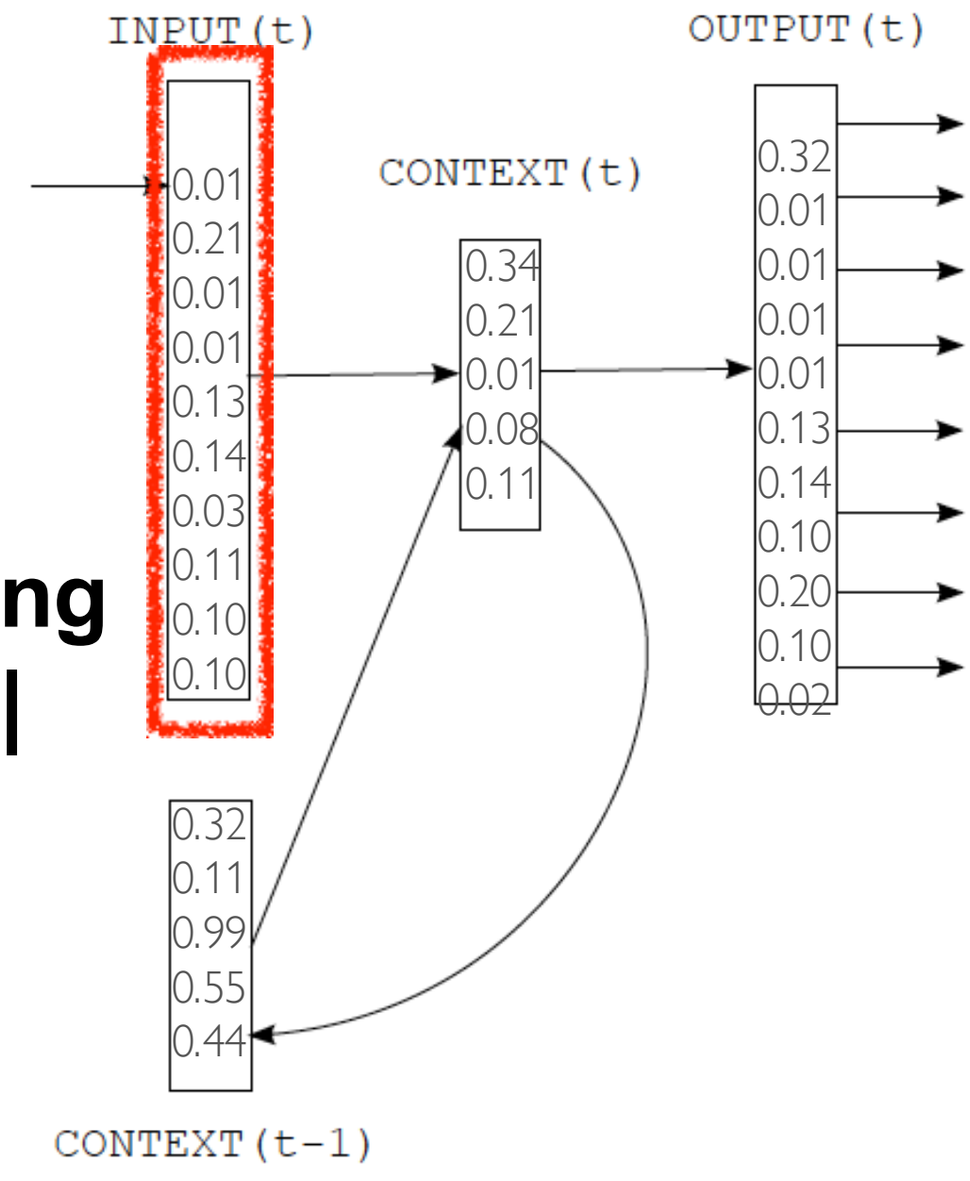
f e rpS-2 f f

rpESub

john

Wt

Indices map to **word embedding** of dimension **|e|**



RNNS FOR DISFLUENCY TAGGING

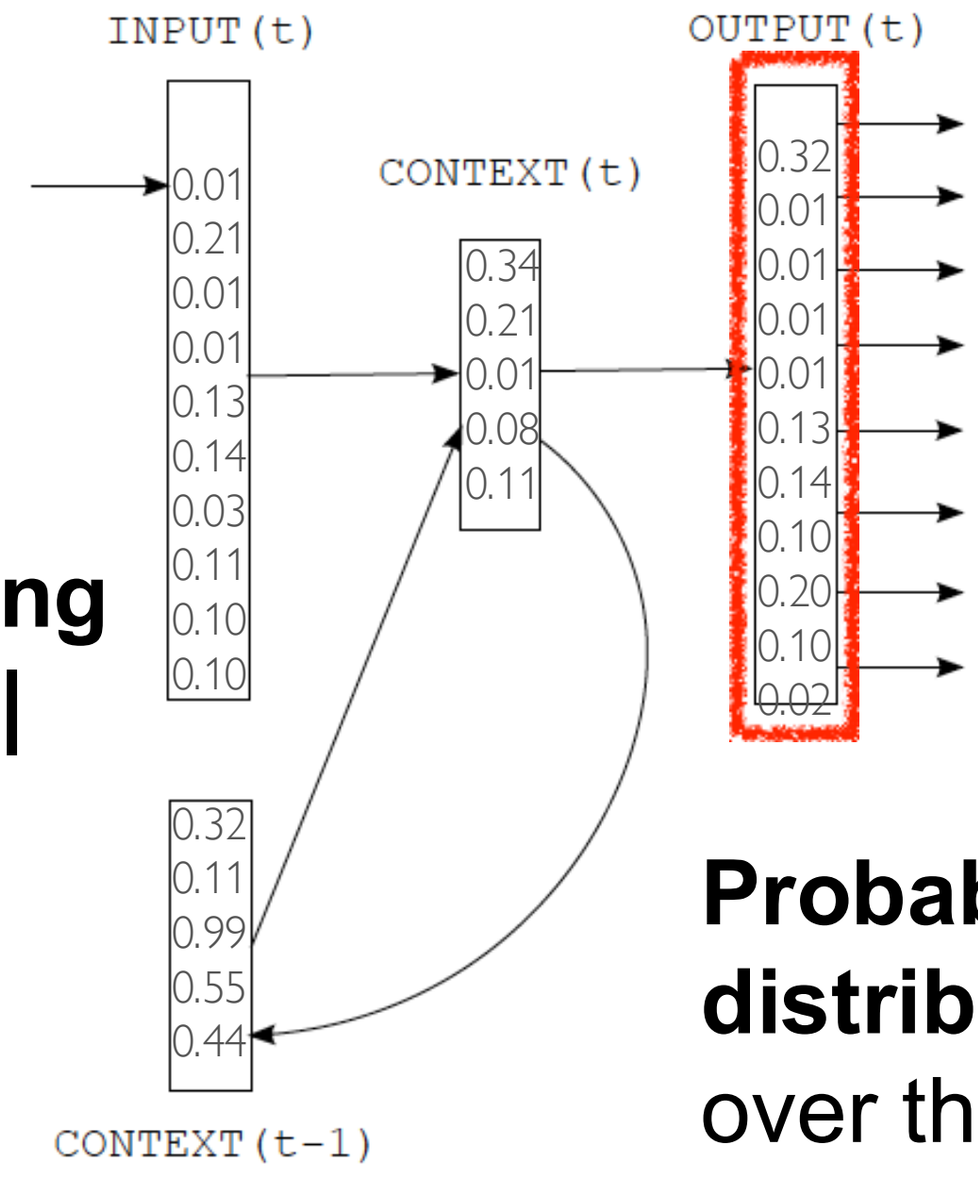
[john, + { uh, } john] likes mary

f e rpS-2 f f

rpESub

john

Wt



Indices map to **word embedding** of dimension **|e|**

Probability distribution over the 1-of-N positions of tags vector of size **|Tags|**

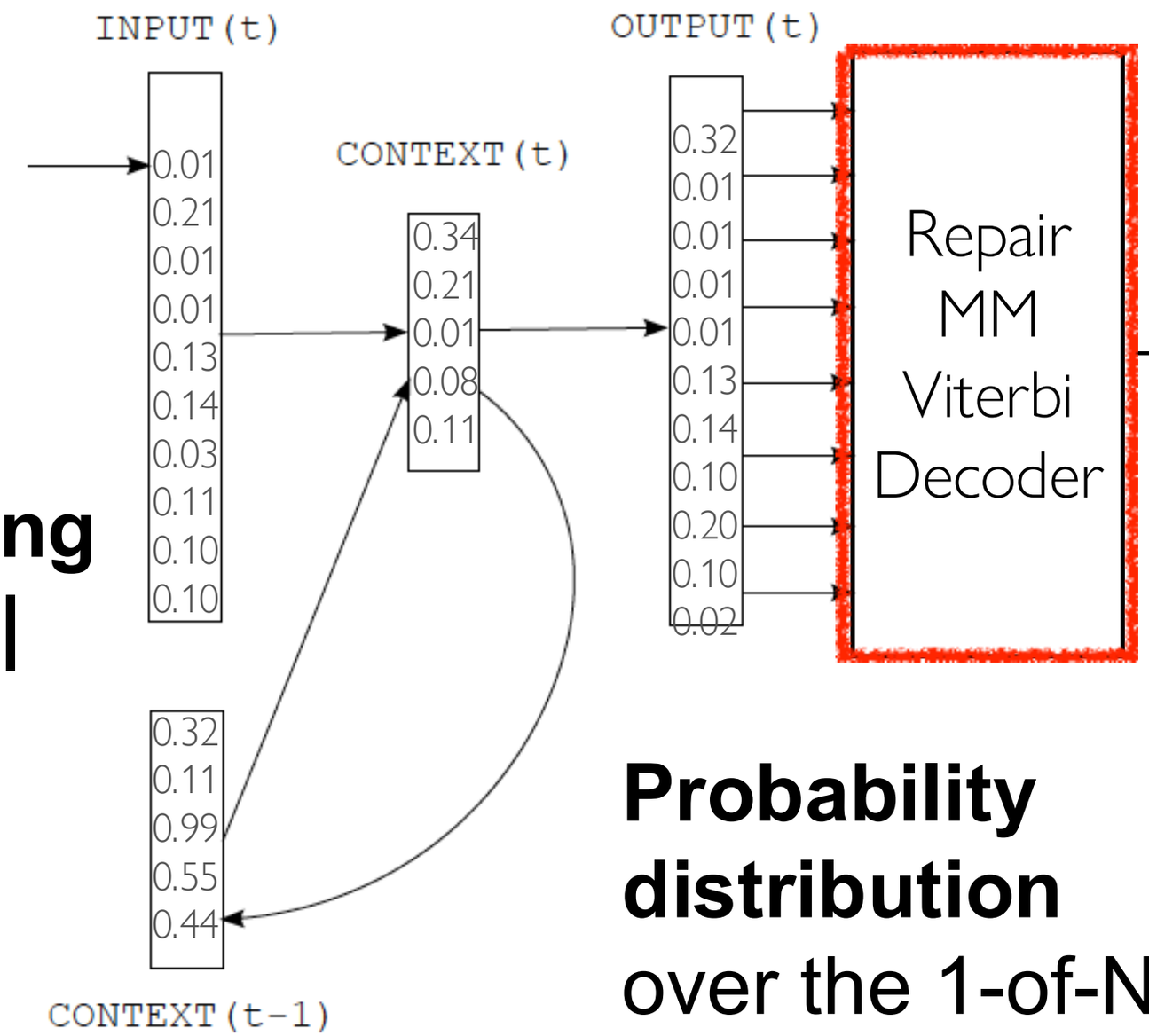
RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

f e rpS-2 rpESub f f

john

w_t



tag_t

Indices map to **word embedding** of dimension **|e|**

Probability distribution over the 1-of-N positions of tags vector of size **|Tags|**

RNNS FOR DISFLUENCY TAGGING

[john, + { uh, } john] likes mary

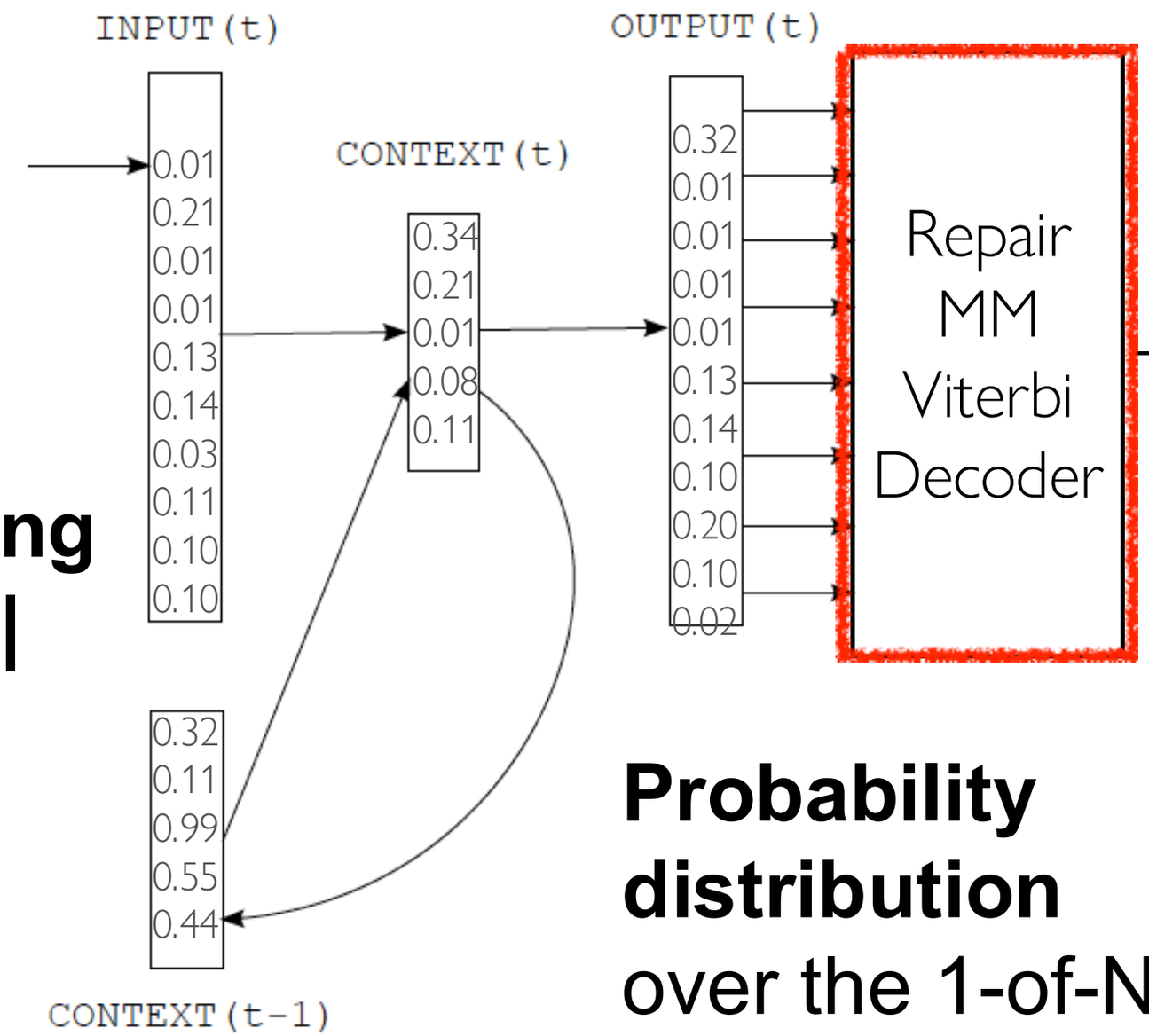
f e rpS-2 f f

rpESub

john

w_t

Indices map to **word embedding** of dimension **|e|**



rpS-2

rpESub

tag_t

Probability distribution over the 1-of-N positions of tags vector of size **|Tags|**

RNNS FOR DISFLUENCY TAGGING

HYPER-PARAMETERS

- ***Architecture:*** Elman (Forward)
- ***Size of hidden layer:*** 50 nodes
- ***Dimension of word embeddings:*** R^{50}
- ***Embeddings:*** pre-trained on cleaned Switchboard data using Word2Vec, fine-tuned during training
- ***Activation functions:*** Hidden layer activation ***F* = sigmoid** and output layer function ***G* = soft max**

RNNS FOR DISFLUENCY TAGGING

HYPER-PARAMETERS

- ***Error function:*** Neg. Loss Likelihood over tag set
- ***Learning rate:*** 0.005, fixed
- ***Batch size****:* 1 X <*word window*, *tag*> (with context up to 9 words back available for back-prop through time)

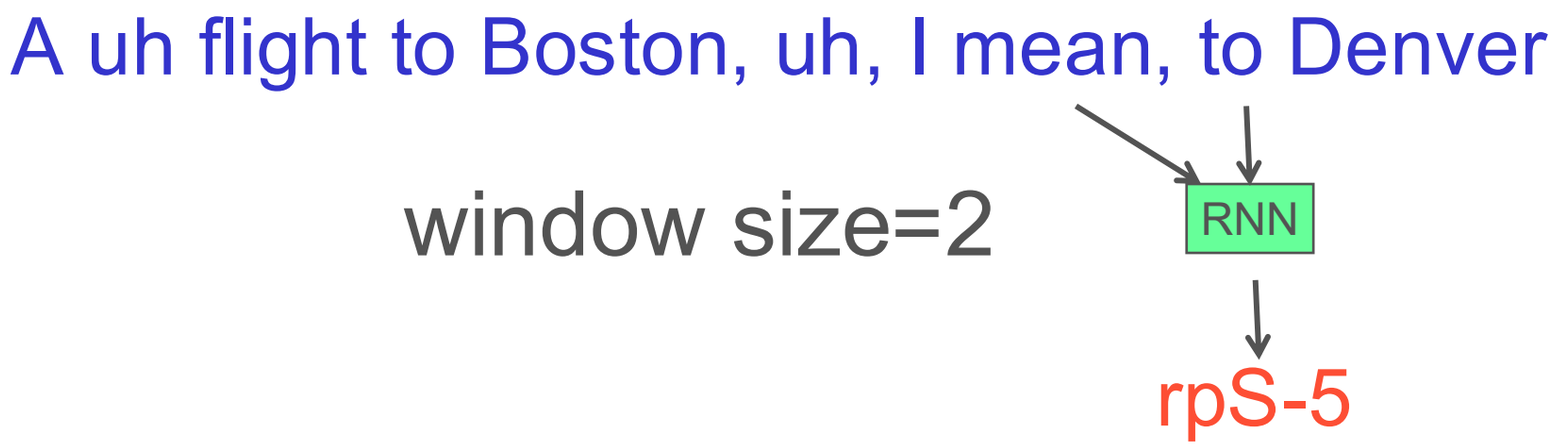
RNNS FOR DISFLUENCY TAGGING

EXPERIMENTS

RNNS FOR DISFLUENCY TAGGING

EXPERIMENTS

- We experiment with different **context window lengths** for each input step (concatenated word embeddings of words strictly backwards from the current word, no lookahead!). E.g.:



- And also with using POS tags concatenated to the input embeddings or not.We found using **POS tags + 2-word windows** worked best (better than 3-word input windows).

