PRINCETON UNIVERSITY

Adversarial Examples in NLP

Elisabetta Cavallo, Seyoon Ragavan

COS598C - Deep Learning for Natural Language Processing

April 16th, 2020

Agenda

Outline

- **What are Adversarial Examples?**
- **Generating Adversarial Examples**
    - **Paper I:** Adversarial Example Generation with Syntactically Controlled Paraphrase Networks (Iyyer at al., 2018)
- **Training with Adversarial Examples**
    - **Paper II:** Adversarial Examples for Evaluating Reading Comprehension Systems (Jia et Liang, 2017)
- **Quiz Time & Bonus Papers!**

What are Adversarial Examples?

Introduction

- Applicable to **any NLP task and model**.
- An adversary or attack slightly perturbs the input to **fool the model**.

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Important: the perturbation doesn't change what the correct answer would be

(Jia and Liang, 2017)

What are Adversarial Examples?

Terminology

- **Adversarial example**: a perturbed input-output pair
- **Adversary/attack:** a method for generating examples
- **Robustness**: how well a model performs against an adversary.
  - Model evaluated with the same metric as in the standard/non-adversarial

Why do Adversarial Examples matter?

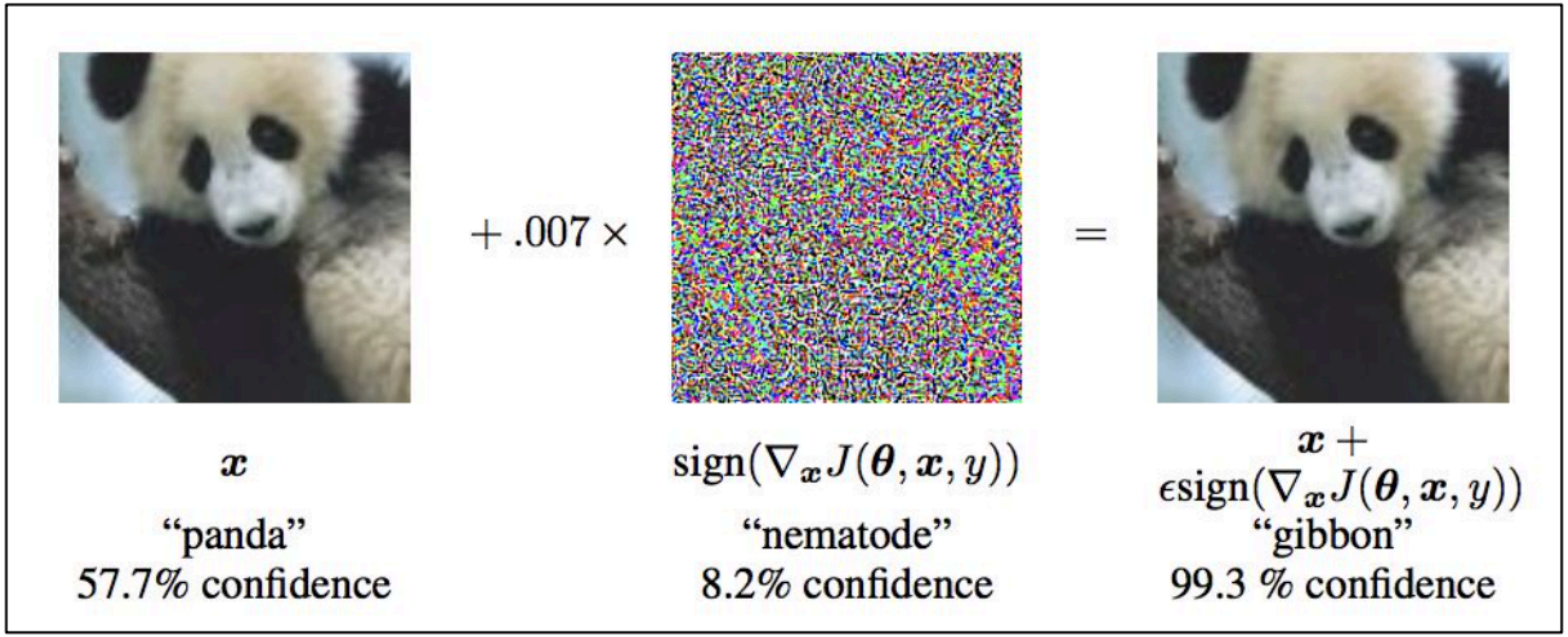
- **Security** is important for some applications
  - Spam detection
  - Healthcare
- **Evaluation** of models and datasets
  - Does the model/dataset really exhibit/test sophisticated understanding?
- **Interpretation** of models
  - What does the model care about, and what does it ignore?
  - Are these bugs that need to be addressed?
- **Robust training** of models
  - Augment training data with adversarial examples

(Singh, 2019)

What are Adversarial Examples?

Adversarial Examples in Computer Vision

- Image classification task
- Gradient-based attacks to increase loss



(Goodfellow et al., 2015)

What are Adversarial Examples?

Computer Vision vs. Natural Language Processing

	Images	Text	
Input type	Continuous	Discrete	
Original Input		*"Quarterback John Elway was 38 in Super Bowl XXXIII."*	*"I'd have to say the star and director are the big problems here."*
Adversarial Input		*"Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV."*	*"By the way, you know, the star and director are the big problems."*
Semantics	Same	Different	Same
Model's mistake	Treats the two as different	Treats the two as the same	Treats the two as different
Exploited weakness	Oversensitivity	Overstability	Oversensitivity

Generating Adversarial Examples

What can we modify in the original sentence to create an adversarial example?

Character-level: flip / insert / delete a character.

Typoglycemic text:

"*Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae.*"

Word-level: change a word in a sequence.

x = ['I' $\boxed{\text{'like'}}$ 'this' 'movie' '.']