RNNS FOR DISFLUENCY TAGGING

EXPERIMENTS

- ***Detection performance***
  Accuracy: **F-score reparandum words and edit term words**

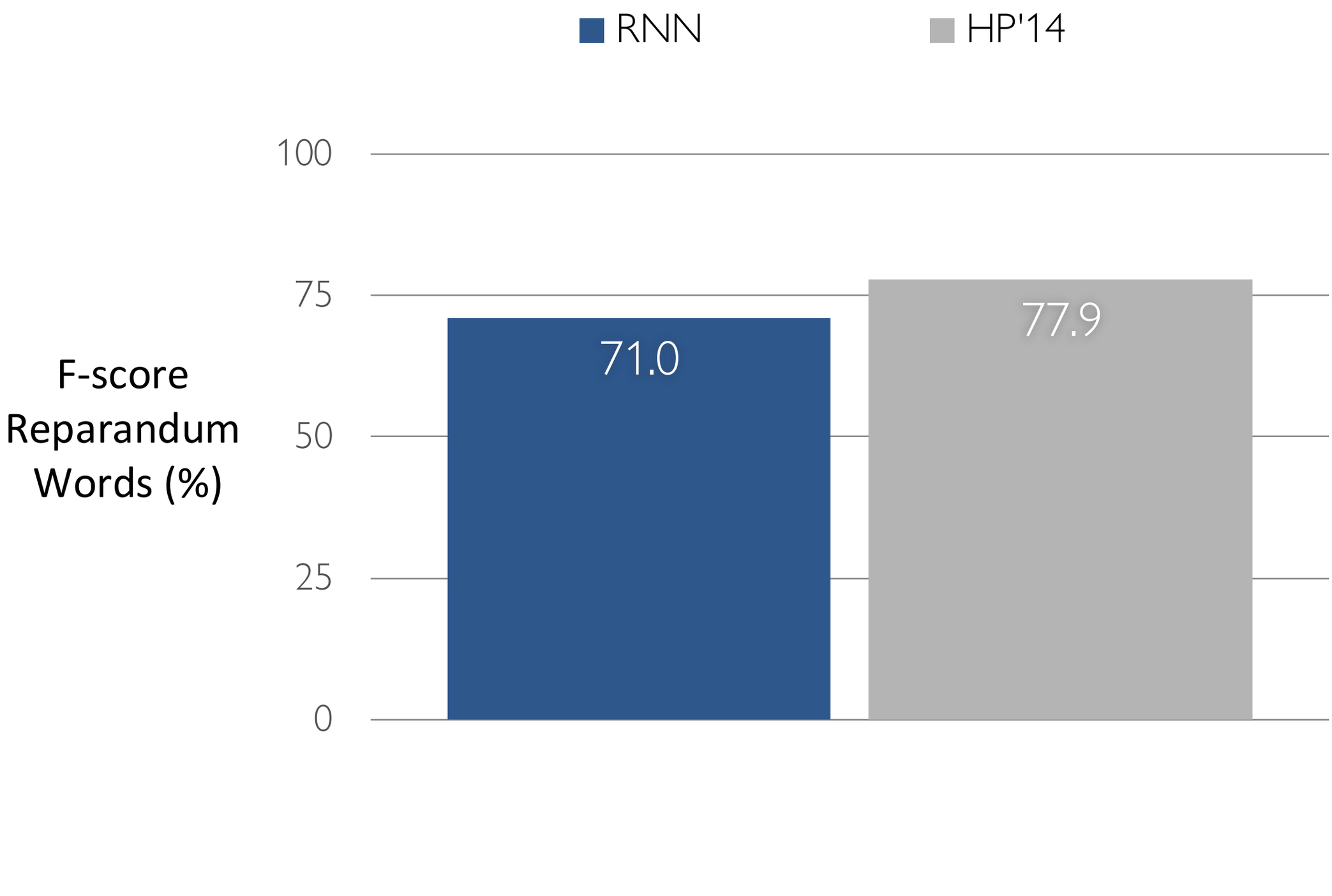
  ***Incremental metrics*:**
  - **Time-to-detection (latency from end of target word)**
  - **Edit Overhead (jitter in output)**

- ***Data:*** *Switchboard train/dev/test (Johnson and Charniak 2004). Transcripts with partial words and punctuation removed.*

- ***Stopping criterion:*** *No improvement after 10 epochs on dev set*

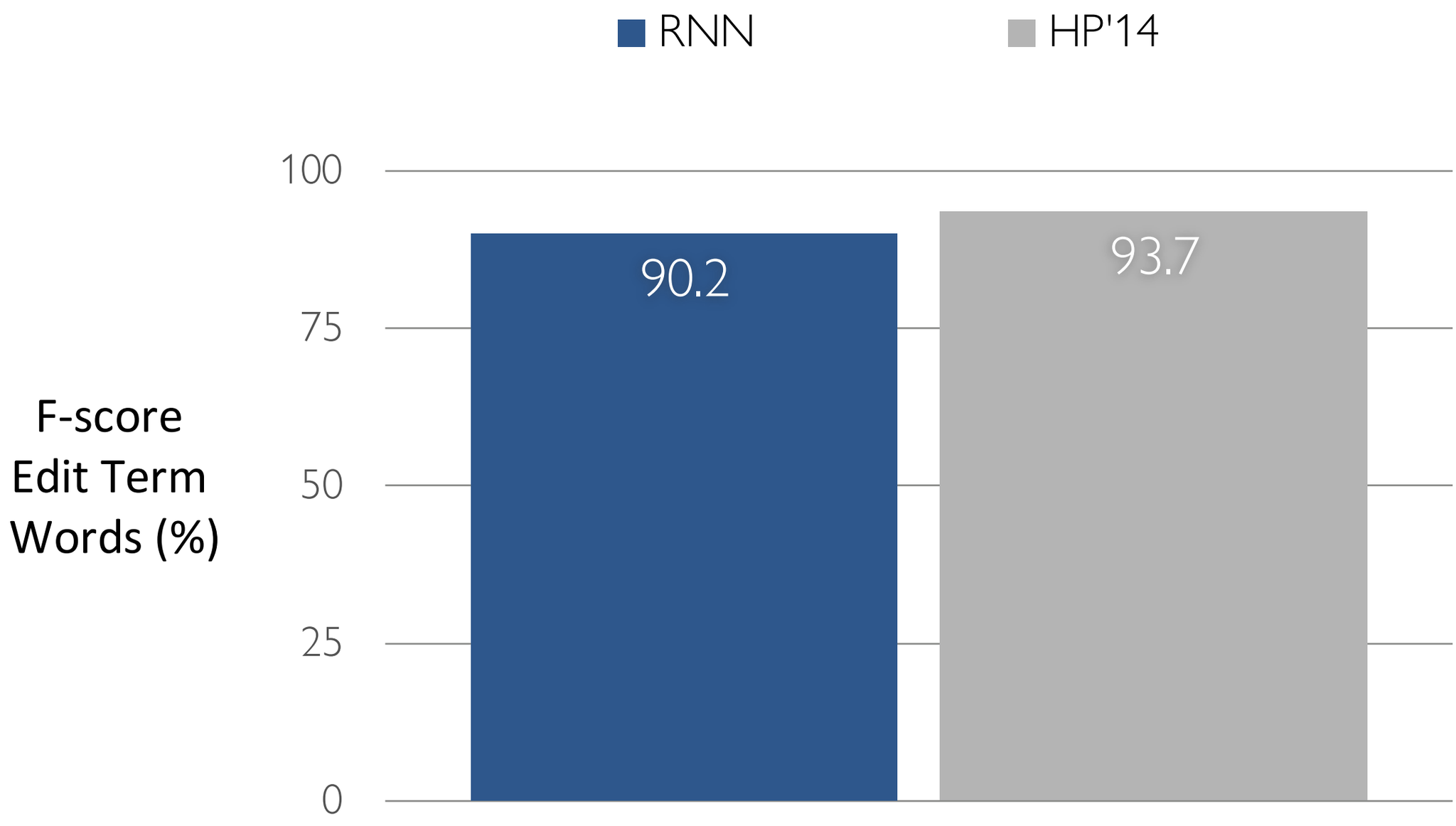
RNNS FOR DISFLUENCY TAGGING

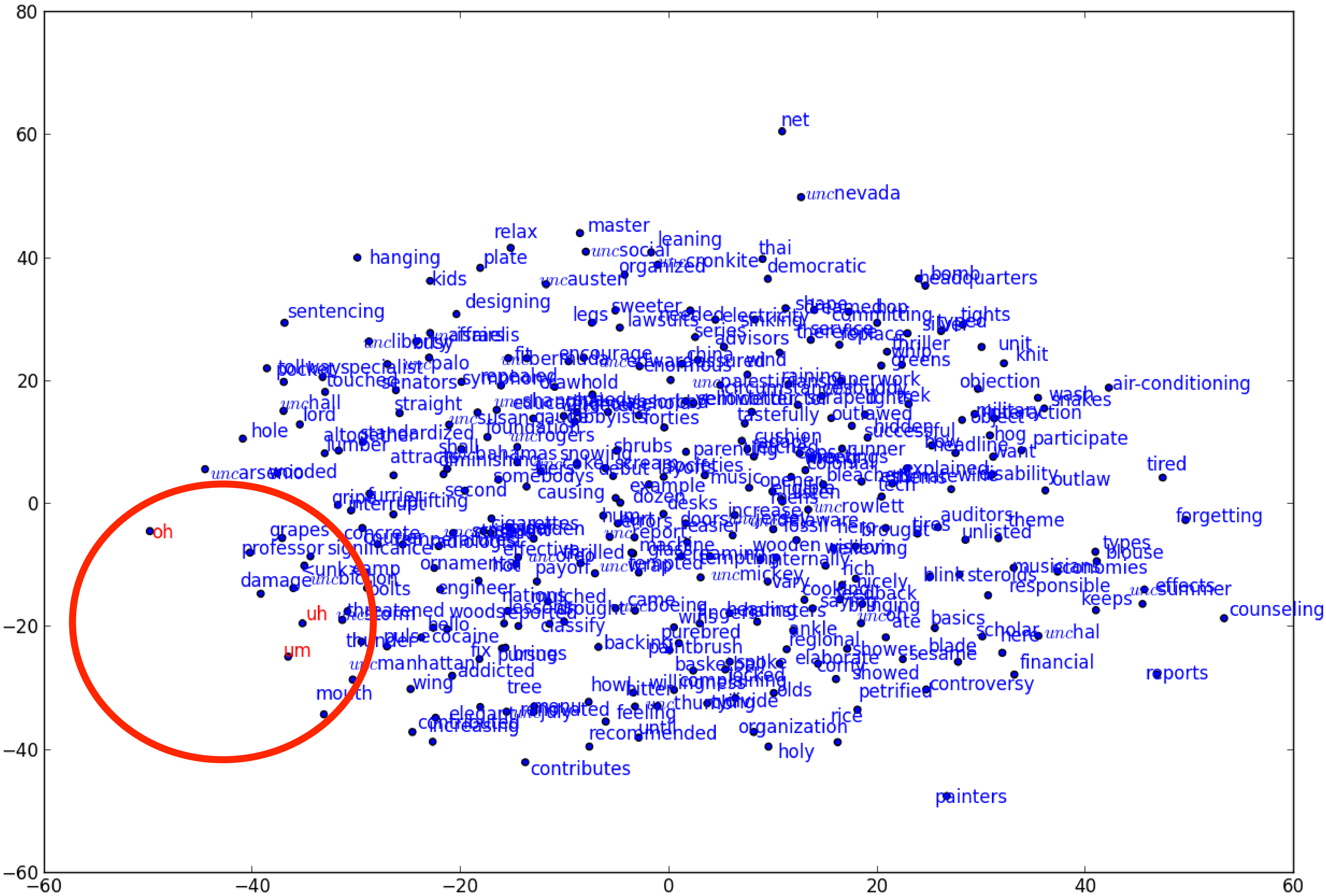
RESULTS: REPAIR DETECTION



RNNS FOR DISFLUENCY TAGGING

RESULTS: EDIT TERM DETECTION

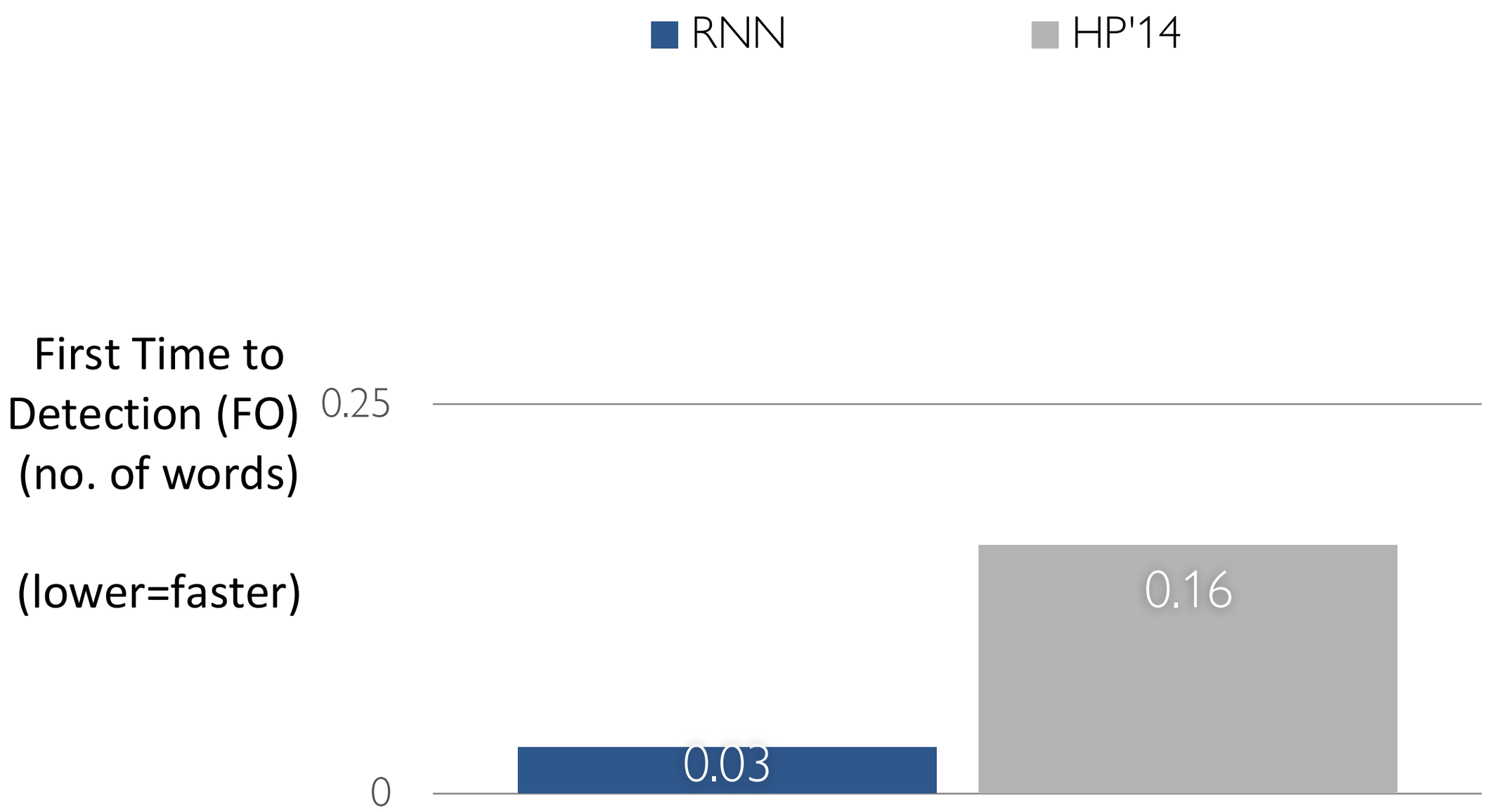




80
60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60
net
nevada
relax
master
leaning
thai
democratic
hanging
plate
kids
austen
headquarters
designing
sentencing
tights
unit
knit
objection
air-conditioning
hole
lord
straight
participate
tired
outlaw
forgetting
theme
unlisted
counseling
types
blouse
effects
summer
reports
financial
controversy
painters
holy
organization
rice
petrified
contributes
recommended
feeling
tree
wing
addicted
mouth
grapes
professor
damage
oh
uh
um
hello
classify
engineer
causing
somebody
second
example
dozen
desks
rowlett
auditors
responsible
keeps
basics
scholar
hal
regional
ankle
purebred
showed

RNNS FOR DISFLUENCY TAGGING

RESULTS: INCREMENTALITY



RNNS FOR DISFLUENCY TAGGING

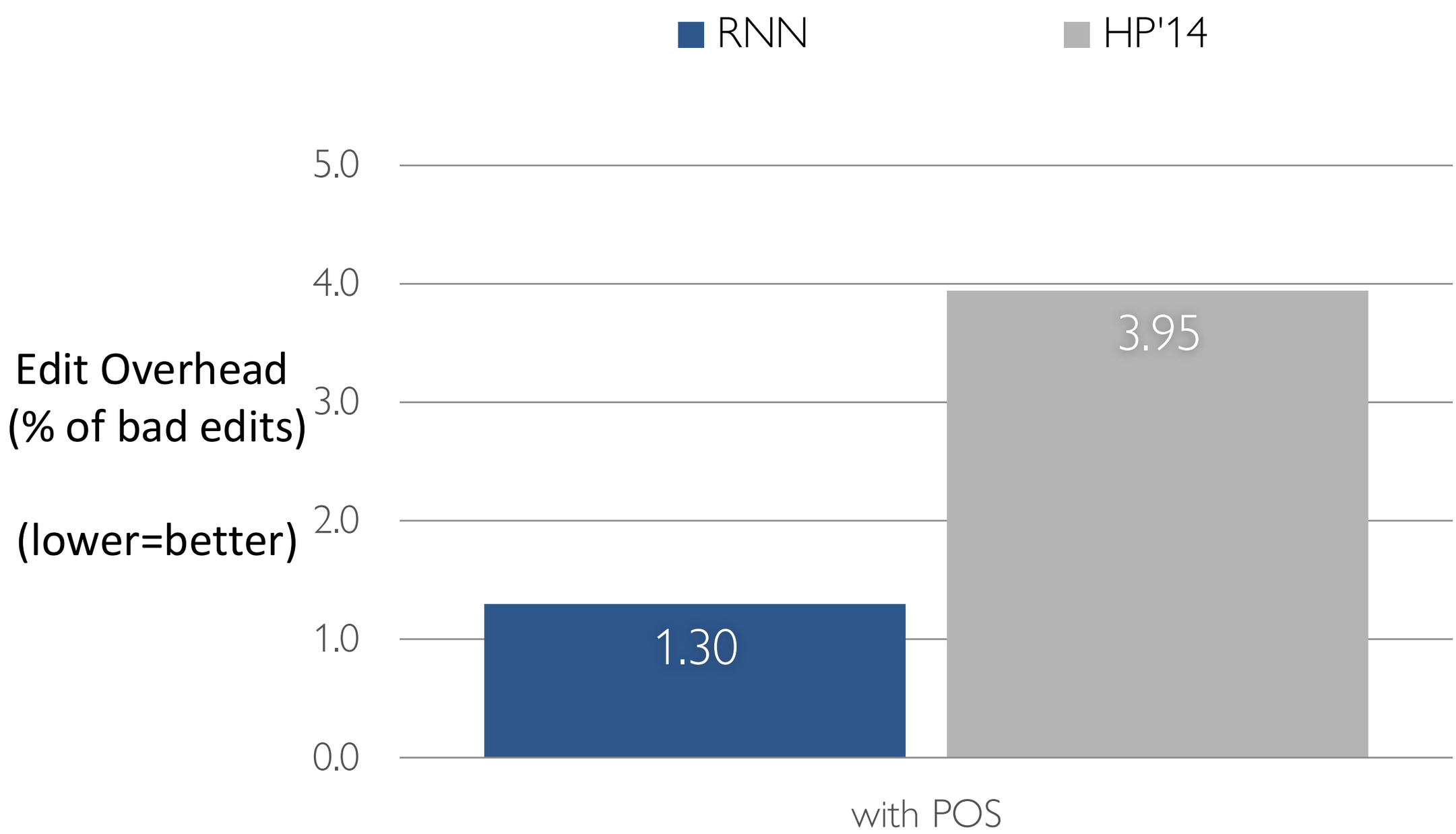
RESULTS: INCREMENTALITY

- ***Edit Overhead = the amount of jitter***

Input and current repair labels	edits
John	
John likes rm rp	$(\oplus rm)\,(\oplus rp)$
John likes uh ed	$(\ominus rm)\,(\ominus rp)\,\oplus ed$
John likes uh loves rm ed rp	$\oplus rm\ \oplus rp$
John likes uh loves Mary rm ed rp	