Let's replace this word

Random word?	x' = [	'I'	$\boxed{\text{'lamp'}}$	'this'	'movie'	'.'	]
Word Embedding?	x' = [	'I'	$\boxed{\text{'really'}}$	'this'	'movie'	'.'	]
Part of Speech?	x' = [	'I'	$\boxed{\text{'eat'}}$	'this'	'movie'	'.'	]

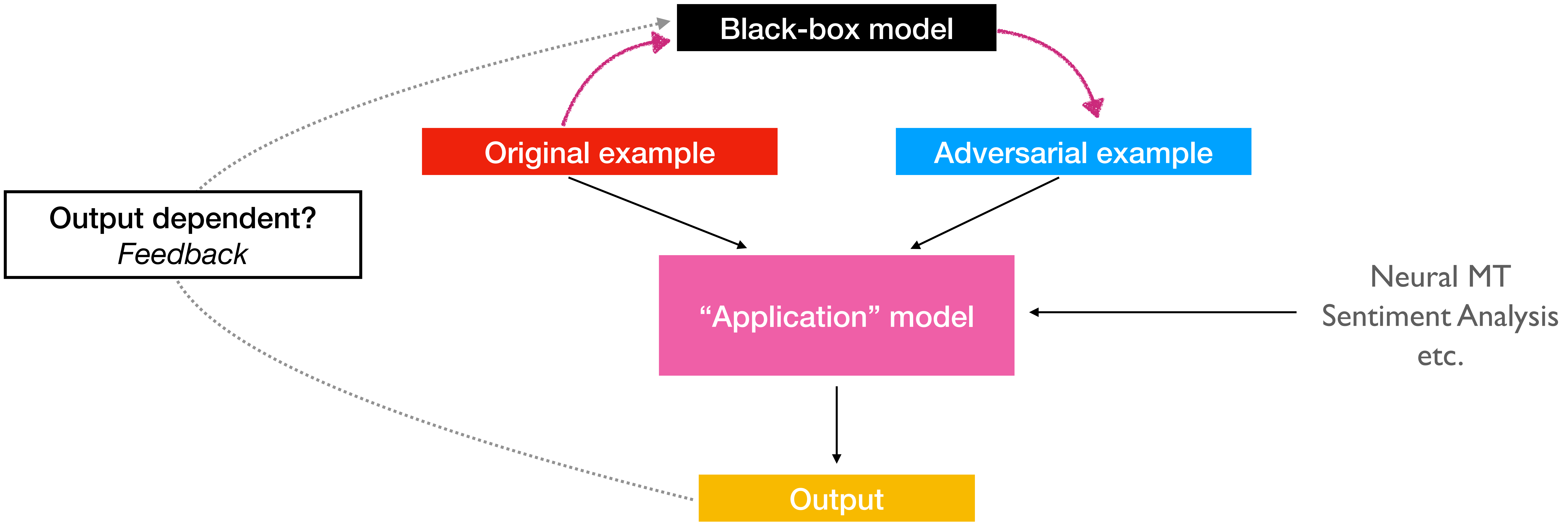
Sentence-level: replace the entire sentence.

"Susan told me she is pregnant" → "I was told by Susan she is expecting a baby"

Generating Adversarial Examples

How do we choose the adversarial attack to perform on our sentence?

- **Black-box:** close to random, relying on heuristic methods, disjoint from the "application" model → not "best" AE!

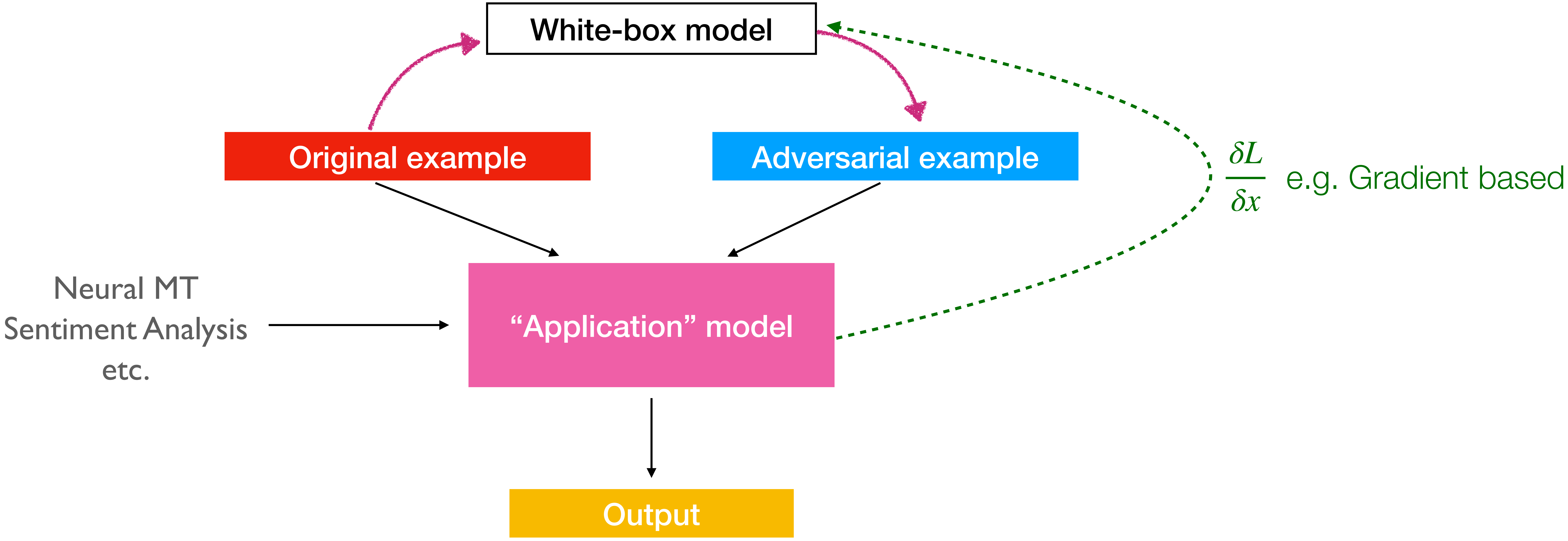


E.g.: **random shuffling of letters** (character-level), **word replacement based on POS tag** (word-level), **paraphrase with back-translation** (sentence-level).

Generating Adversarial Examples

How do we choose the adversarial attack to perform on our sentence?

- **White-box:** approximates the worst-case attack for a particular model and input, within some allowed set of perturbations.



E.g. Gradient based (discrete optimization): compute the gradient of the loss function relative to the input representation x, step in that direction and set the adversarial example x' equal to the nearest neighbour.