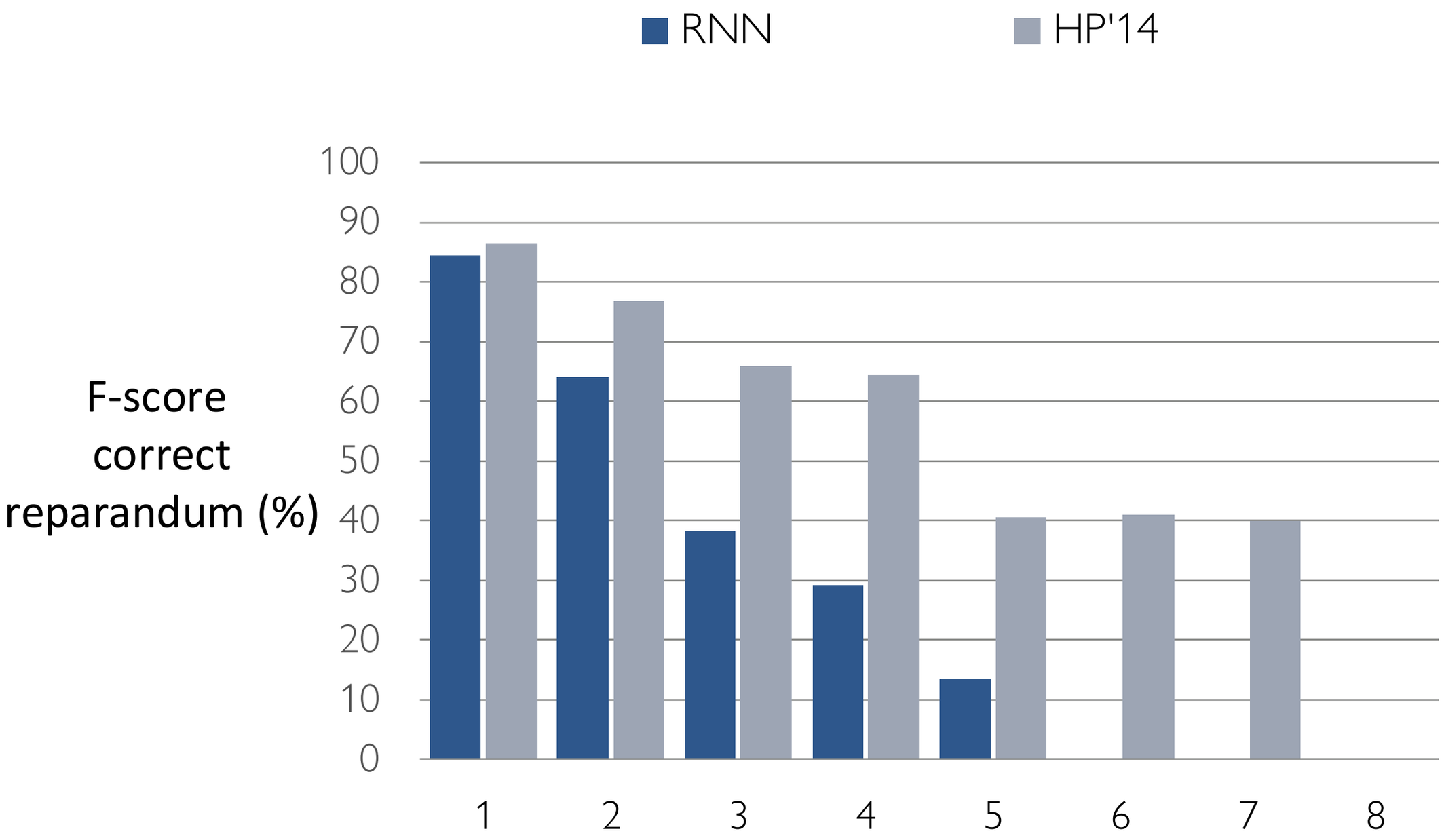
RNNS FOR DISFLUENCY TAGGING

RESULTS: INCREMENTALITY



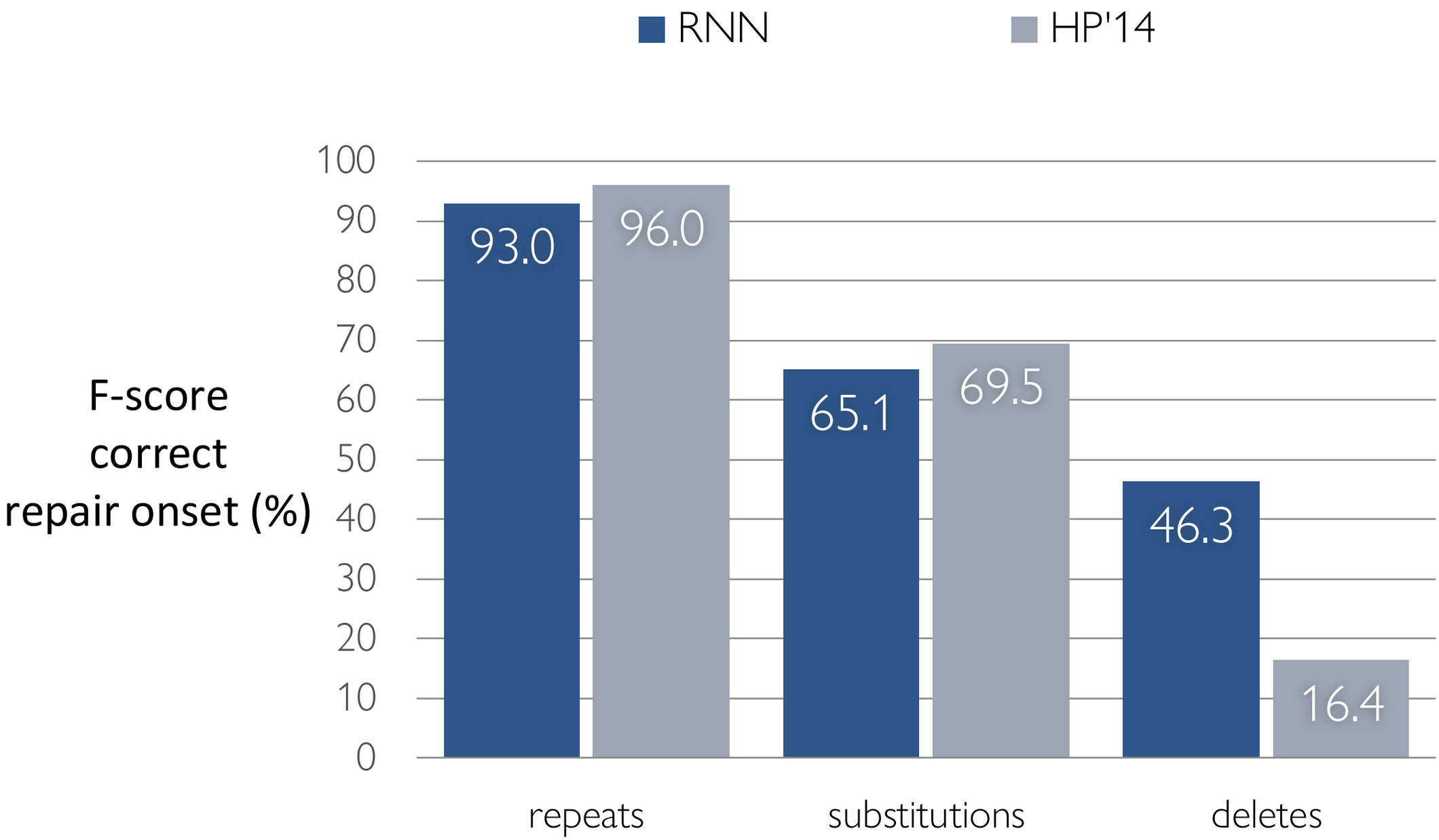
RNNS FOR DISFLUENCY TAGGING

ERROR ANALYSIS: DIFFERENT REPAIR LENGTHS



RNNS FOR DISFLUENCY TAGGING

ERROR ANALYSIS: DIFFERENT REPAIR TYPES



RNNS FOR DISFLUENCY TAGGING

DISCUSSION

- Treating disfluency detection as left-right tagging task can work well without other structural information/feature engineering.

- RNNs a promising method: lightweight/reproducible, good accuracy, very good incrementally for dialogue systems.

- Poor performance on longer length repairs suggests **vanishing gradient problem** (de Mulder et al., 2015)- use of LSTM may help.

- *Given it is trained and tested on pre-segmented transcript data, is it really suitable for **live use**? Need to do this on ASR results.*

CONTENTS

LEFT-RIGHT UTTERANCE SEGMENTATION

| A uh flight [to Boston + { uh I mean } to Denver] on Friday | Thank you |

.w- -w- -w- -w- -w- -w- -w- -w- -w- -w- -w- -w. .w- -w.

Tag	Meaning
.w-	Beginning of an utterance (B)
-w-	In-utterance word (I)
-w.	End-utterance word (E)
.w.	Single word utterance (S)

LEFT-RIGHT UTTERANCE SEGMENTATION

Tag	% of words
-w-	76
.w-	10
-w.	10
.w.	4

Not too sparse ☺

LEFT-RIGHT JOINT TASK TAGGING

CONSTRAINTS ON TAG SETS:

- **C1** Repair onsets cannot begin an utterance/dialogue act (by definition of first position repairs needing a preceding reparandum).

- **C2** Repairs must be completed within the utterance/dialogue act in which they begin.

- **C3** Utterances/dialogue acts can be interrupted or abandoned, but these are different to within-dialogue-act repairs.

LEFT-RIGHT JOINT TASK TAGGING

TAG SETS:

A	uh	flight	[	to Boston +	{ uh	I	mean }	to	Denver]	on Friday		Thank you
.f-	*-e-*	*-f-*		*-f- -f-*	*-e-*	*-e-*	*-e-*	*-rpS−5-*	*-rpESub-*	*-f-*	*-f.*	*.f-* *-f.*

	-w-	-w.	.w-	.w.
f	1	1	1	1
e	1	1	1	1
rpS-[1-8]	1	0	0	0
rpMid	1	0	0	0
rpESub	1	1	0	0
rpEDel	1	1	0	0
rpS-[1-8]ESub	1	1	0	0
rpS-[1-8]EDel	1	1	0	0

LEFT-RIGHT JOINT TASK TAGGING

TAG SETS:

A	uh	flight	[	to Boston	+	{	uh	I	mean	}	to	Denver]	on Friday	|	Thank you	|
.f-	-e-	-f-		-f-	-f-		-e-	-e-	-e-		-rpS-	-f-	-f-	-f.	.f-	-f.

	-w-	-w.	.w-	.w.
f	1	1	1	1
e	1	1	1	1
rpS	1	1	0	0

LEFT-RIGHT JOINT TASK TAGGING

PROCESSING SPEECH WITH TIMING

- From incremental ASR results we can use the timing information which should help utterance segmentation.

DNNS FOR JOINT TASK TAGGING

DNNS FOR JOINT TASK TAGGING

RESEARCH QUESTIONS:

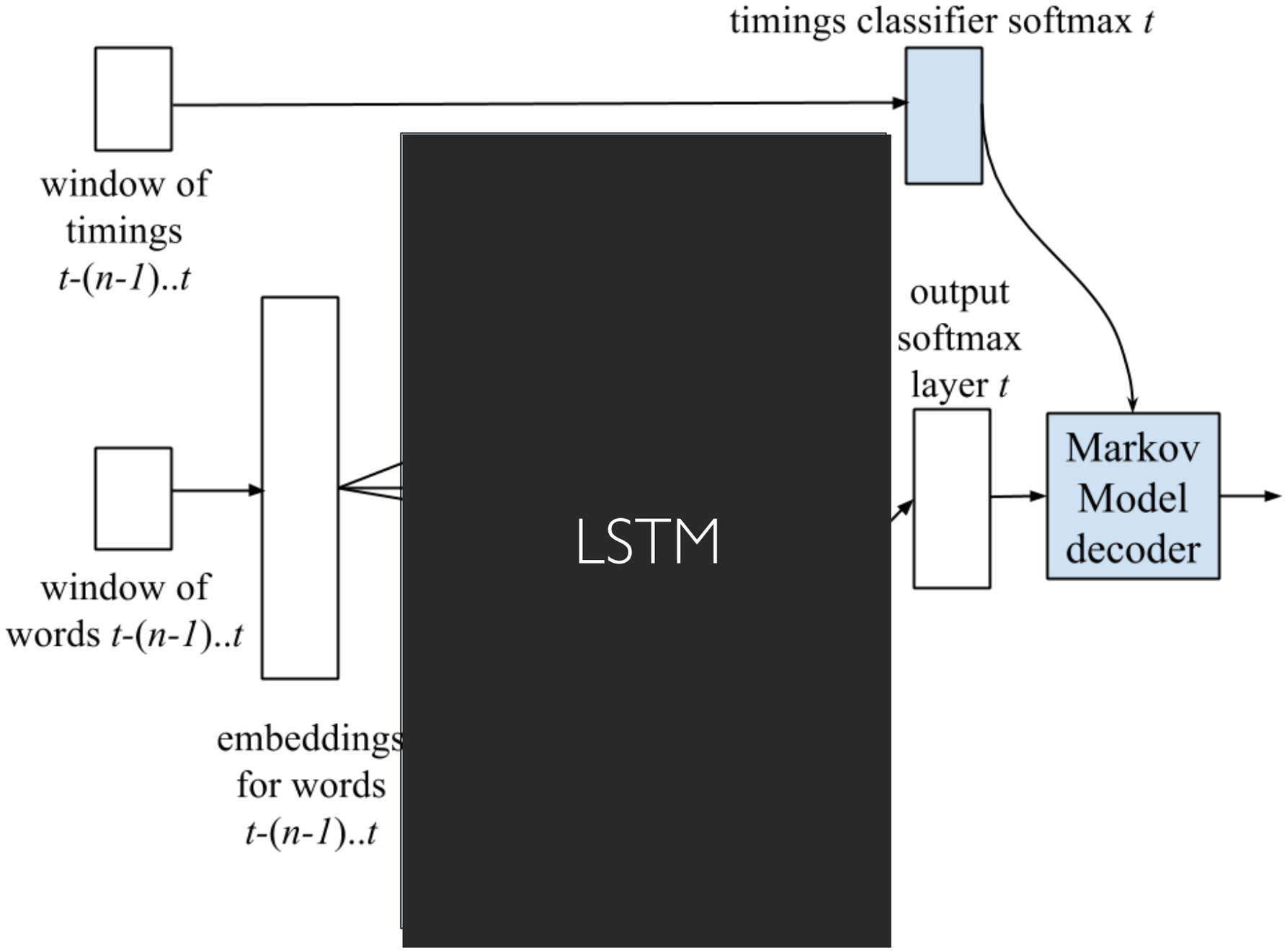
- **Q1** Can a deep neural net (DNN) system which performs both **jointly** help improve equivalent systems doing the individual tasks?

- **Q2** To what extent does an **LSTM** improve over an equivalent RNN, particularly for **longer** repairs?

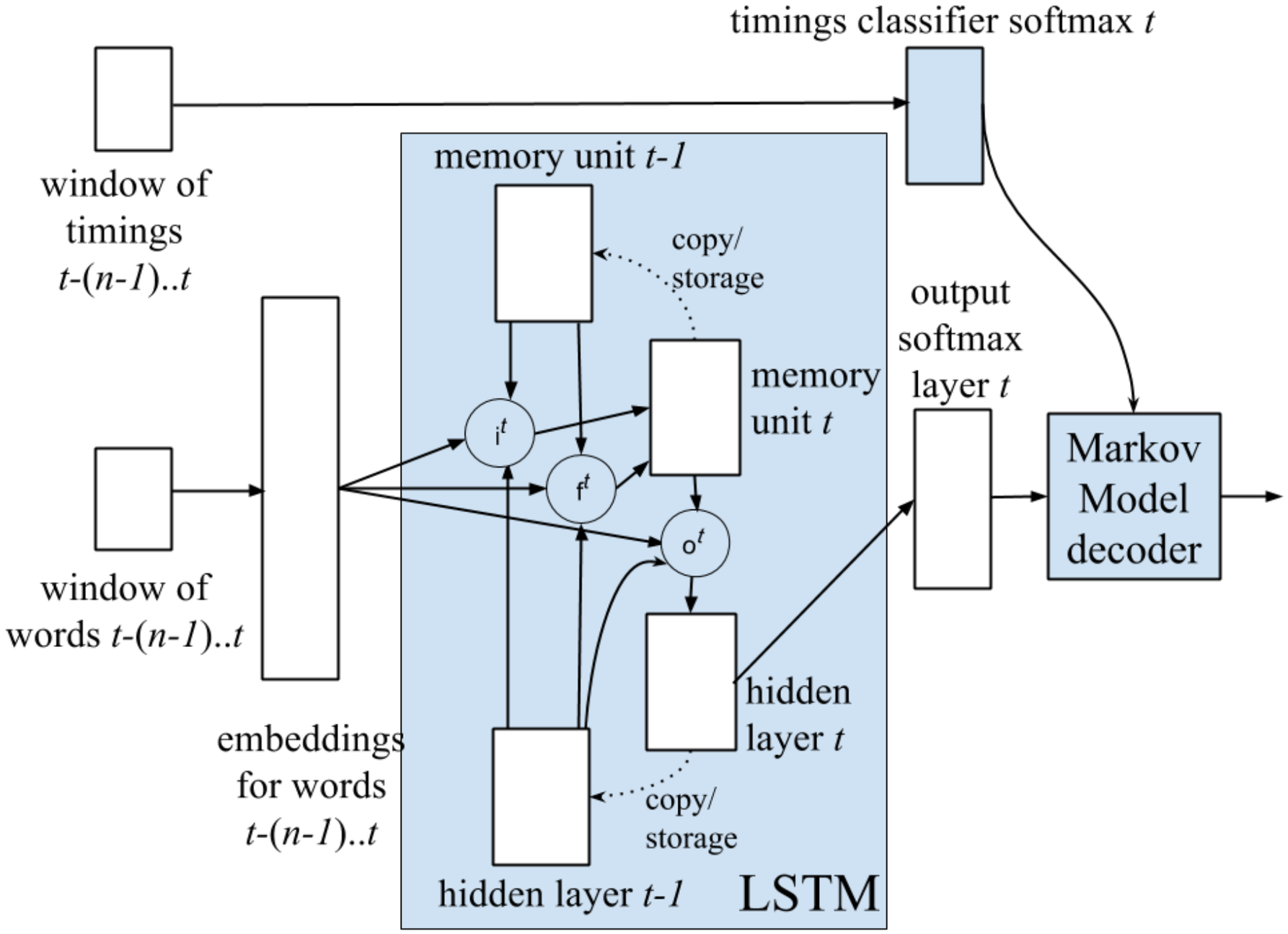
- **Q3** To what extent can **word timing data** help performance on either task?

(Hough and Schlangen, 2017 EACL)

DNNS FOR JOINT TASK TAGGING



DNNS FOR JOINT TASK TAGGING



DNNS FOR JOINT TASK TAGGING

HYPER-PARAMETERS

- ***Architecture:*** Elman (Forward) RNN or LSTM
- ***Size of hidden layer:*** 50 nodes
- ***Dimension of word embeddings:*** R^{50}
- ***Embeddings:*** pre-trained on cleaned Switchboard data using Word2Vec, fine-tuned during training

DNNS FOR JOINT TASK TAGGING

EXPERIMENTS

DNNS FOR JOINT TASK TAGGING

EXPERIMENTS

- ***Independent variables (12 conditions):***
  - *Tag set:* Disf only vs. Utt. seg. only vs. joint (3)
  - *Input:* Words & POS with timings vs. without timing (2)
  - *Architecture:* Elman RNN vs. LSTM (2)

- ***Dependent variables (performance):***
  - ***Accuracy:*** **F-score repair onset and edit word detection. Utterance segmentation accuracy. Correlation of repair rate against gold per speaker.**
  - ***Incremental performance*: Time-to-detection (latency), Edit Overhead (jitter in output).**

DNNS FOR JOINT TASK TAGGING

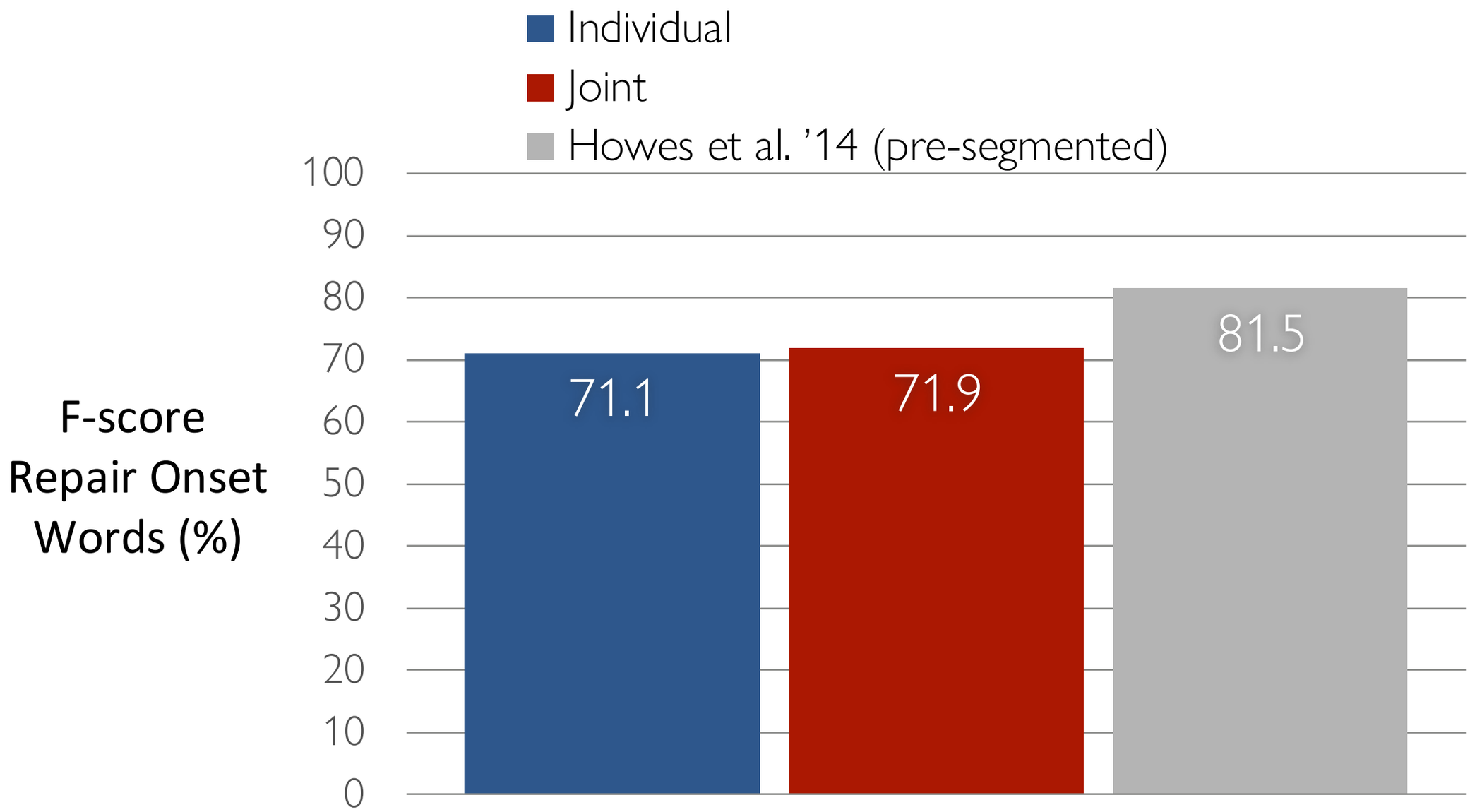
EXPERIMENTS

- ***Data:*** Switchboard train/dev/test (Johnson and Charniak 2004)

- ***ASR results*:** from IBM Watson trial system. Google filters out disfluencies already (Baumann et al, 2017)

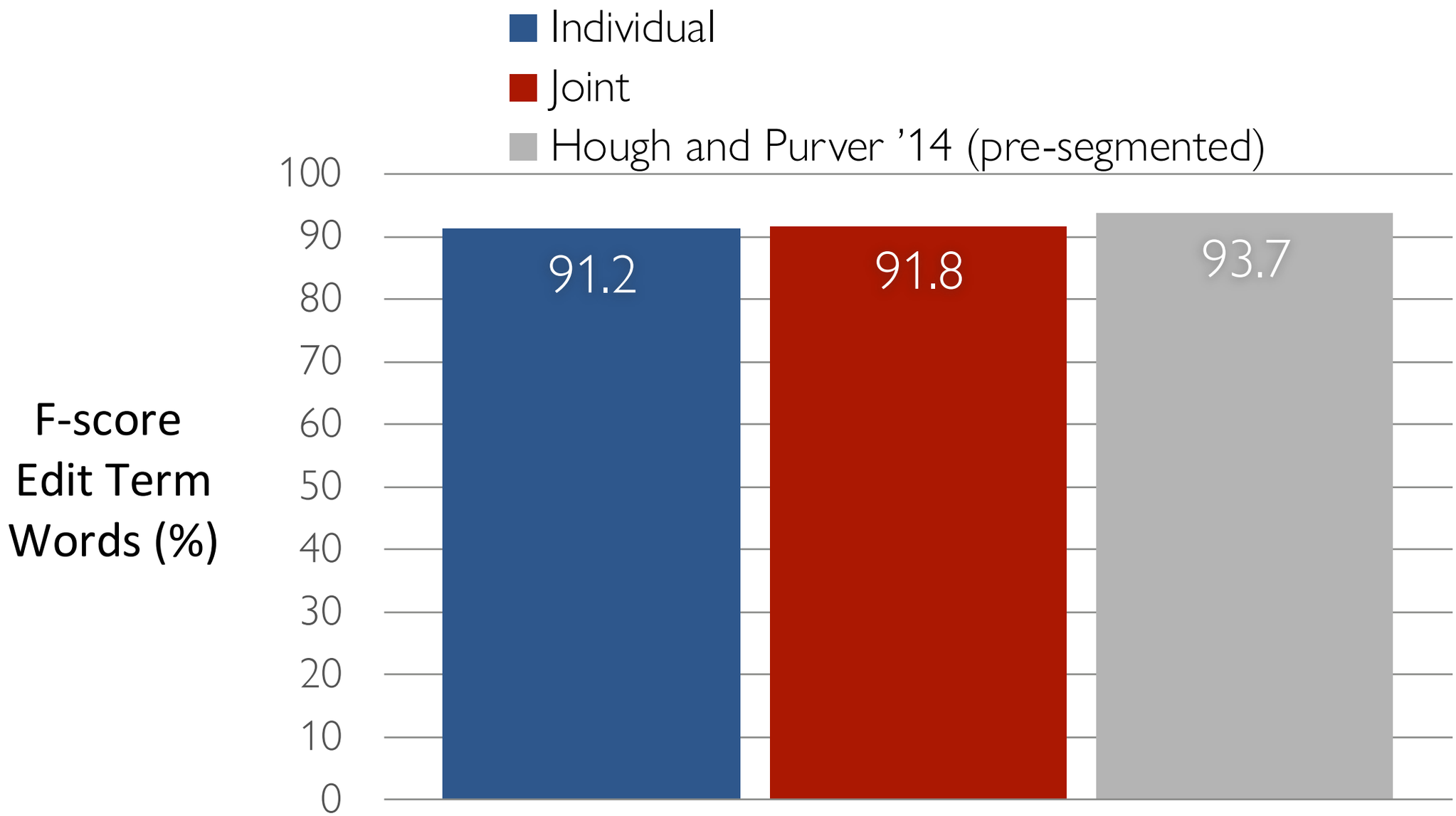
DNNS FOR JOINT TASK TAGGING

RESULTS: REPAIR DETECTION



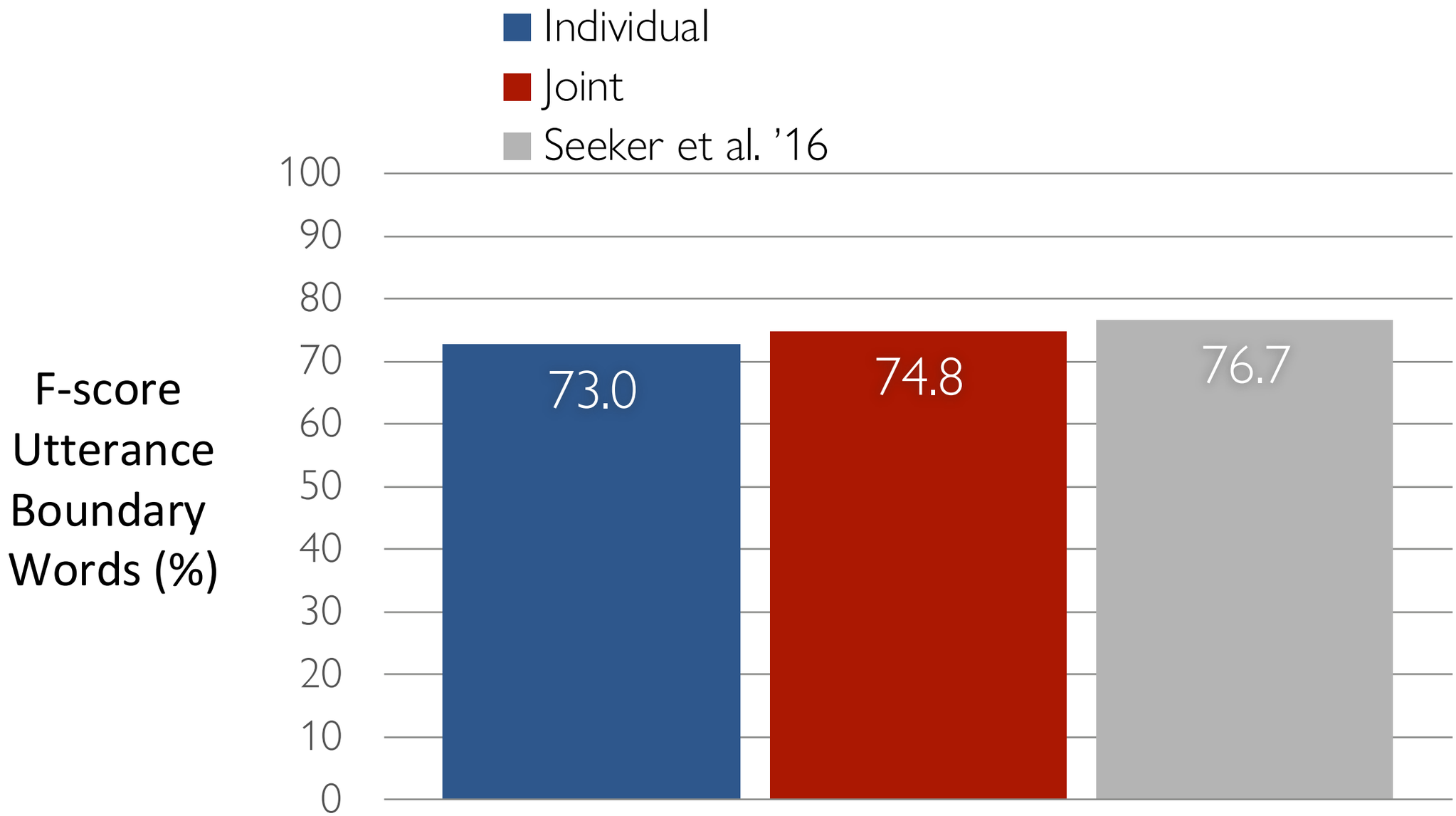
DNNS FOR JOINT TASK TAGGING

RESULTS: EDIT TERM DETECTION



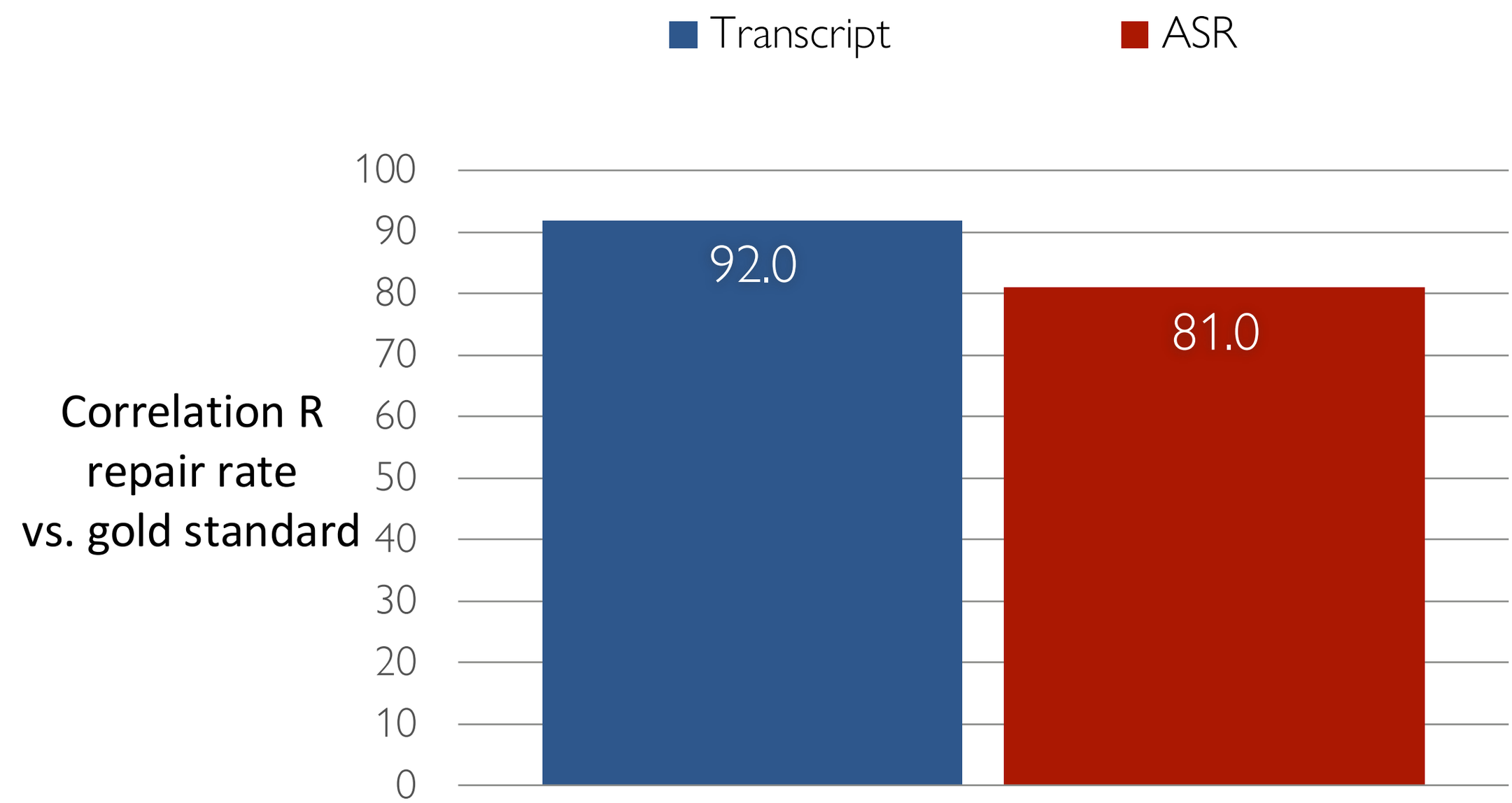
DNNS FOR JOINT TASK TAGGING

RESULTS: UTTERANCE SEGMENTATION



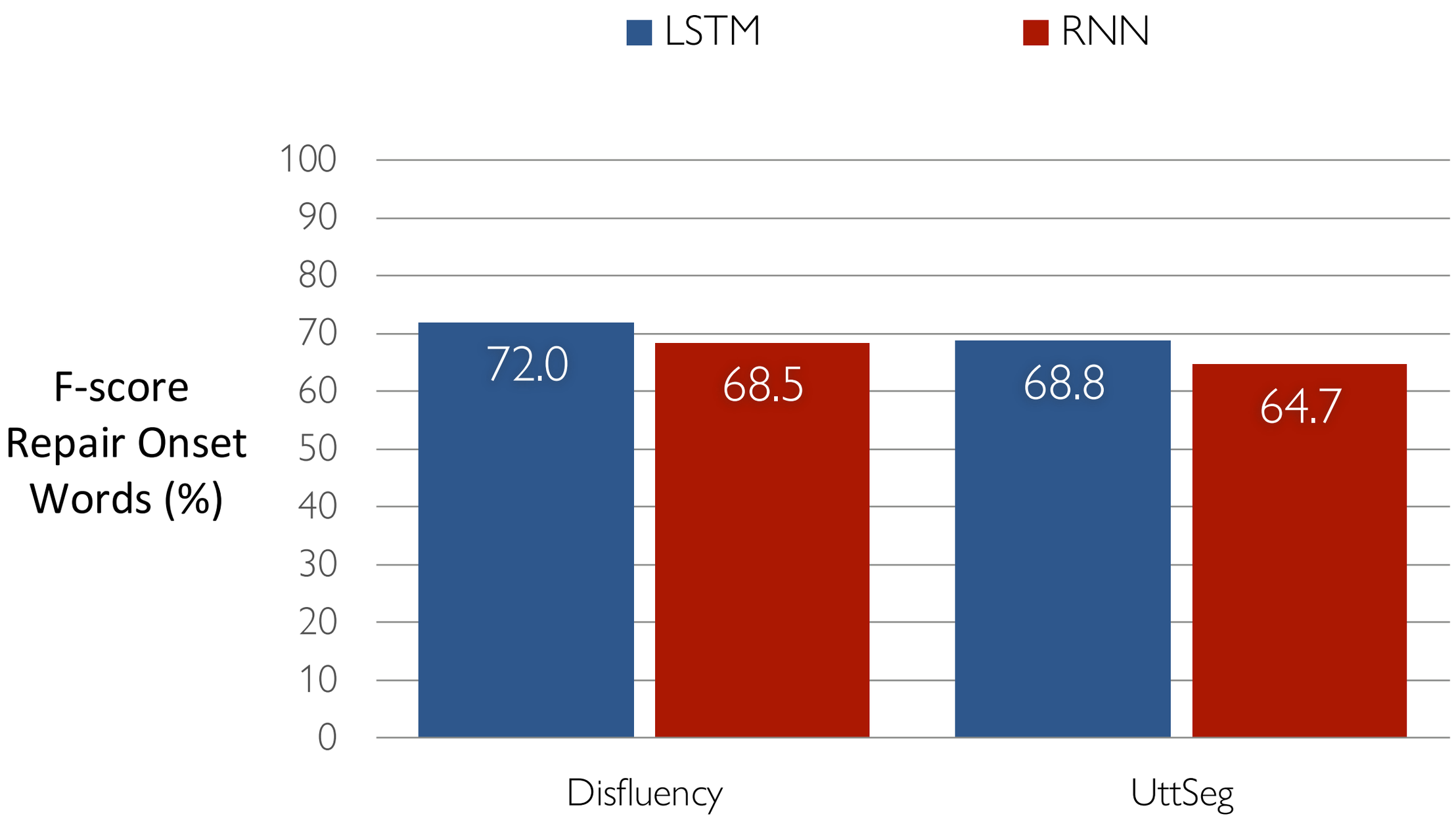
DNNS FOR JOINT TASK TAGGING

RESULTS: REPAIR RATE CORRELATION (VS ASR)



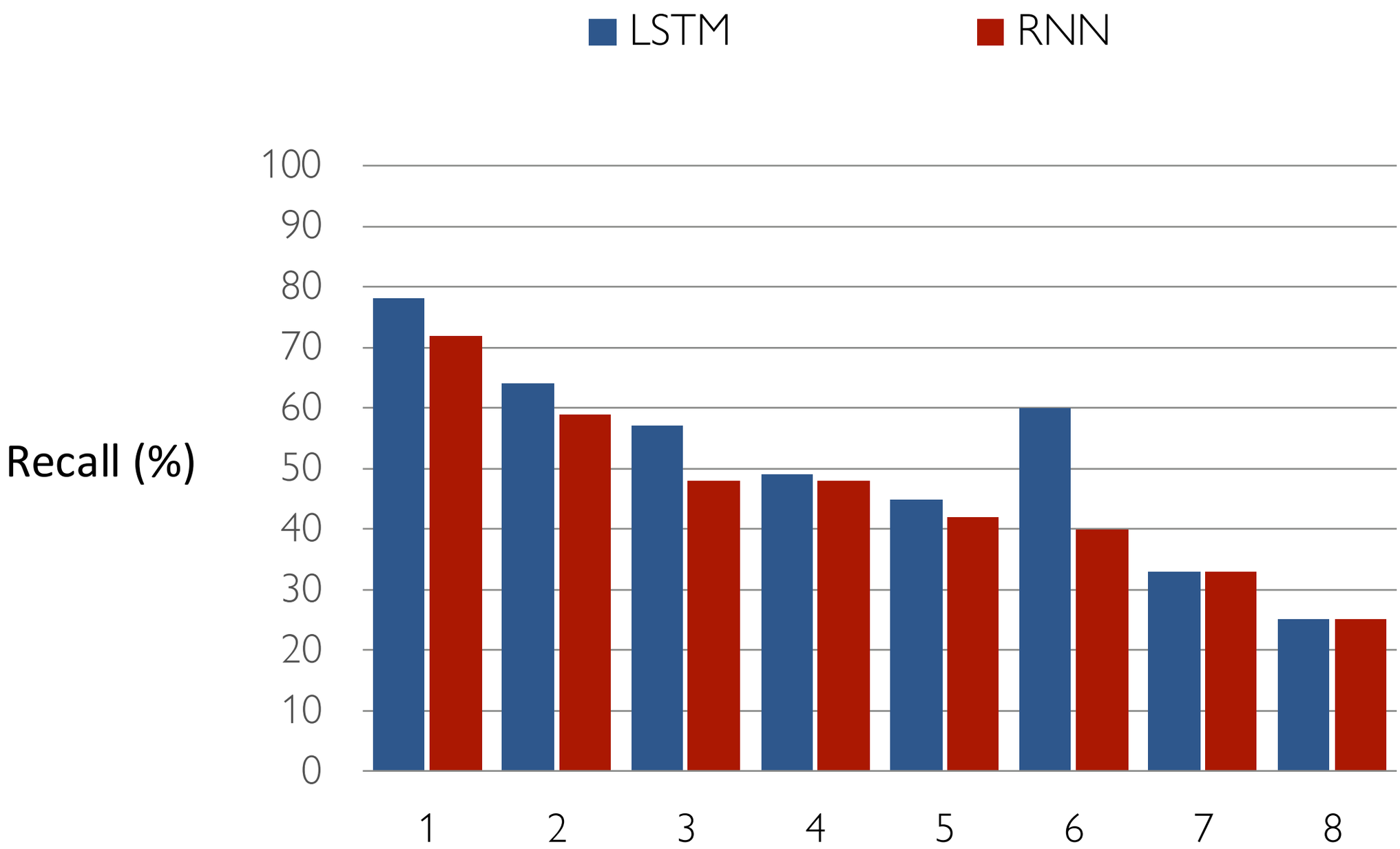
DNNS FOR JOINT TASK TAGGING

RESULTS: REPAIR DETECTION (ARCHITECTURE)