Generating Adversarial Examples

Discrete Optimization problem

$$\min_{x'} \|x - x'\|$$

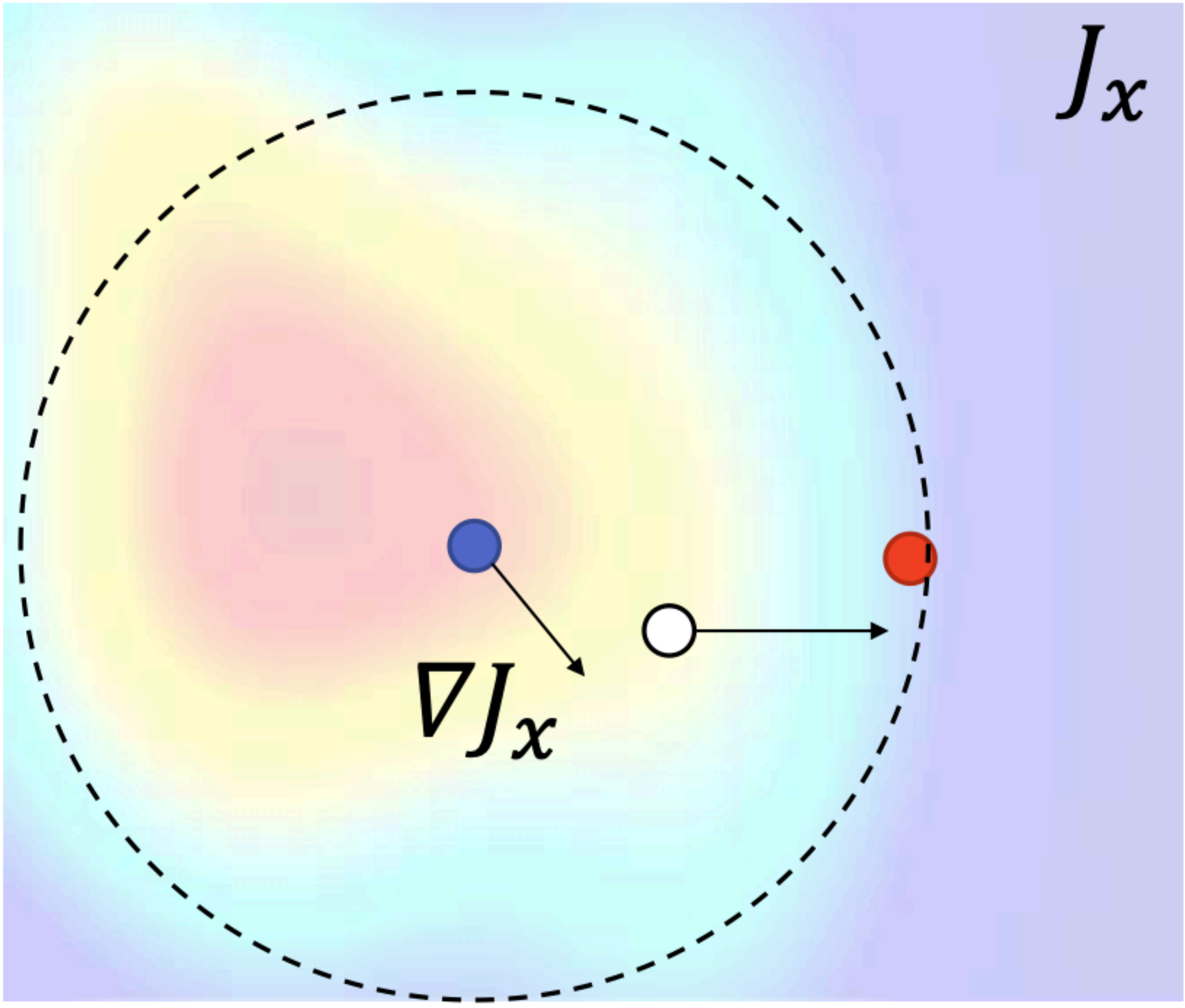
$$\text{s.t. } f(x') \neq f(x)$$

+

Model information

Loss function and gradient w.r.t input

Possible solution:
Approximate gradient method



Adversarial Example Generation with Syntactically Controlled Paraphrase Networks

Iyyer et Al. (2018)

Adversarial Example Generation with SCPN

Paraphrase generation: previous work

Black Box

Template-based: hand-crafted rules and grammars, thesaurus-based substitution, etc.

Translation-based: lattice-based SMT, statistical techniques, etc.

White Box

Gradient based: atomic flip operation (HotFlip by Ebrahimi et al, 2018), etc.

→ Lexical adversaries

"Exactly the kind of **unexpected delight** one hopes for every time the lights go down"

Positive

→

"Exactly the kind of **thrill** one hopes for every time the lights go down"

Negative

Adversarial Example Generation with SCPN

Paraphrase generation: SCPN

This paper: first learning approach for generate a *syntactically controlled* paraphrase of a given sentence.

→ Syntactical adversaries

"American drama doesn't get any more meaty and muscular than this"

Positive

+

Target **syntactic form** (e.g., a constituency parse)

SCPN

"Doesn't get any more meaty and muscular than this American drama"

Negative

Black box with output feedback

Adversarial Example Generation with SCPN

Overview

1. Training Data

Backtranslation from Wieting et al. (2017)

2. Sentence parsing

Get $\langle p_1, p_2 \rangle$ for each $\langle s_1, s_2 \rangle$

Template relaxation

Use template t_2 for p_2

3. Model

1) Parse generator

Produce complete parses from template Parses (from t_2 to p_2).

2) SCPN

Generate s_2 from s_1, p_2.

4. Evaluation

Intrinsic evaluation

Adversarial evaluation

Adversarial Example Generation with SCPN

1. Training data

No large-scale dataset of sentential paraphrases exists publicly.

→ use the pre-trained **PARANMT-50M corpus from Wieting and Gimpel (2017)**: 50 million paraphrases obtained by backtranslating the Czech side of the CzEng.

Original sentence: EN

“Despite being scared of flying, I went to visit my sister in Japan.”

Translated sentence: CZ

"Přestože jsem se bál létání, šel jsem navštívit svou sestru v Japonsku.”

Paraphrased sentence: EN

"Although I was afraid of flying, I went to visit my sister in Japan.”

2. Sentence Parsing

Parse the backtranslated paraphrases using the Stanford parser.

→ Get the pair of constituency parses $\langle p_1, p_2 \rangle$ for each $\langle s_1, s_2 \rangle$.

"**She**	**drove**	**home**."	
(S(NP(PRP))	(VP(VBD)	(NP(NN)))	(.))

Relax the target syntactic form to a parse *template* (top two levels of the linearized parse tree):

"**She** **drove** **home**."