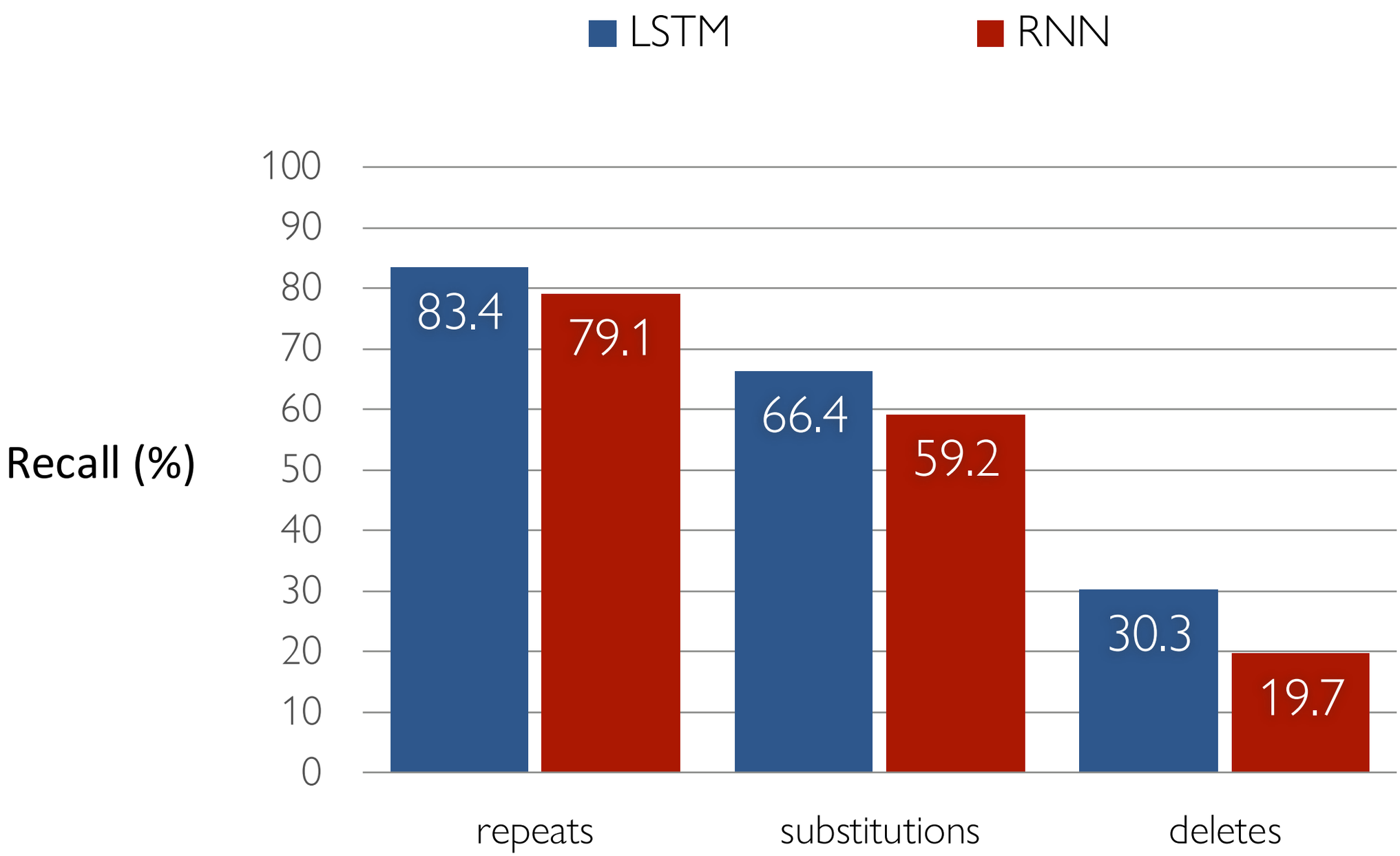
DNNS FOR JOINT TASK TAGGING

ERROR ANALYSIS (ARCHITECTURE): REPAIR ONSET DETECTION FOR DIFFERENT REPAIR LENGTHS



DNNS FOR JOINT TASK TAGGING

ERROR ANALYSIS (ARCHITECTURE): REPAIR ONSET DETECTION FOR DIFFERENT REPAIR TYPES



DNNS FOR JOINT TASK TAGGING

DISCUSSION

- Repair detection competitive with state-of-the-art despite no segmentation.
- Edit term and utterance segmentation detection at the state-of-the-art.
- **Joint task improves over individual ones**: utt seg is particularly improved when using disfluency detection.
- Performance on ASR for repair rate correlation remains good.
- **Memory** still a problem for learning longer and rarer repairs.
- **Timing info** invariably improves utterance segmentation, not repair detection (see paper).

CONTENTS

MULTI-TASK LEARNING

- We clearly have benefits of a joint tag set, however we would want to include other tasks (e.g. POS tagging, LM modelling)- tagset becomes large.

- **Morteza Rohanian's** PhD project reformulates an old, good idea from Heeman and Allen (1999): recasting language modelling from word-based model alone to a joint optimization problem:

$$p(w_i \ | w_{0\ldots i-1}, pos_{0\ldots i-1,}, utt_{0\ldots i-1}, disf_{0\ldots i-1})$$

MULTI-TASK LEARNING

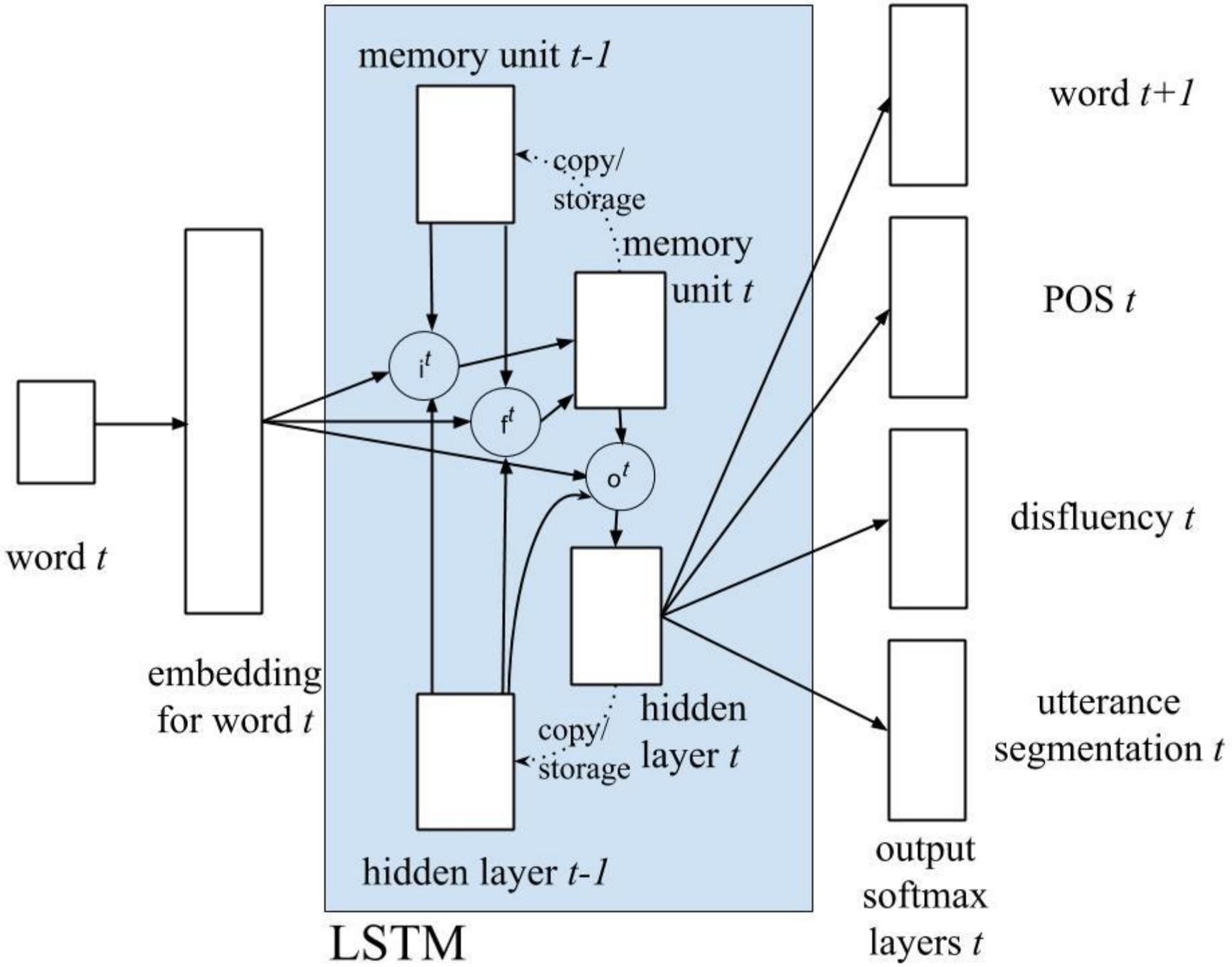
- In fact we reformulate language modelling, POS-tagging, utterance segmentation and disfluency tagging as one big **multi-task learning** set up:

$$p(w_i\ , pos_i\ , utt_i\ , disf_i\ |w_{0\ldots i-1}, pos_{0\ldots i-1,}, utt_{0\ldots i-1}, disf_{0\ldots i-1})$$

- We do this using an LSTM which predicts all four values at once for each time-step with **different error functions** for each task in training.

(Rohanian and Hough, COLING 2020)

MULTI-TASK LEARNING



MULTI-TASK LEARNING

- We experiment with different **loss functions** for training to see which is most effective – we try *naive* summing of log loss of all tasks and ***uncertainty* loss**:

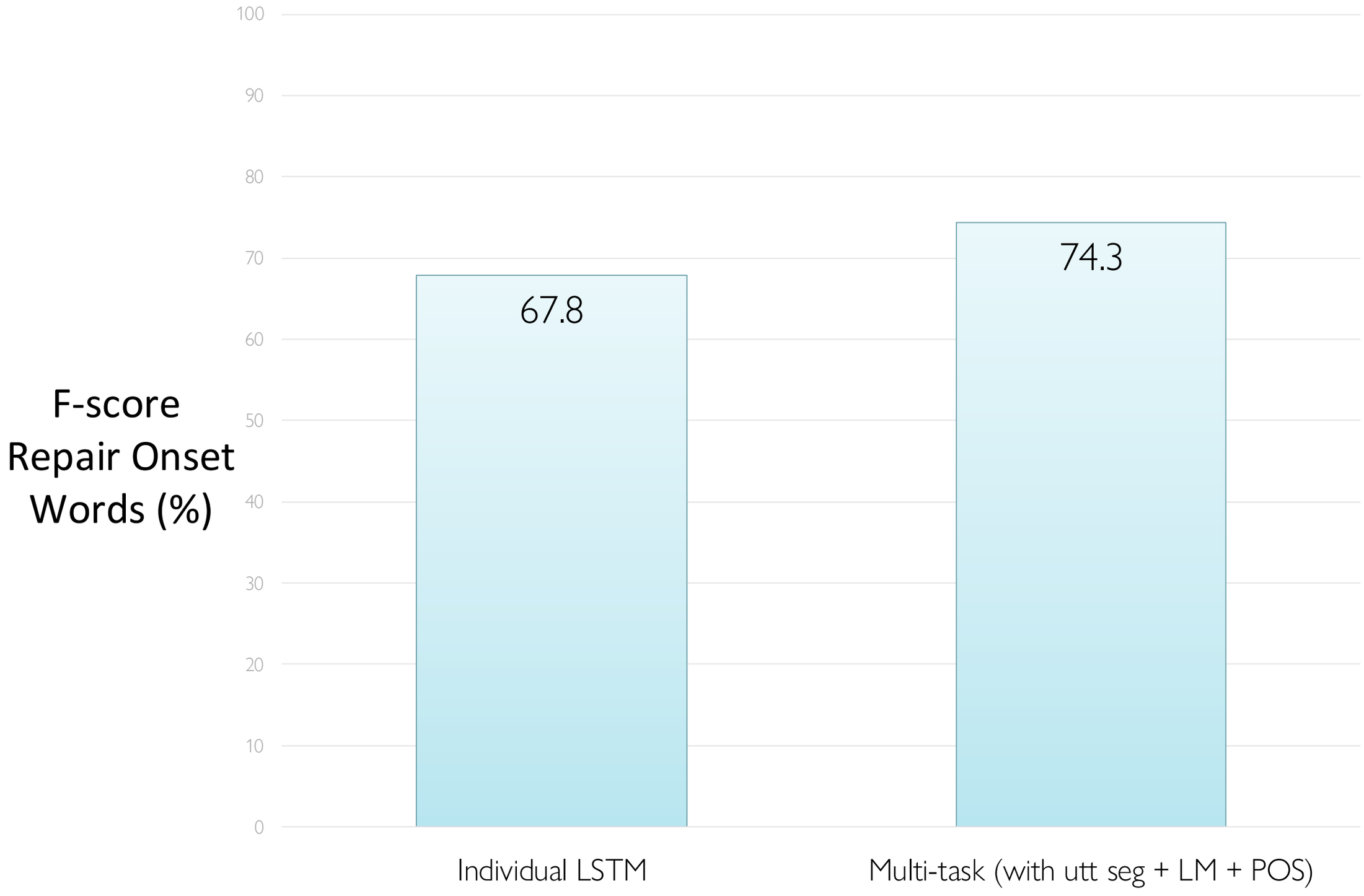
$$\widetilde{E} = \sum_{i=1}^{4} \frac{E_i}{\sigma_i^2} + \sum_{i=1}^{4} log(\sigma_i)$$

- Where E_i is one of four losses indexed to the individual tasks: E_{LM} , E_{Disf} , E_{Seg} and $E_{POS.}$ σ_i is the model's trainable observation noise parameter for each task *i*.

- We found uncertainty loss massively outperforms the 'naïve' method of summing the negative log loss.

MULTI-TASK LEARNING

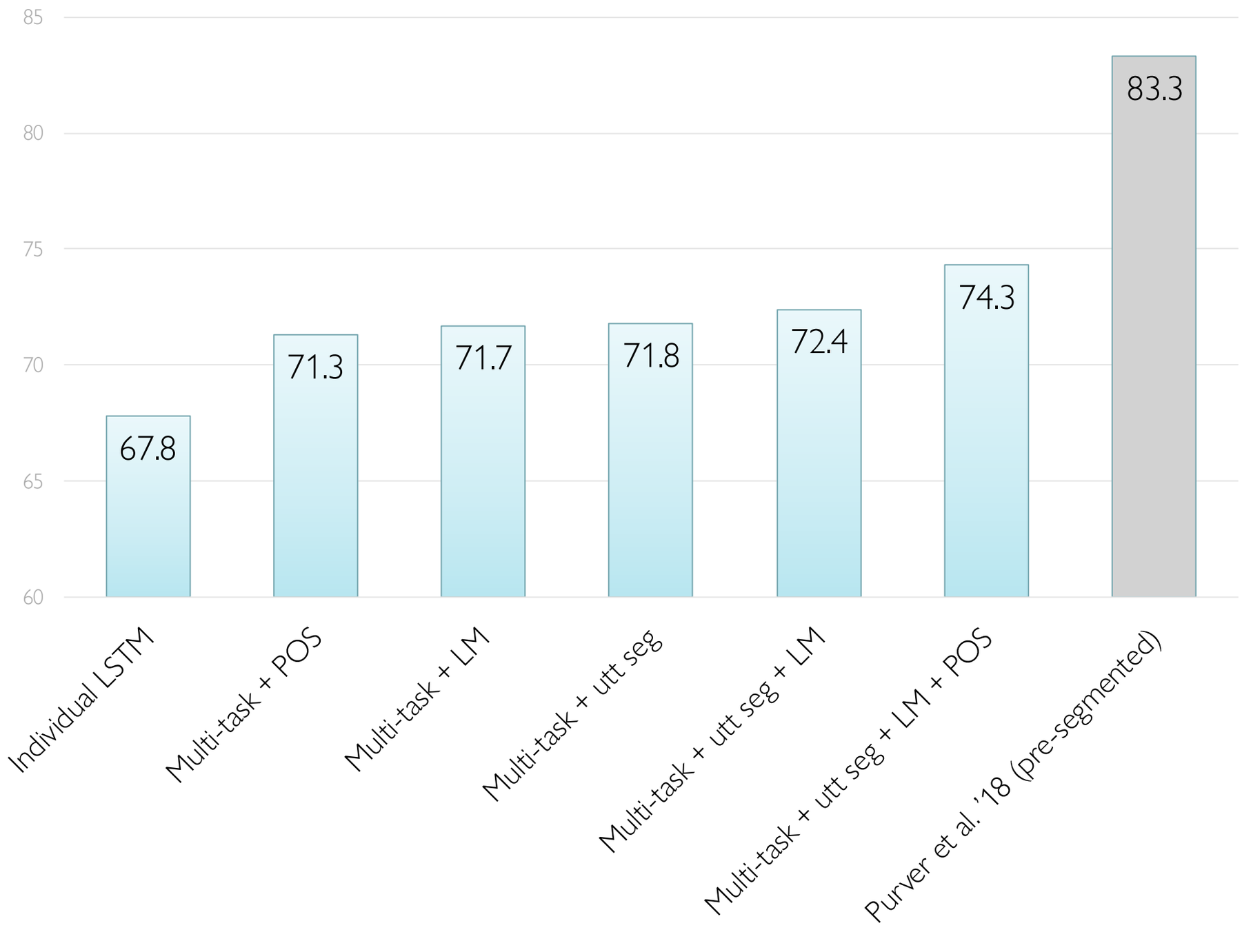
RESULTS: REPAIR DETECTION



MULTI-TASK LEARNING

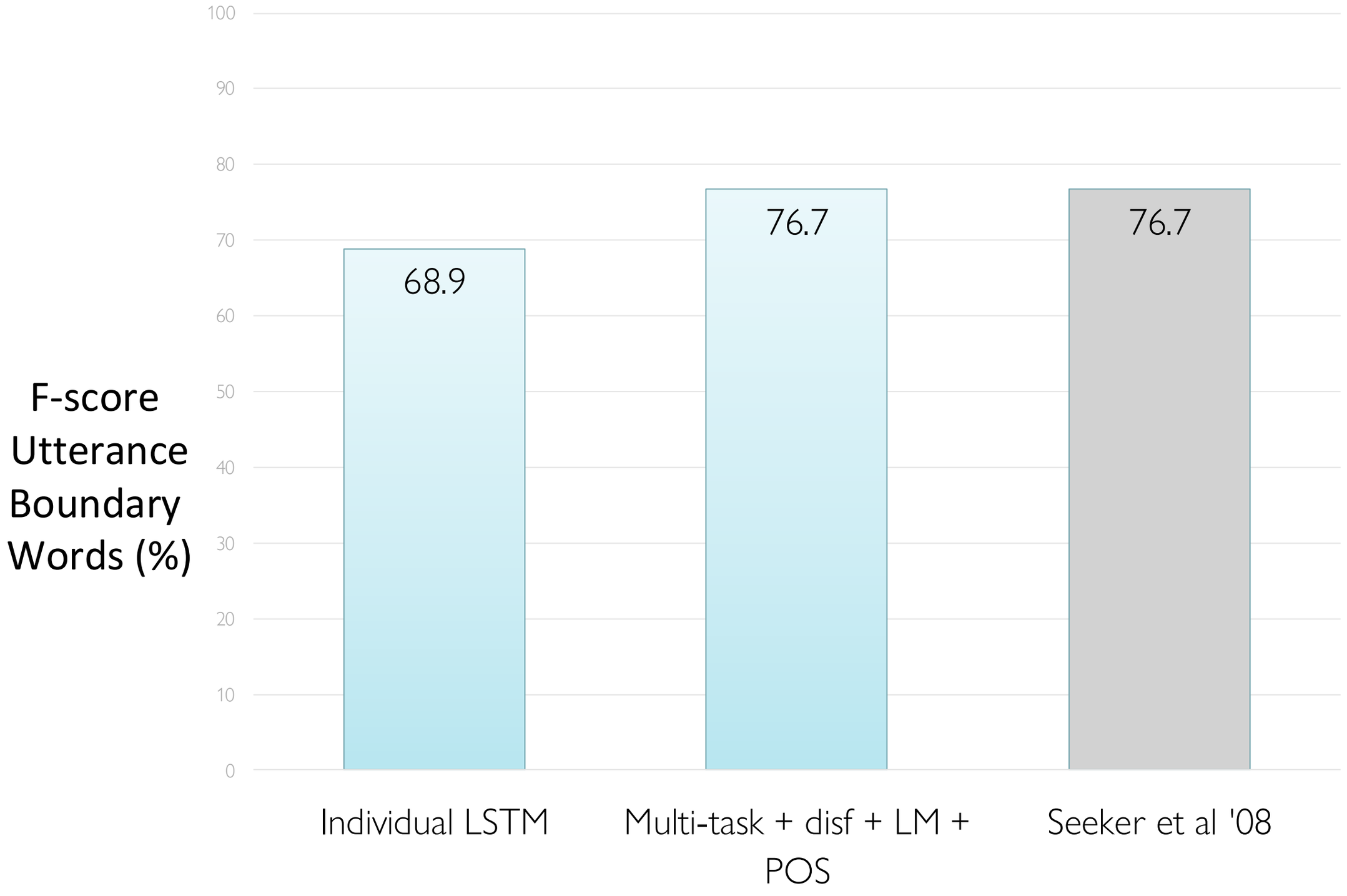
RESULTS: REPAIR DETECTION

F-score
Repair Onset
Words (%)