$S \rightarrow NP \quad VP$

→ Consider 20 most frequent templates in PARANMT-50M

To overcome learned biases, also include the reversed pairs $\langle s_2, s_1 \rangle$ are included during training

Adversarial Example Generation with SCPN

2. Sentence Parsing - Template filtering

template	paraphrase	
original	with the help of captain picard , the borg will be prepared for everything .	
(SBARQ(ADVP)(,)(S)(,)(SQ))	now , the borg will be prepared by picard , will it ?	
(S(NP)(ADVP)(VP))	the borg here will be prepared for everything .	
(S(S)(,)(CC)(S) (:)(FRAG))	with the help of captain picard , the borg will be prepared , and the borg will be prepared for everything ... for everything .	
(FRAG(INTJ)(,)(S)(,)(NP))	oh , come on captain picard , the borg line for everything .	← ✗ Failure
original	you seem to be an excellent burglar when the time comes .	
(S(SBAR)(,)(NP)(VP))	when the time comes , you 'll be a great thief .	
(S(``)(UCP)('')(NP)(VP))	" you seem to be a great burglar , when the time comes . " you said .	
(SQ(MD)(SBARQ))	can i get a good burglar when the time comes ?	
(S(NP)(IN)(NP)(NP)(VP)	look at the time the thief comes .	← ✗ Failure

Templates may be not be appropriate for particular input sentences (semantic divergence/ungrammatical)

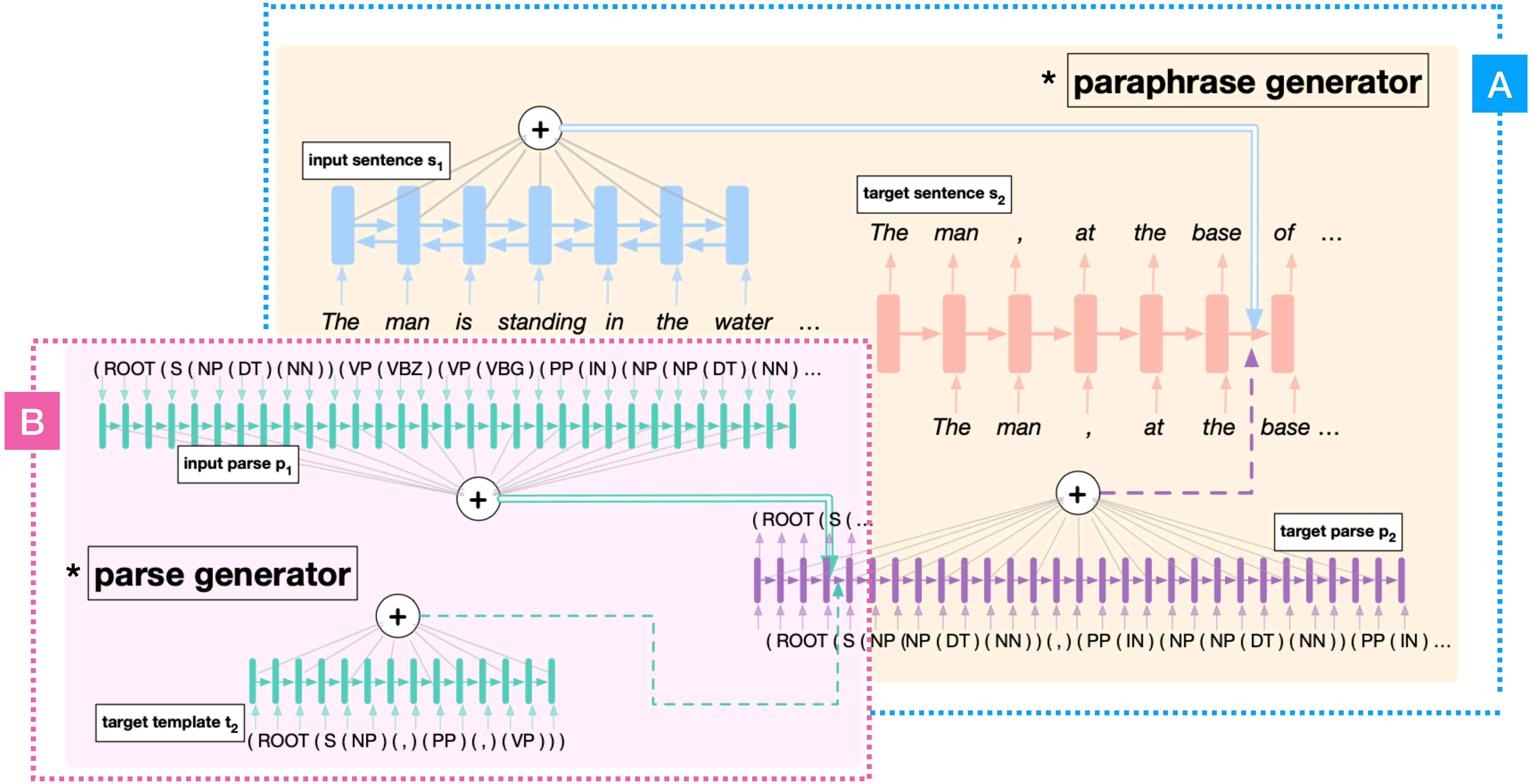
→ **Feedback mechanism from output**: generated paraphrases are filtered using n-gram overlap and paraphrastic similarity (Wieting and Gimpel, 2017).

Adversarial Example Generation with SCPN

3. Model - Parse Generator + SCPN

Given a paraphrase pair $\langle s_1, s_2 \rangle$ and corresponding target syntax trees $\langle p_1, p_2 \rangle$, the model is such that:

Inputs: s_1 and p_2 → **Output**: trained to produce s_2

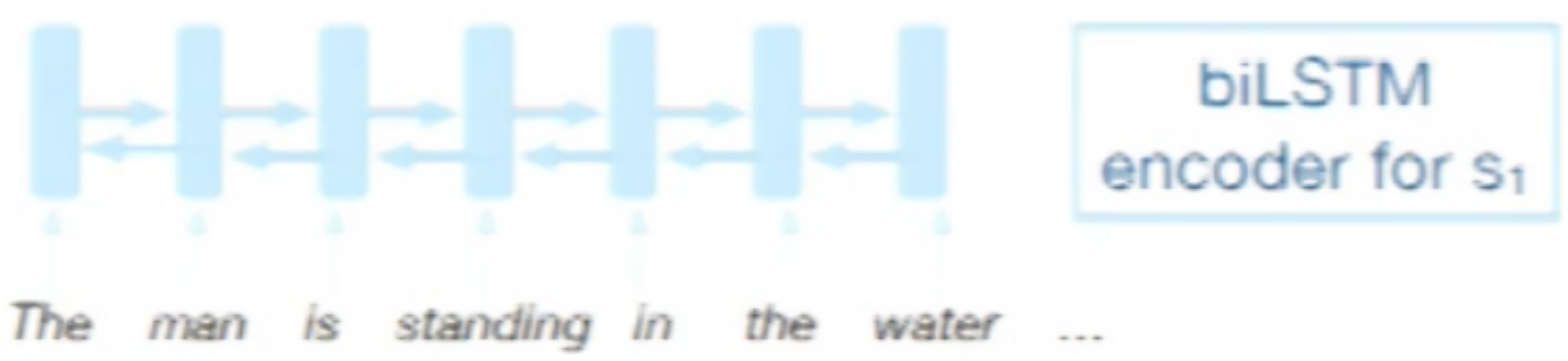


* trained separately

Adversarial Example Generation with SCPN

3. Paraphrase Generator Architecture - Encoder

1. Input sentence encoding



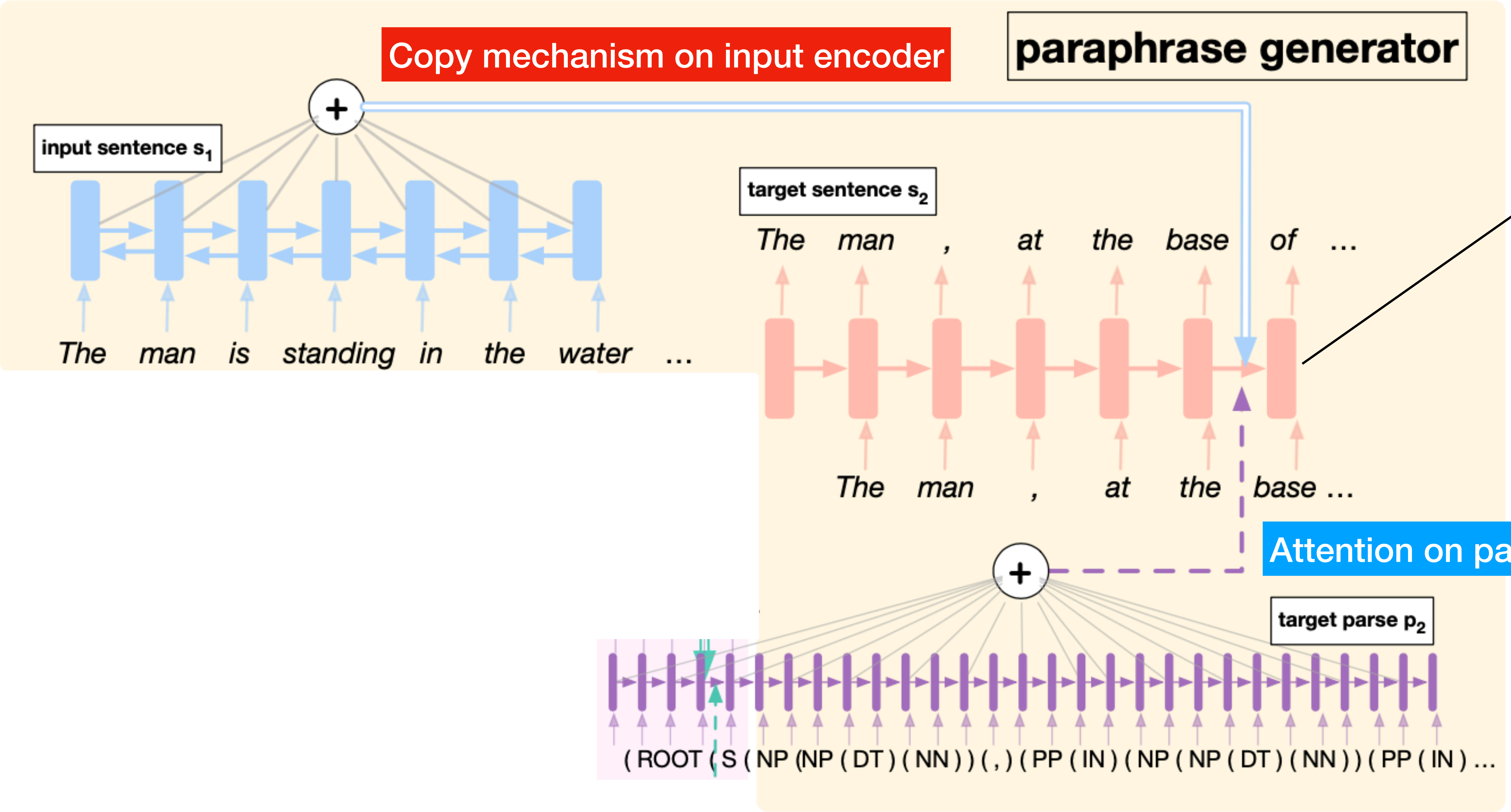
Adversarial Example Generation with SCPN