MULTI-TASK LEARNING

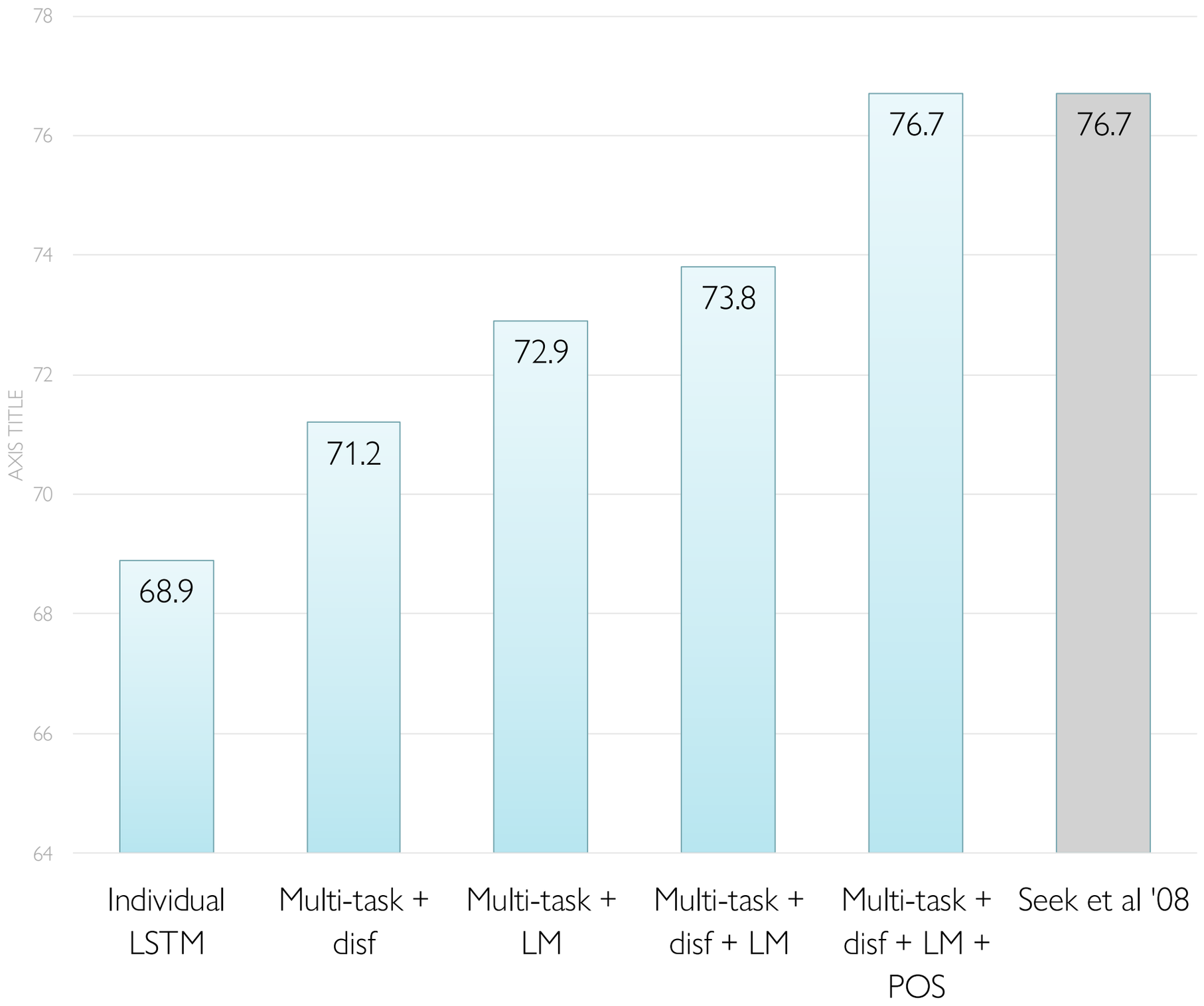
RESULTS: UTTERANCE SEGMENTATION



MULTI-TASK LEARNING

RESULTS: UTTERANCE SEGMENTATION

F-score Utterance Boundary Words (%)



BEST OF BOTH? BERT WITH GPT-2

- MTL certainly improves things over individual disfluency detection, but sequence models suffer from the **vanishing gradient problem** for longer repairs.

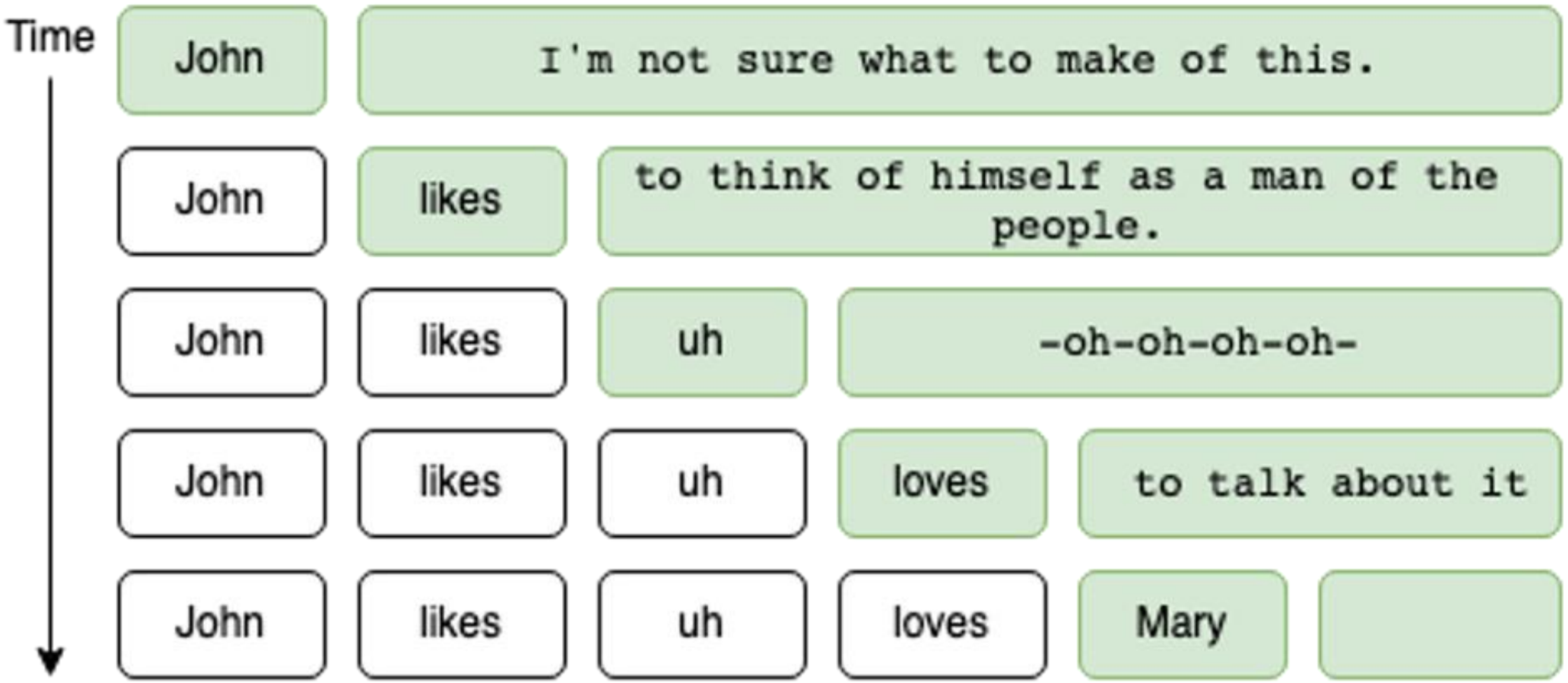
- Why don't we use the advantages of large **sentence-based transformer based models** with pre-training (e.g. BERT, GPT-2/3)?

- Rohanian and Hough (2021 ACL-ICNLP) **incrementalize** the non-incremental/utterance-global transformers.

- **We predict rightwards context of the current word** so as to use a BERT-based sequence classifier on that predicted full utterance, trying different **prophecy** methods.

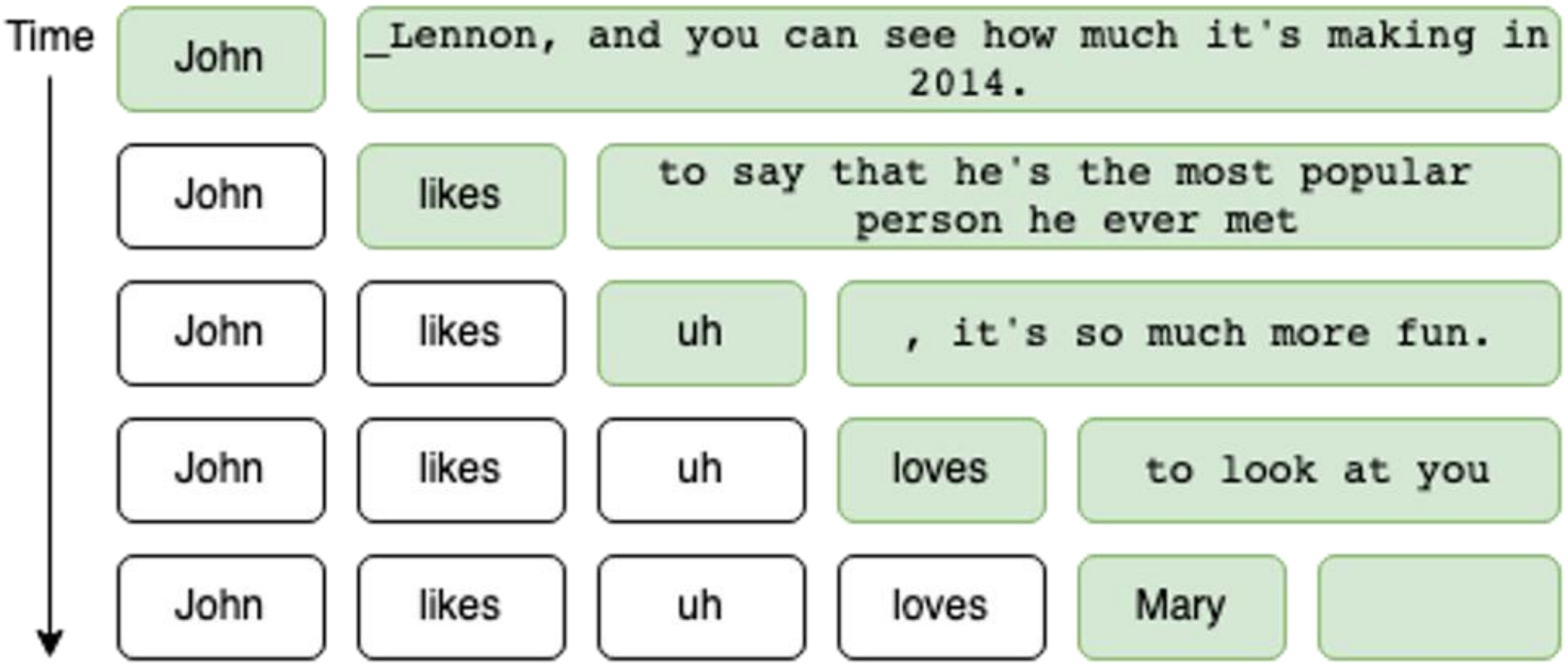
BEST OF BOTH? BERT WITH GPT-2

Beam search prophecy of following words with GPT-2



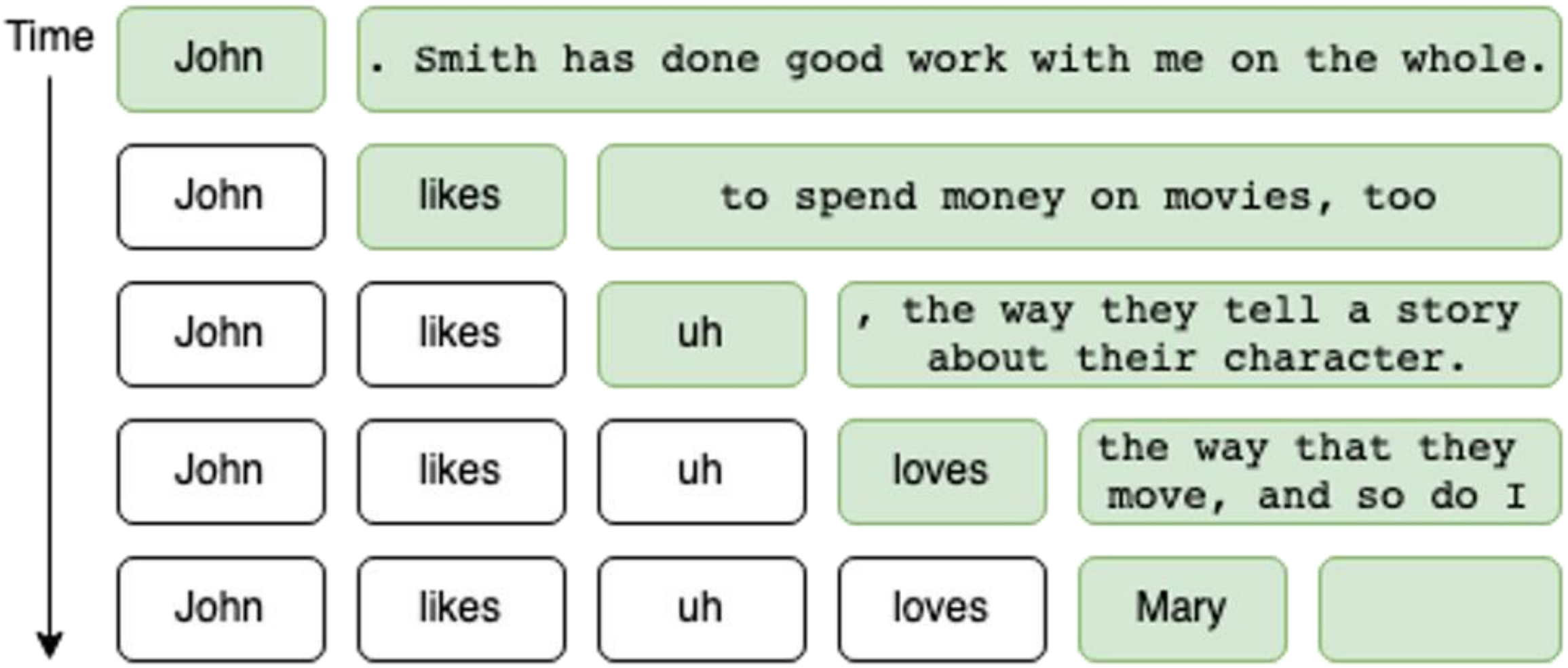
BEST OF BOTH? BERT WITH GPT-2

Top-*k* sampling prophecy of following words with GPT-2



BEST OF BOTH? BERT WITH GPT-2

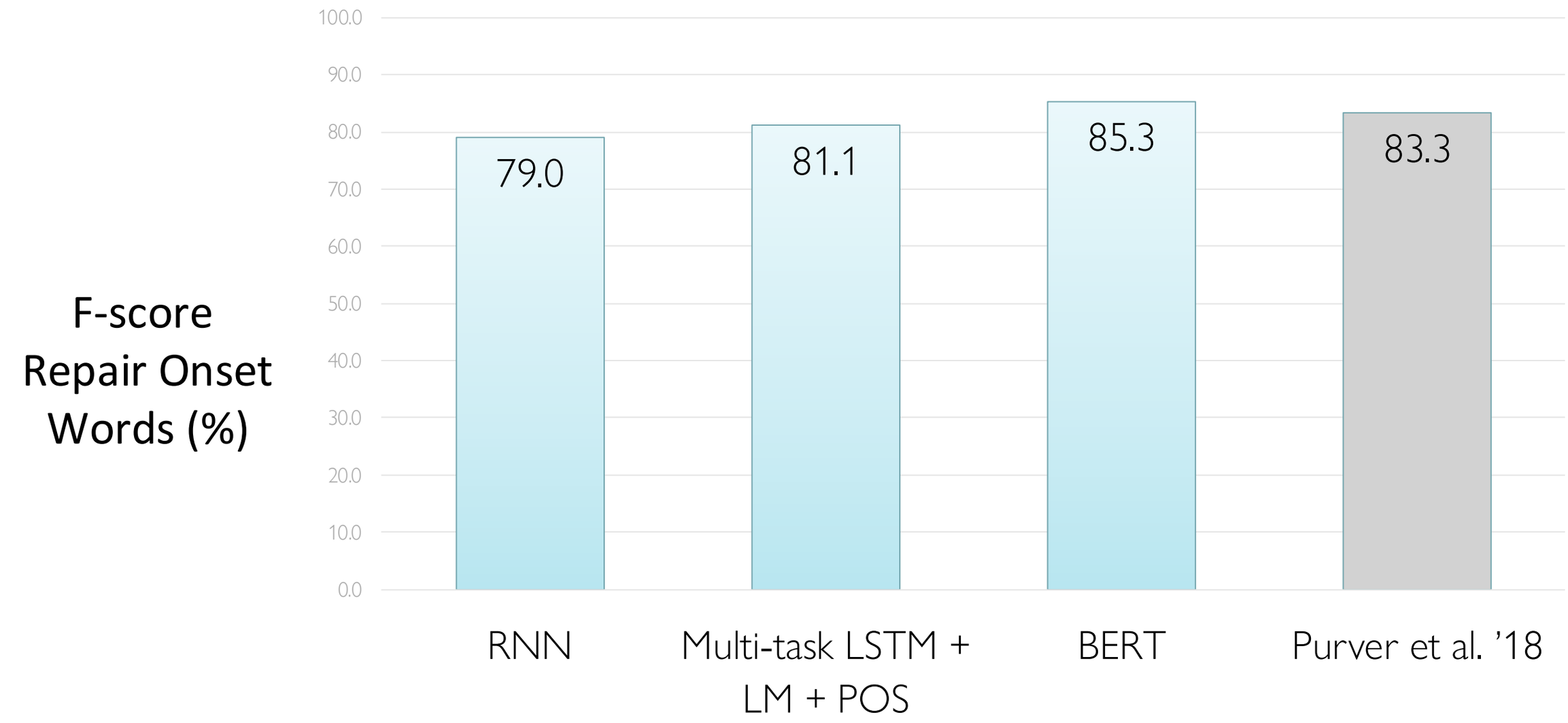
Top-*p* sampling prophecy of following words with GPT-2



- Send predictions to a pre-trained **BERT** architecture (Devlin et al., 2019) with a **Conditional Random Field (CRF)** output to tag sequences. **SOTA detection results!**

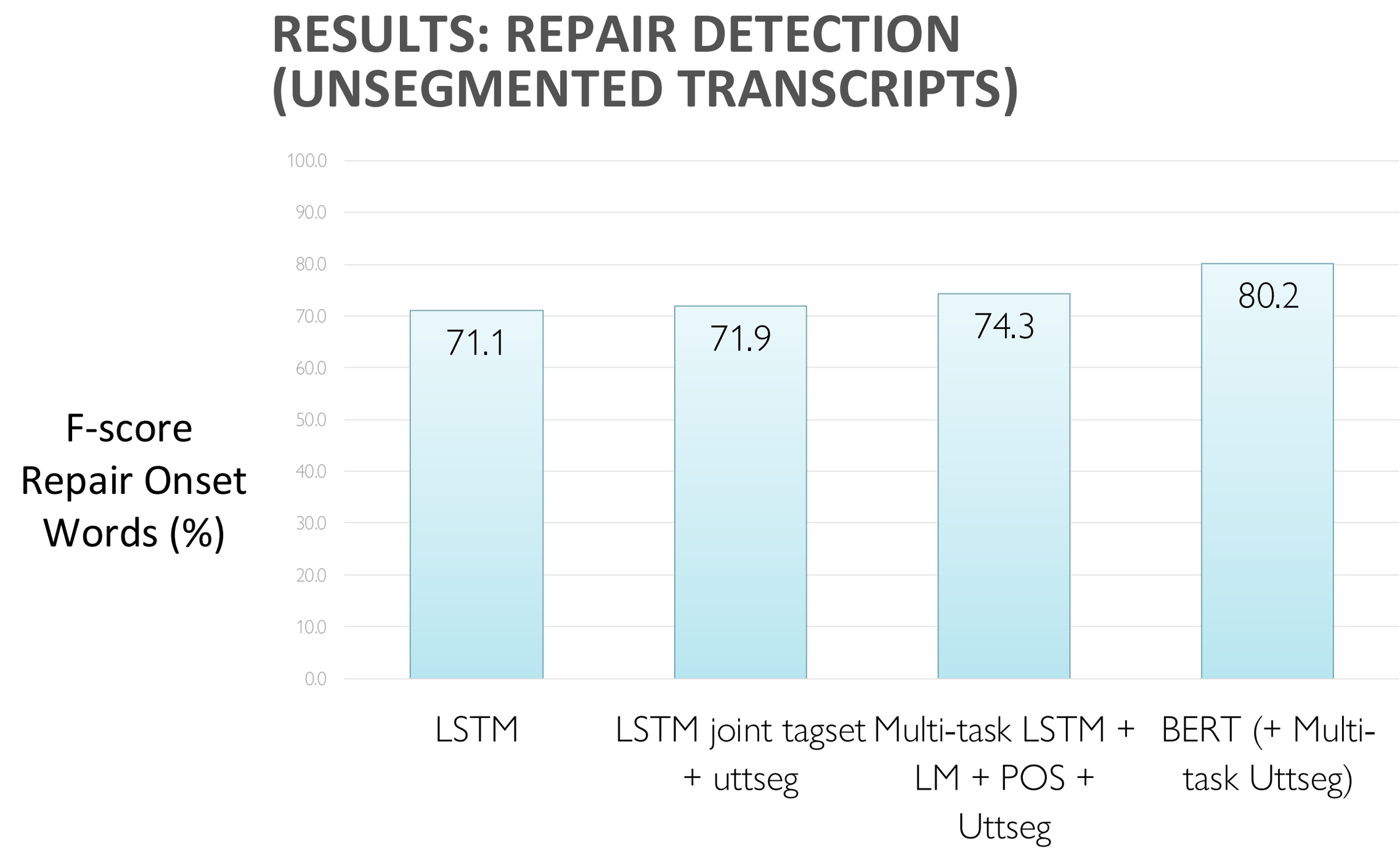
BEST OF BOTH? BERT WITH GPT-2

RESULTS: REPAIR DETECTION (PRE-SEGMENTED UTTERANCES)



- BERT is SOTA!