3. Paraphrase Generator Architecture - Encoder

1. Input sentence encoding

2. Linearized parse token sequence encoding

3. Paraphrase Generator Architecture - Decoder



LSTM decoder to produce s_2

$$h_t = LSTM([w_{t-1}; a_t; z_t])$$

previous word in s_2

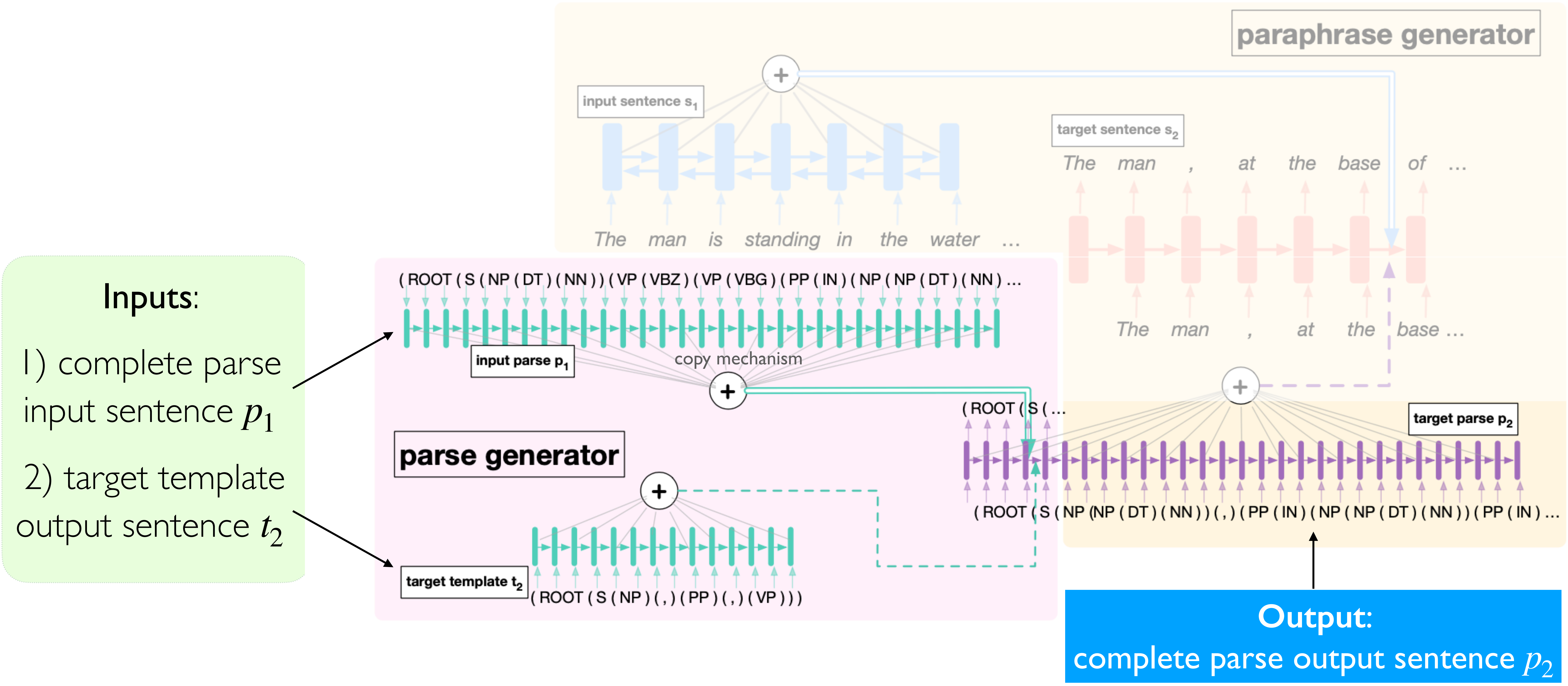
copy-mechanism over encoded input

attention-weighted average of LSTM parse hidden encoding

Adversarial Example Generation with SCPN

3. Parse Generator Architecture

Generate complete target parses from parse templates → similar architecture to the paraphrase generator.



Adversarial Example Generation with SCPN

Evaluation

1) **Intrinsic evaluations:** paraphrase quality, do the generated paraphrases follow the target distribution?
2) **Adversarial evaluations:** validity of adversarial examples, improvement in robustness of downstream models.

Baseline: NMT-BT → uncontrolled neural back-translation.

→ compare the ten most probable beams from NMT-BT to controlled paraphrases generated by SCPN

template	original	paraphrase
(S(ADVP)(NP)(VP))	moody , heartbreaking , and filmed in a natural , unforced style that makes its characters seem entirely convincing even when its script is not .	so he 's filmed in a natural , unforced style that makes his characters seem convincing when his script is not .
(S(PP)(,)(NP)(VP))	there is no pleasure in watching a child suffer .	in watching the child suffer , there is no pleasure .
	every nanosecond of the the new guy reminds you that you could be doing something else far more pleasurable .	each nanosecond from the new guy reminds you that you could do something else much more enjoyable .
	harris commands the screen , using his frailty to suggest the ravages of a life of corruption and ruthlessness .	harris commands the screen , using his weakness to suggest the ravages of life of corruption and recklessness .

→ **SCPN: Syntactic adversaries**

→ **NMT-BT: Lexical adversaries**

Adversarial Example Generation with SCPN

Intrinsic Evaluation

1) **Paraphrase quality:** score of a paraphrase pair ⟨source, generated⟩ by crowdworkers

→ SCPN vs. NMT-BT **outputs:** comparable in quality and grammatical correctness (but not in terms of syntactic difference from original).

→ Templates-fed vs. Full parses-fed SCPN **quality**: close to same.

2) Do the paraphrases follow the target specification?

Model	Parse Acc.
SCPN w/ gold parse	64.5
SCPN w/ generated parse	51.6
Parse generator	99.9

→ generated parses can differ from the ground-truth target parse in terms of ordering or existence of lower-level constituents

Accuracy is measured by exact template match (i.e., how often do the top two levels of the parses match).

Adversarial Example Generation with SCPN

Adversarial Evaluation

1) **Sentiment Analysis** - Stanford Sentiment Tree-bank (SST) (Socher et al., 2013)

→ contains complicated sentences with high syntactic variance.

2) **Entailment Detection** - SICK (Marelli et al., 2014)

→ almost exclusively consists of short, simple sentences.

			No augmentation		With augmentation	
Model	Task	Validity	Test Acc	Dev Broken	Test Acc	Dev Broken
SCPN	SST	77.1	83.1	41.8	83.0	31.4
NMT-BT	SST	68.1	83.1	20.2	82.3	20.0
SCPN	SICK	77.7	82.1	33.8	82.7	19.8
NMT-BT	SICK	81.0	82.1	20.4	82.0	11.2

SCPN generates more legitimate adversarial examples than NMT-BT

Augmenting data improves robustness of downstream models

Conclusions

SCPN:

- avoids lexical substitution in favor of making **syntactic changes**
- **paraphrases follow their target specifications** without decreasing paraphrase quality of unrestricted baselines.
- no quality drop when trained with **templates** vs. full parses.
- generates **valid adversarial** examples.

→ Possible future research:

- **Provide down-stream signals to SCPN** when training to allow for further lexical and syntax substitution.
- Dynamically integrates **templates based on** factors such as the **length of the input sentence.**

Adversarial Examples for Evaluating Reading Comprehension Systems

Jia and Liang (2017)

Adversarial Examples for Reading Comprehension

Contributions

- Show that simple adversarial attacks are effective against models trained on SQuAD.
- Analyse adversarial examples → evidence that many models trained on SQuAD rely on shallow heuristics, e.g. keyword matching.

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

(Jia and Liang, 2017)

Refresher: Reading Comprehension and SQuAD

- Input: (paragraph, question)
- Output: span of the paragraph
- Evaluation: F1 score

Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

By what main attribute are computational problems classified utilizing computational complexity theory?

Ground Truth Answers: inherent difficulty | their inherent difficulty | inherent difficulty

Prediction: inherent difficulty

(Rajpurkar et al., 2016)

Refresher: Limitations of SQuAD

- Questions were constructed looking at passages → lexical and syntactic overlap.
- Should be doable with type and keyword-matching.
- **Goal**: create an adversary that exploits this.

Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

By what main attribute are computational problems classified utilizing computational complexity theory?