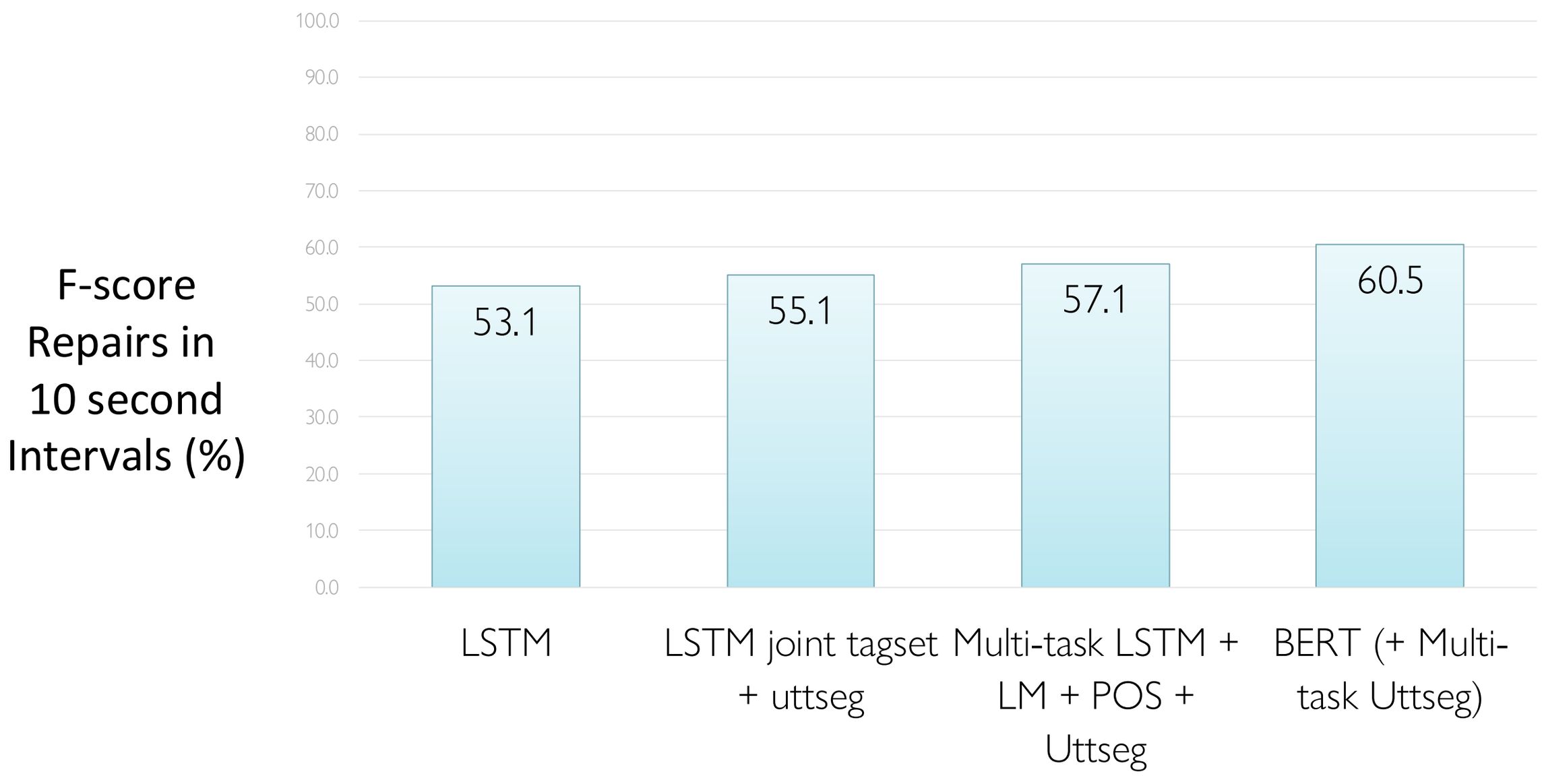
BEST OF BOTH? BERT WITH GPT-2



- BERT is SOTA!

BEST OF BOTH? BERT WITH GPT-2

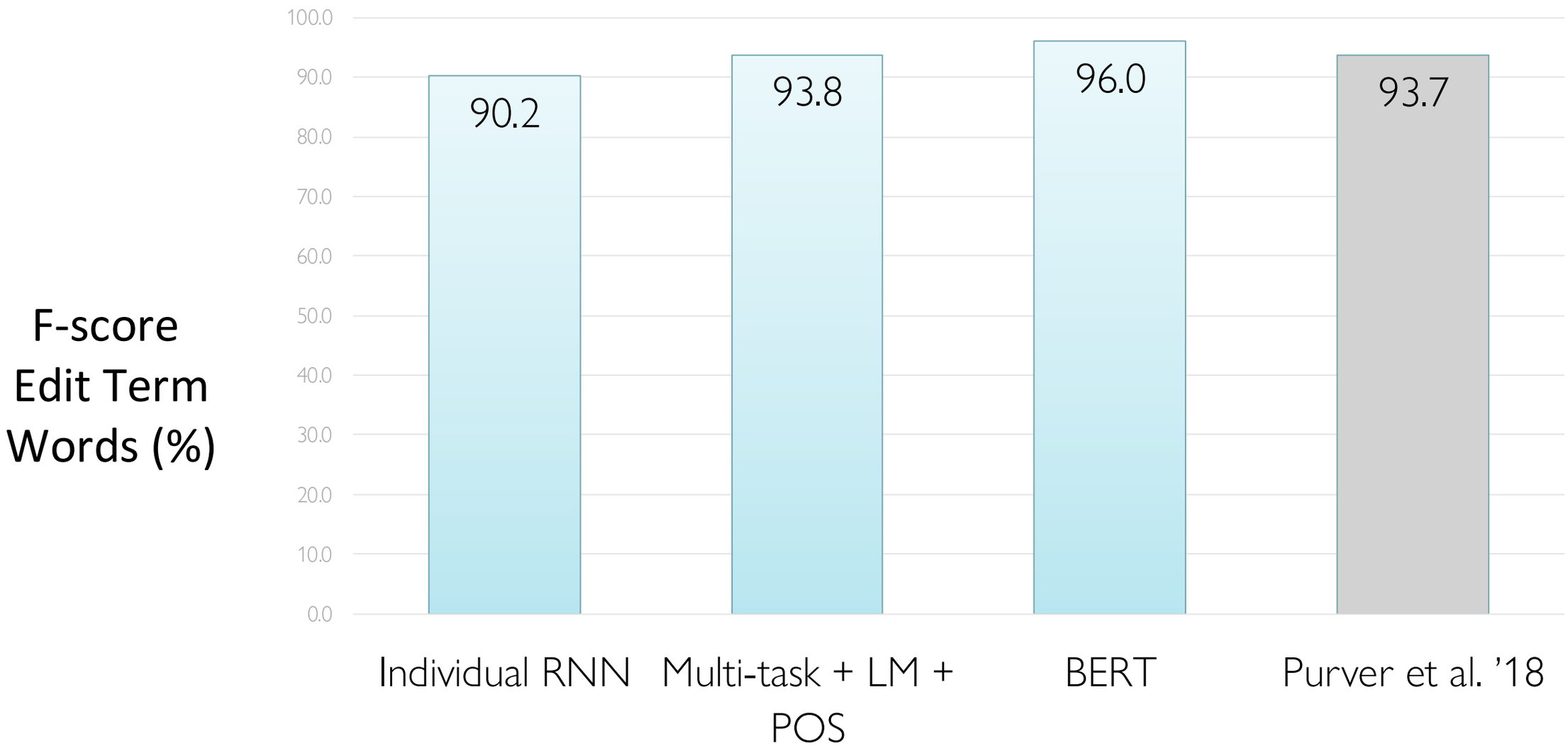
RESULTS: REPAIR DETECTION (ASR RESULTS)



- BERT is SOTA!

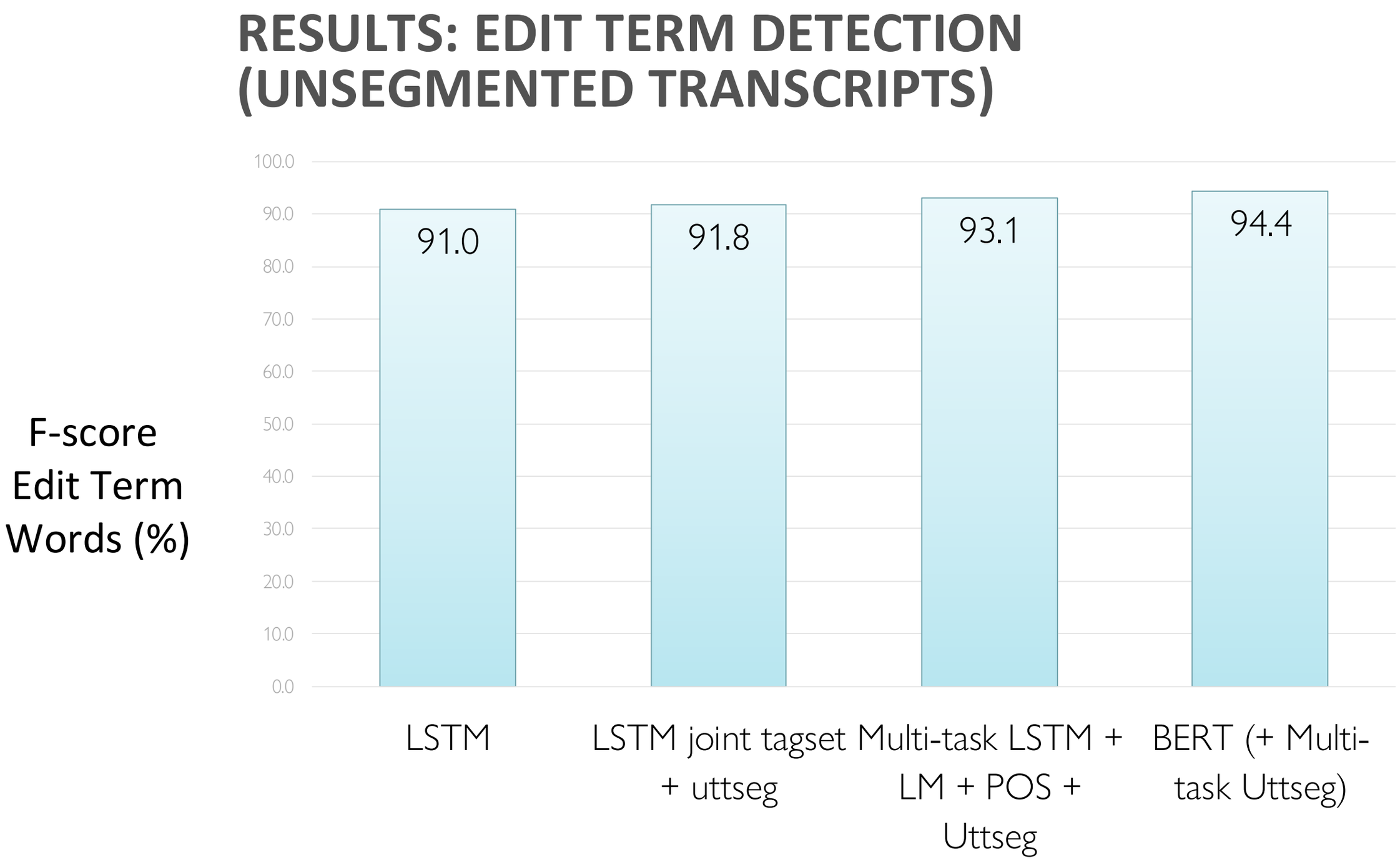
BEST OF BOTH? BERT WITH GPT-2

RESULTS: EDIT TERM DETECTION (PRE-SEGMENTED UTTERANCES)



- BERT is SOTA!

BEST OF BOTH? BERT WITH GPT-2

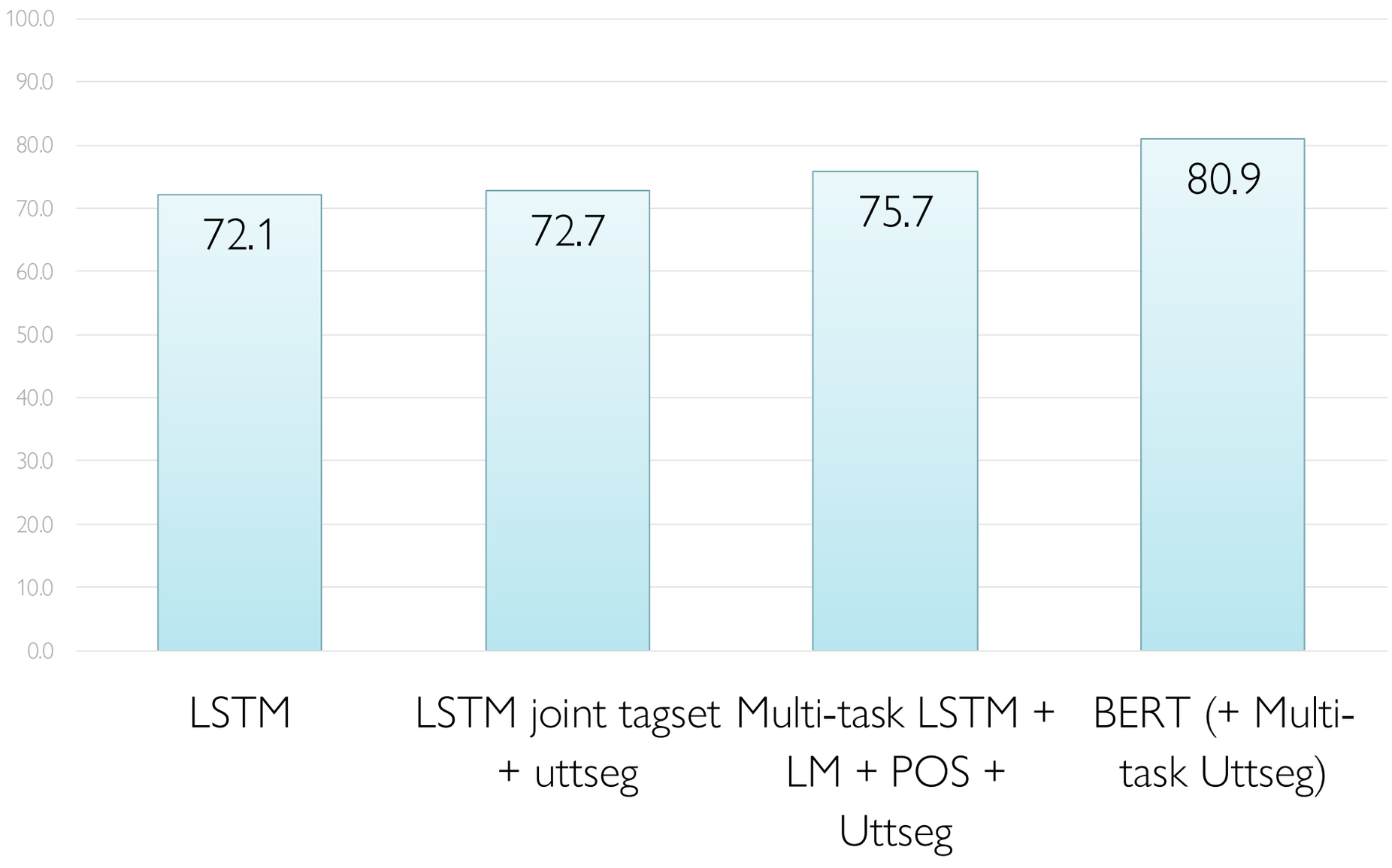


- BERT is SOTA!

BEST OF BOTH? BERT WITH GPT-2

RESULTS: EDIT TERM DETECTION (ASR RESULTS)

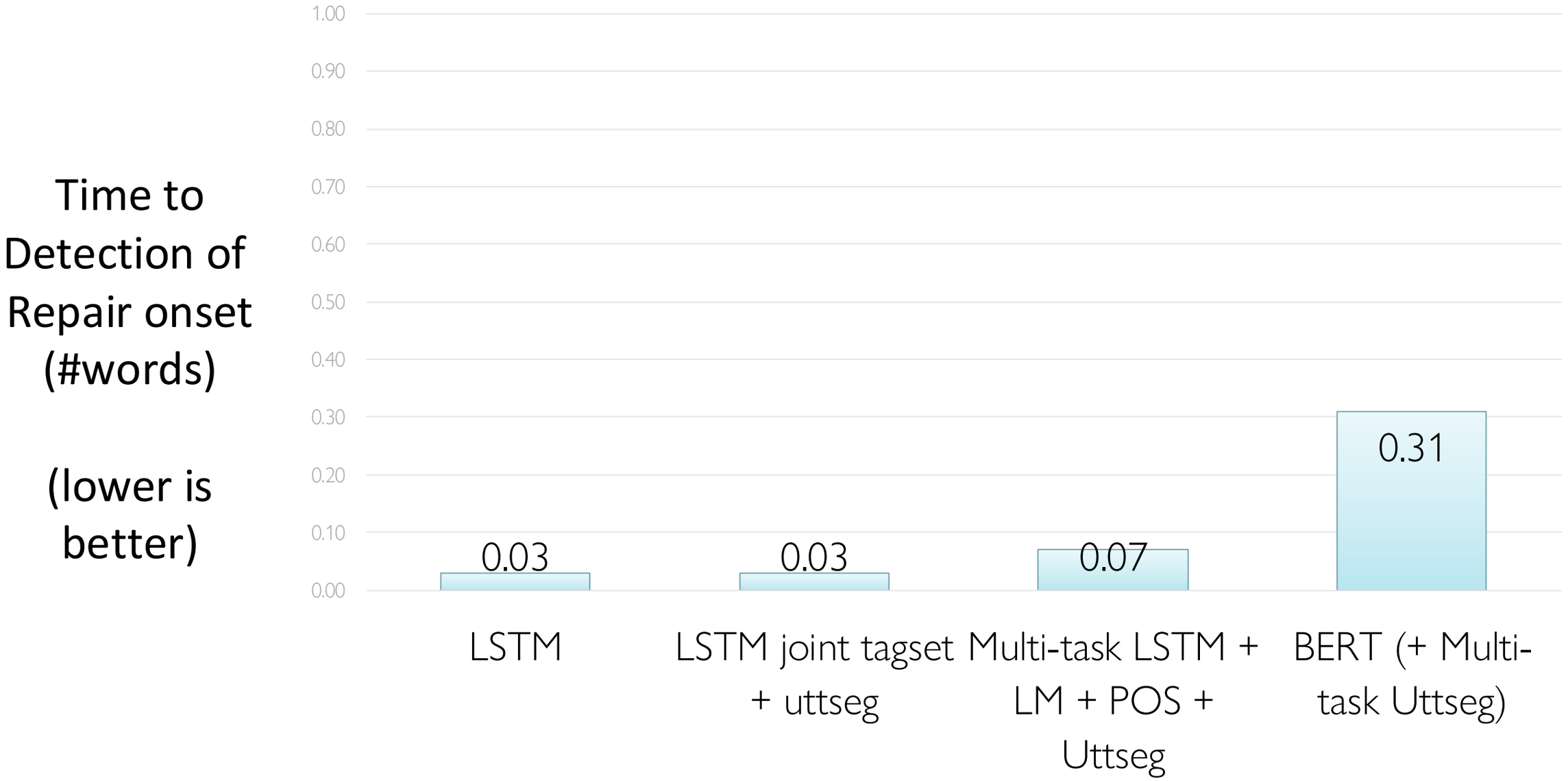
F-score Edit Terms in 10 second Intervals (%)



- BERT is SOTA!