Ground Truth Answers: inherent difficulty | their inherent difficulty | inherent difficulty

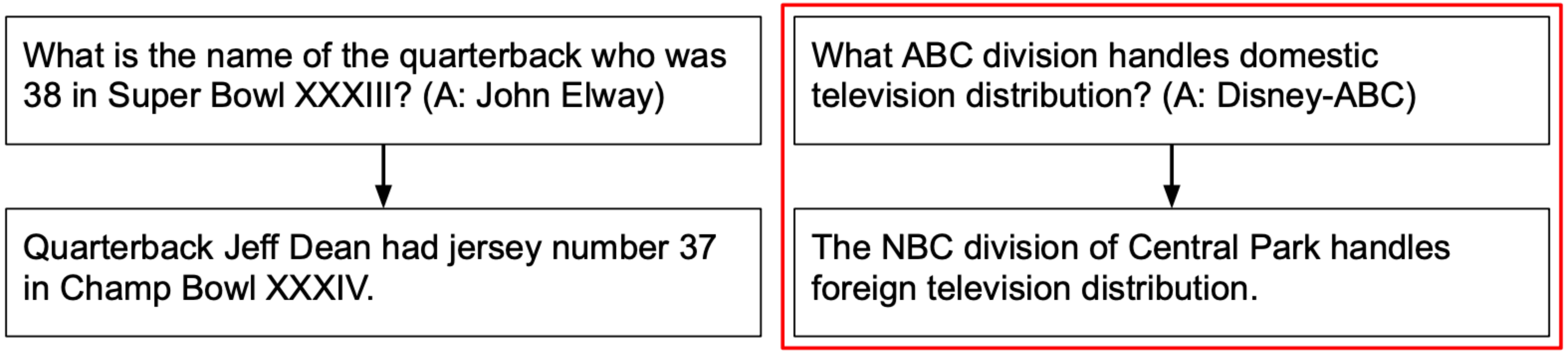
Prediction: inherent difficulty

(Rajpurkar et al., 2016)

Adversarial Examples for Reading Comprehension

Adversaries: AddSent and AddOneSent

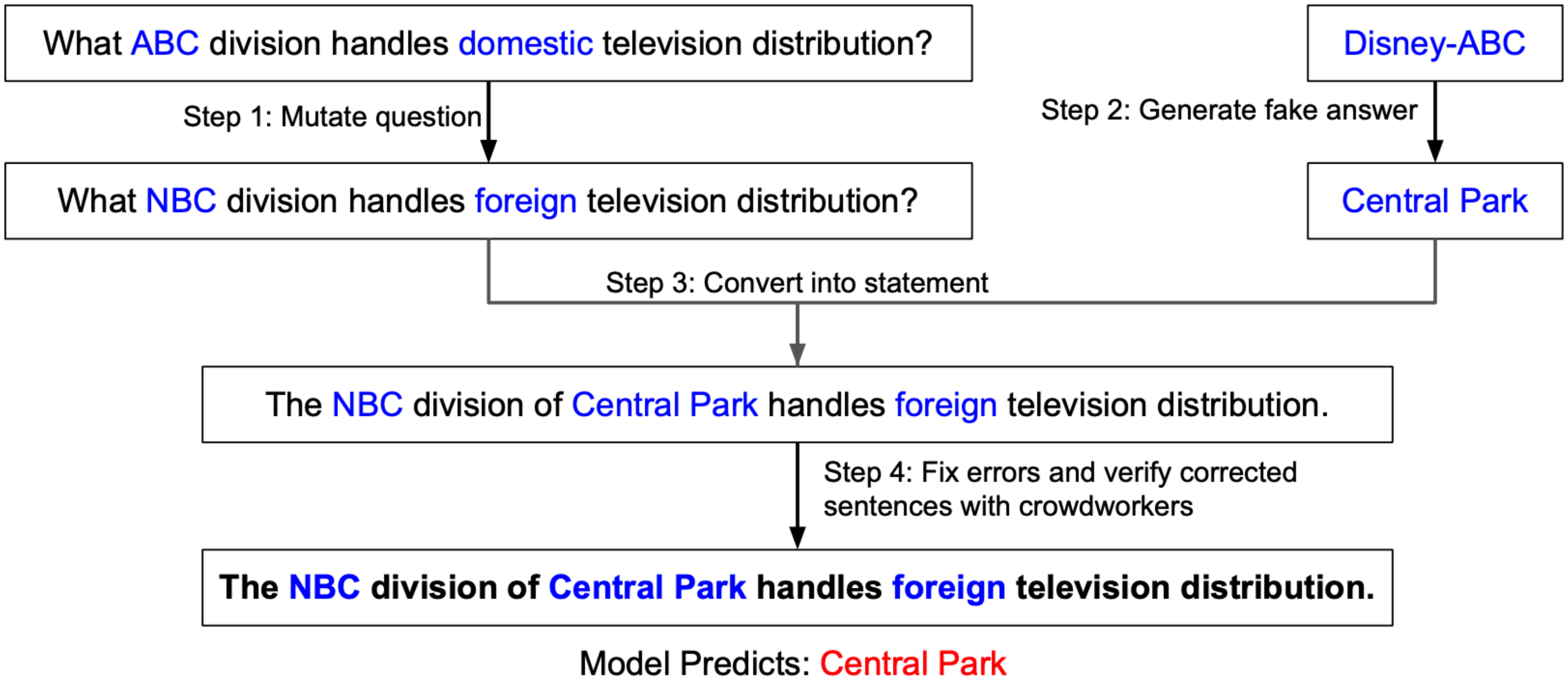
- **Concatenative**: append a distracting sentence to the input paragraph
- **Word-level** changes to the question/answer
  - High lexical overlap with the question but does not actually answer it
- Semantics-altering
- No dependence on the input paragraph



(Jia and Liang, 2017)

Adversarial Examples for Reading Comprehension

AddSent and AddOneSent: Overview



(Jia and Liang, 2017)

What are Adversarial Examples?

Step 1: Mutate Question

- Alter the question's semantics → generated sentence will not contradict the paragraph
  - Nouns, adjectives → antonyms from WordNet
  - Named entities, numbers → nearest word in GloVe embedding space with the same POS

What ABC division handles domestic television distribution?