BEST OF BOTH? BERT WITH GPT-2

RESULTS: INCREMENTAL TIME-TO-DETECTION (UNSEGMENTED TRANSCRIPTS)



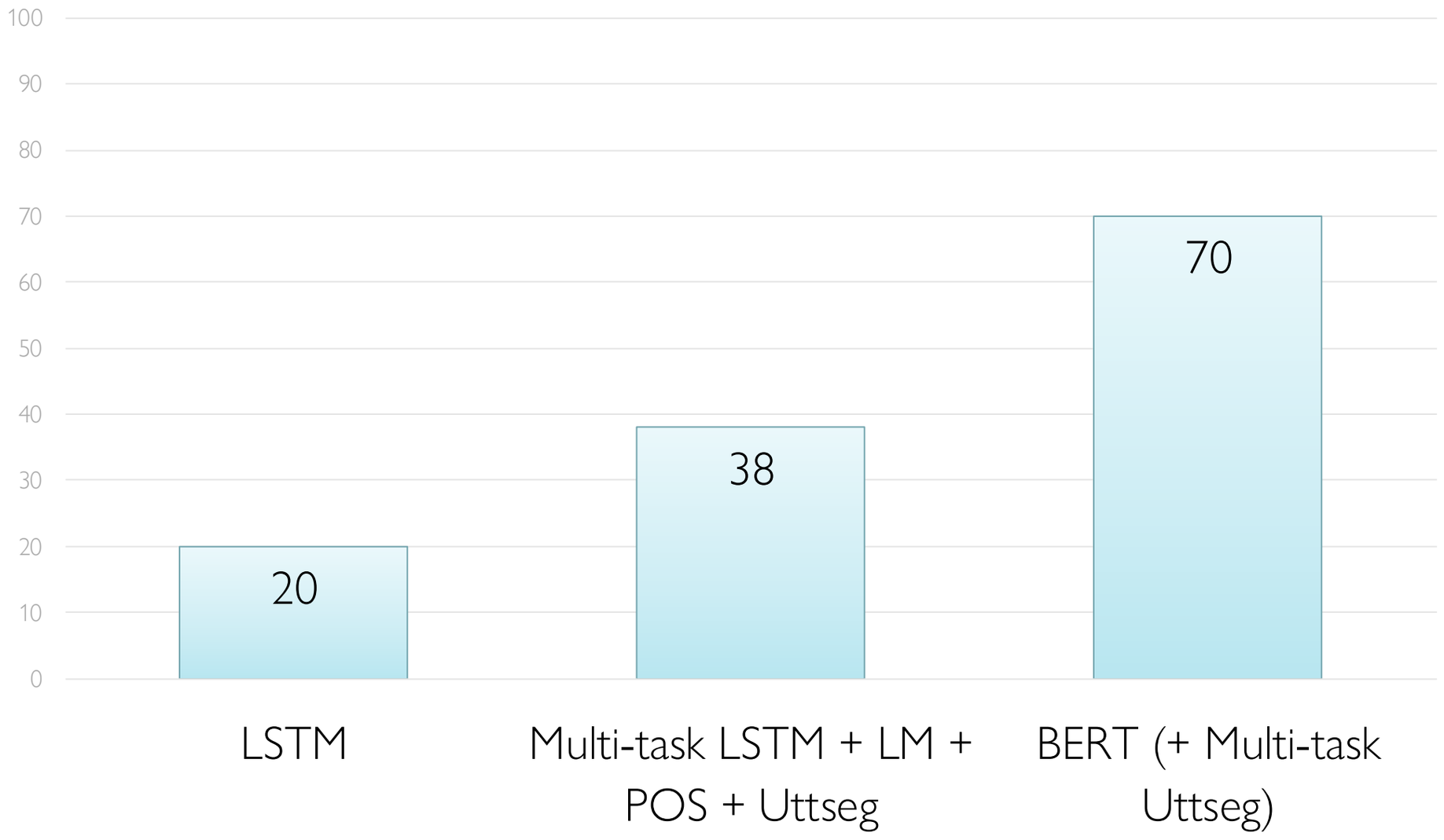
- BERT slightly slower to detect, but still quite fast.

BEST OF BOTH? BERT WITH GPT-2

RESULTS: INCREMENTAL EDIT OVERHEAD (UNSEGMENTED TRANSCRIPTS)

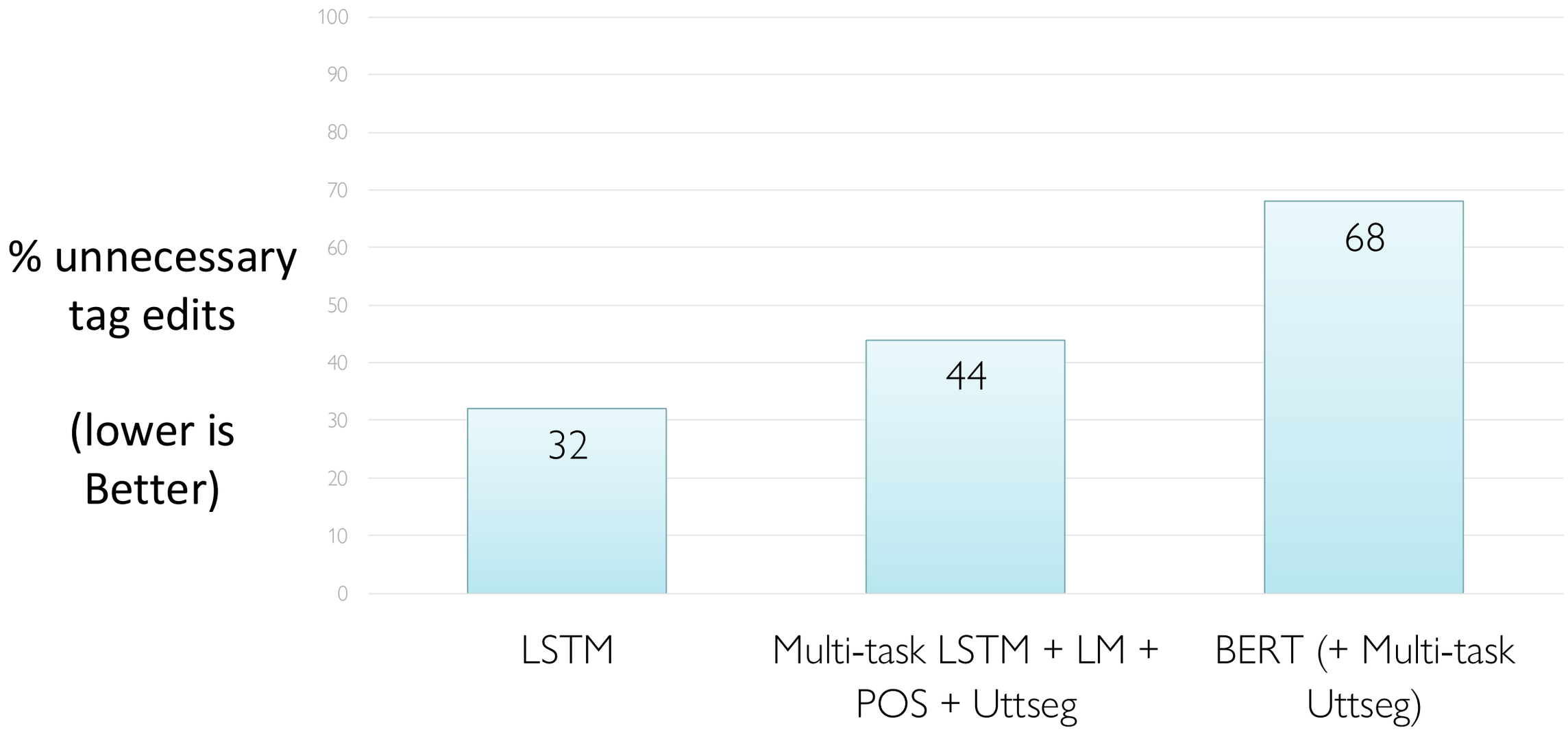
% unnecessary tag edits

(lower is Better)



BEST OF BOTH? BERT WITH GPT-2

RESULTS: INCREMENTAL EDIT OVERHEAD (ASR RESULTS)

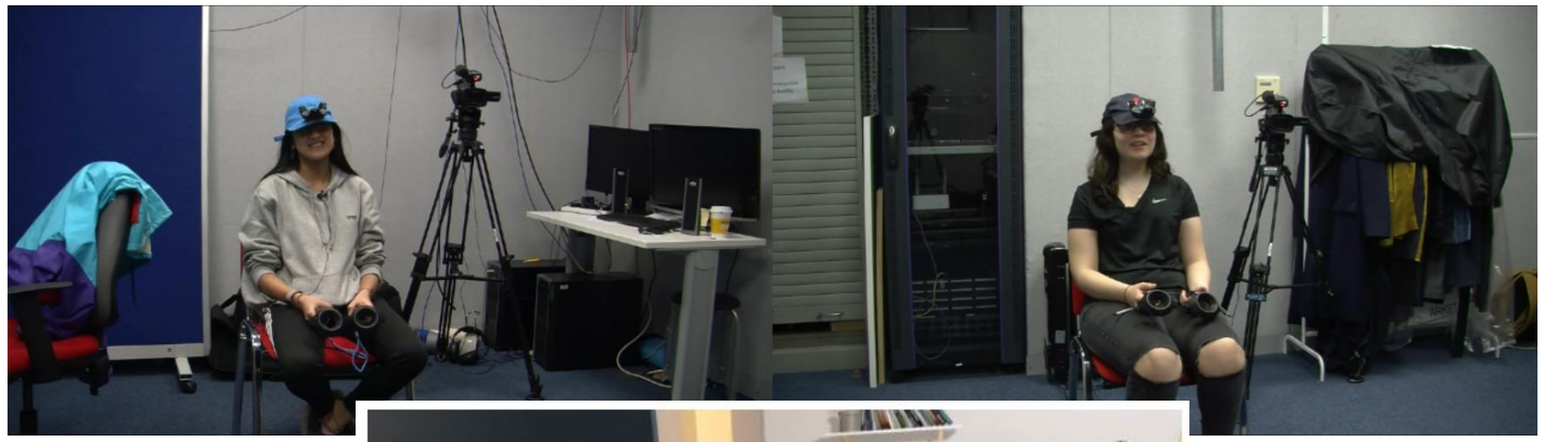


- BERT slightly less stable than MTL, but does not get worse on ASR results.

CONTENTS

APPLICATIONS

EACL 2017 model used to testing theories of face-to-face communication:



- Testing models of **head movement** relationship to disfluency in real time in VR interactions and face-to-face conversation (Gurion et al. 2018 & 2020 SemDial).
- Testing models of **hand movement** in relationship to disfluency (Özkan et al 2021 ICDL).

APPLICATION: ALZHEIMER'S DETECTION

Nasreen et al (2021 Frontiers in CS) use the EACL 2017 tagger on 30 patient-doctor conversations (15 AD patients, 15 non-AD patients). Findings:

- AD patients do significantly **more edit terms, repetitions and substitutions** than non-AD controls (cognitive effect).

- Doctors in conversation with AD patients significantly **more edit terms and repetitions** compared to when talking to non-AD controls (interactive effect).

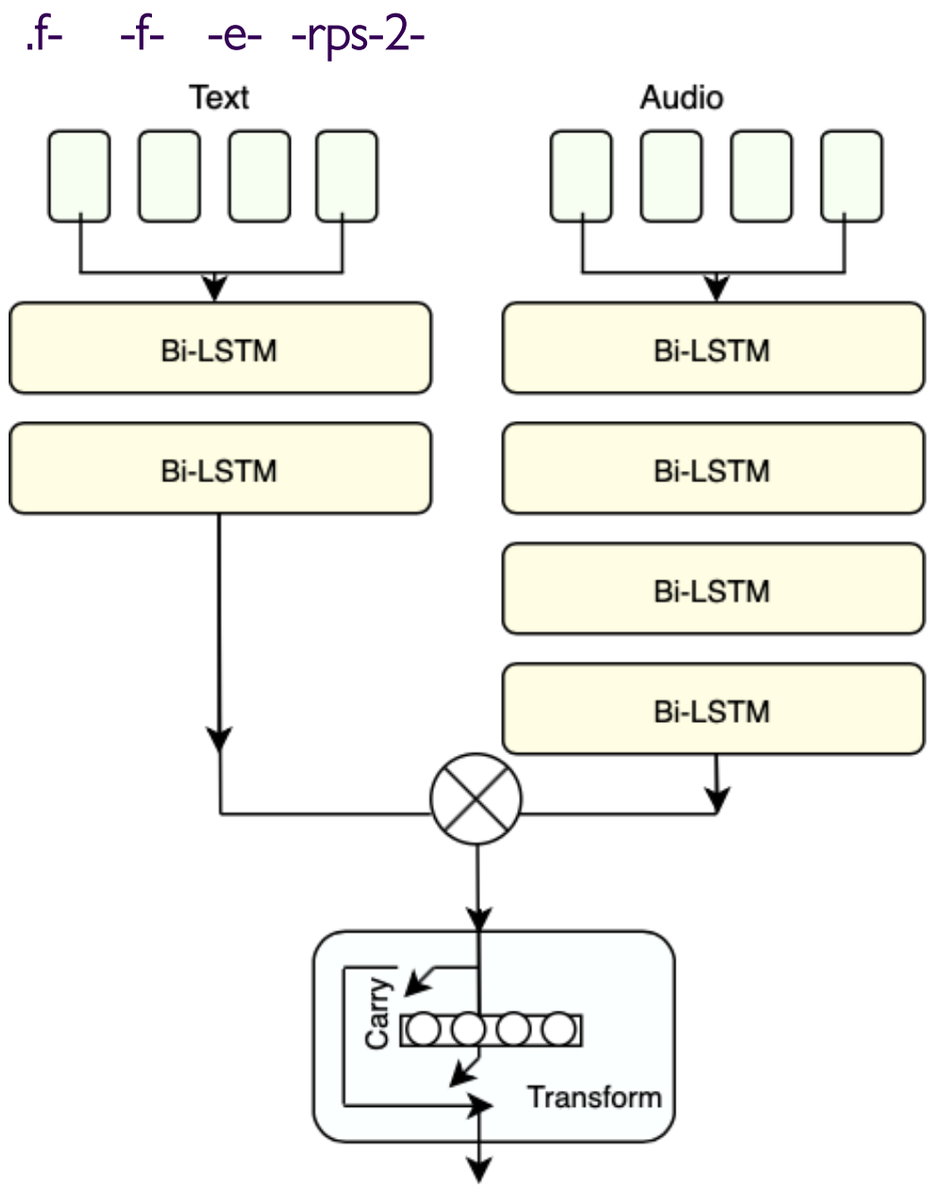
Disfluency features alone used in an SVM classifier can predict AD with **83%** accuracy, combined with other features **90%.**

APPLICATION: ALZHEIMER'S DETECTION

- Rohanian et al (2020-21) use **multi-modal fusion** between lexical and acoustic feature sequences with gating to account for noise.

- Using our disfluency tag features on speech, combined with other features:

- **79% acc. AD detection** on transcripts using EACL 2017 model (2020).
- **84% acc. AD detection** on speech with IBM Watson ASR results and COLING 2020 model (2021).



CONCLUSION

- We need incremental disfluency detection in our dialogue systems and speech processing systems.

- DNN tagging methods with no feature engineering can work well for disfluency detection: only words, POS tags and word timings for input.

- Treating disfluency detection, utterance segmentation, POS tagging and language modelling as joint task works well, showing inter-dependence of tasks.

- Good performance can be achieved in overall repair rate correlation on ASR results, can be used in a variety of live applications and helps in the clinical application of AD detection.

PYTHON IMPLEMENTATION

deep_disfluency

https://github.com/clp-research/deep_disfluency

- Abides by FAIR principles. Hough and Schlangen (2017 EACL) results largely completely reproducible independently (Grimm et al, 2021):
  - Repair onsets F-score 69.2%(-2.8% published result)
  - Edit term words F-score 90.9% (-0.9% published result).
  - Utt segmentation F-score 68.3%/72.0% (-6.5%/-2.8% published results)
- Python 3 release with Tensor Flow coming soon with additional models.

REFERENCES

de Mulder, W., Bethard, S., & Moens, M. F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, *30*(1), 61-98.

Gurion, T, Healey, P and Hough, J. (2020). Comparing Models of Speakers' and Listeners' Head Nods. In Proceedings of *SemDial 2020*

Hough, J. and Purver, M. (2014). Strongly Incremental Repair Detection. *EMNLP 2014.*

Hough, J. and Schlangen, D. (2015). Recurrent Neural Networks for Incremental Disfluency Detection. Interspeech, 2015.

Hough, J and Schlangen, D. (2017). Joint, Incremental Disfluency Detection and Utterance Segmentation from Speech. EACL 2017.

Mesnil, G, Xiaodong He, Li Deng and Yoshua Bengio. (2013). Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding. Interspeech, 2013.

Nasreen, S, Rohanian, M, Hough, J and Purver, M. (2021) Alzheimer's Dementia Recognition from Spontaneous Speech using Disfluency and Interactional Features . In *Frontiers in Computer Science*

Rohanian, M and Hough, J. (2020). Re-framing Incremental Deep Language Models for Dialogue Processing with Multi-task Learning. COLING 2020.

Rohanian, M and Hough, J. (2021). Best of Both Worlds: Making High Accuracy Non-incremental Transformer-based Disfluency Detection Incremental. ACL-IJCNLP 2021

Rohanian, M, Hough, J. and Purver, M.((2020). Multi-modal fusion with gating using audio, lexical and disfluency features for Alzheimer’s Dementia recognition from spontaneous speech. INTERSPEECH 2020

Rohanian, M, Hough, J. and Purver, M.(2021). Alzheimer’s Dementia Recognition Using Acoustic, Lexical, Disfluency and Speech Pause Features Robust to Noisy Inputs. INTERSPEECH 2021

Shriberg, E. E. (1994). Preliminaries to a theory of speech disfluencies (Doctoral dissertation, University of California at Berkeley).

Zwarts, S, Johnson, M. and Dale, R. (2010). Detecting speech repairs incrementally using a noisy channel approach. In Proceedings of the 23rd International Conference on Computational Linguistics. COLING ’10, Stroudsburg, PA,USA. ACL

APPENDIX

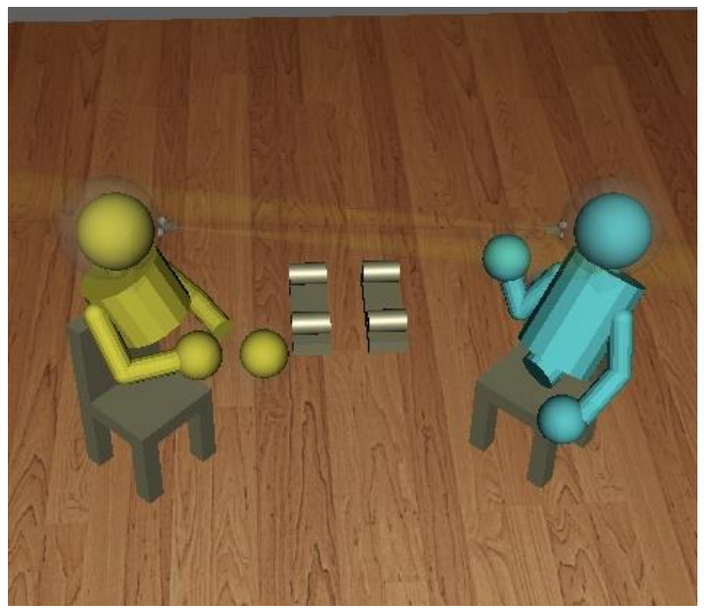
DUEL: A Unique Dataset For Disfluency, Laughter and Exclamations

Julian Hough, Ye Tian, Chiara Mazzocconi,
David Schlangen and Jonathan Ginzburg
(and the rest of the DUEL team)

The DUEL Corpus

Video and audio data
(MP4 + wav)

Experimental Metadata
Task descriptions of dialogue tasks:
Dream Apartment
Film Script
Border Control



Kinect 2 skeleton data
(time-stamped XML)

Transcription and Annotation
Inline disfluency and laughter annotation,
Separate laughter tiers
(TextGrid and csv)

DUEL
24 hours of French, German and Chinese dialogue

searchable with

Mumodo.py
(Kousidis et al., 2013)

The DUEL Corpus

DUEL annotation: simple, consistent mark-up

Repairs and restarts (Shriberg, 1994)

(ich + ich) will
(ich + (ich + ich)) will
(nicht verwinkelt so dass **+)** und breit

Partial words and variants (Deshmukh et al., 1999)

<p="Wohnzimmer">Wo-</p>
<v="einen">'n</v>

Abandoned utterances

Ja eigentlich <v="wäre es"> wär's </v> cool in der Küche <v="einen">'n</v> kleinen Tisch zu haben wo man **-**
Yes actually it would be cool to have a small table in the kitchen where you -

Filled pauses, fillers, unfilled pauses, Exclamations

das **{F ähm }** Wohnzimmer
(das + { also } das) Wohnzimmer
das . Wohnzimmer
das .. Wohnzimmer
das ... Wohnzimmer

Laughter & laughed speech

(Und mit einem +) mit vielleicht Sachen die nicht **<laughter>** auseinander brechen **</laughter> <laughterOffset/>**
(And with a +) with perhaps things that do not <laughter> fall apart </laughter> <laughterOffset/>

Category	**Agreement**	*K*-free
Reparandum	0.9477	0.8954
Repair	0.9677	0.9353
Filled pause	0.9968	0.9937
Laughed speech	0.9558	0.9117