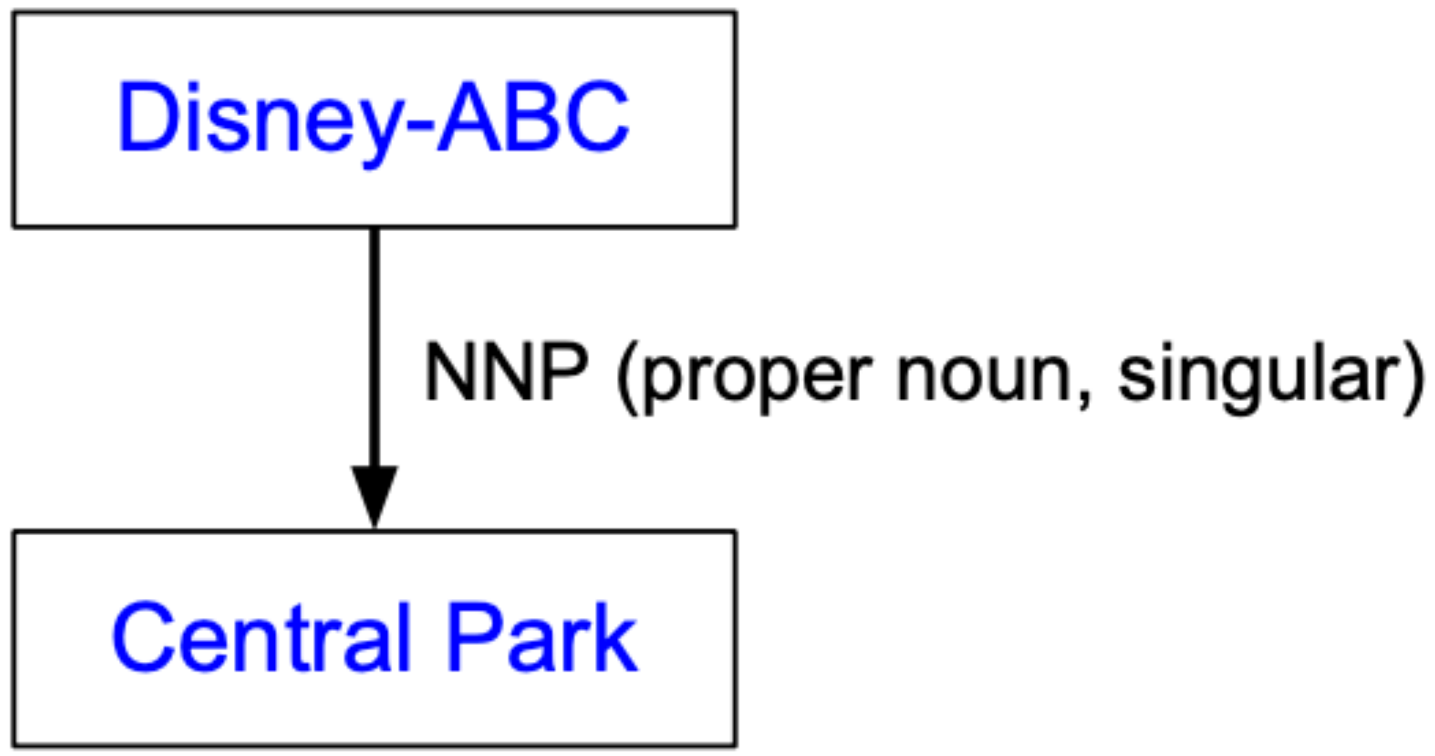
↓

What NBC division handles foreign television distribution?

(Jia and Liang, 2017)

Step 2: Generate Fake Answer

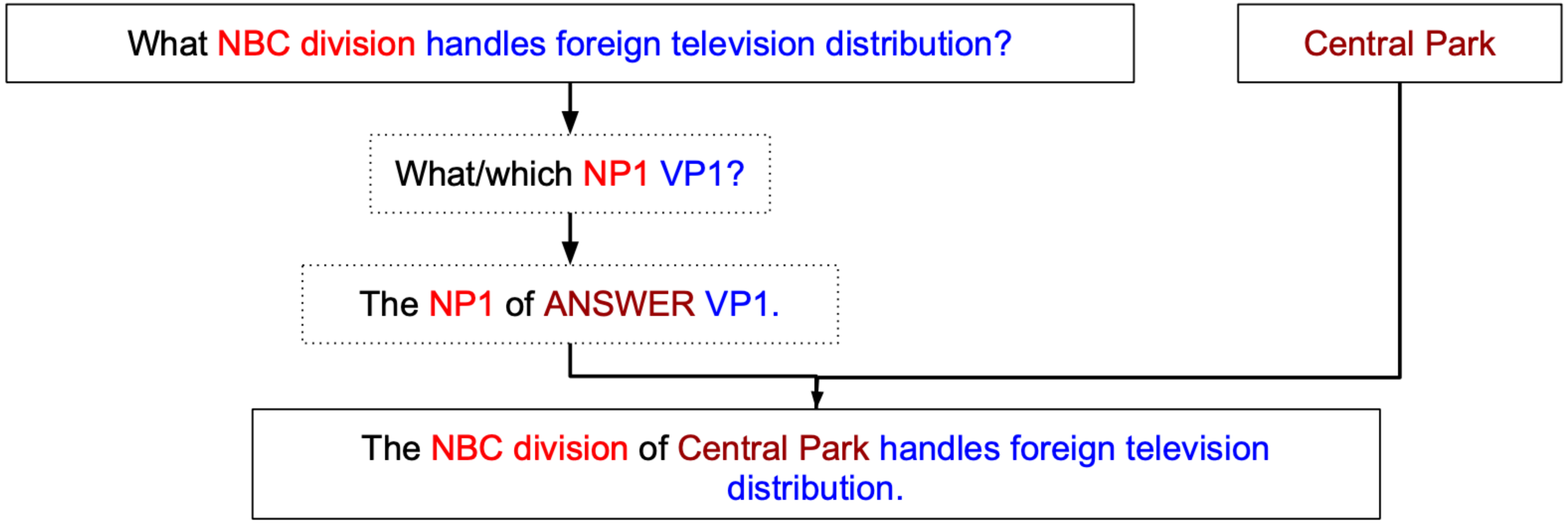
- Fake answer should have the same "type" as the original answer
- Predefine 26 types
  - NER and POS tags from Stanford CoreNLP
  - Custom categories e.g. abbreviations
- Fix a fake answer for each type



(Jia and Liang, 2017)

Step 3: Combine Fake Question/Answer

- Use 50 manually defined rules over CoreNLP constituency parses
- Incomplete and error-prone



(Jia and Liang, 2017)

Step 4: Fix Grammatical Errors

- Crowdsource via Amazon Mechanical Turk
- Edited independently by 5 workers → 5 sentences

- **AddSent**: try all 5 sentences on the model and choose the one where the model gives the worst (in terms of F1 score) answer
  - **This is the only part where the model is used!**

- **AddOneSent**: choose one of these 5 sentences randomly
  - **Completely model-independent**

(Jia and Liang, 2017)

Adversaries: AddAny

- **Still concatenative**
- But the appended "sentence" can be any **sequence of d words** → could (and will probably) be total nonsense
- Step 1: Initialise the words randomly from a list of common English words

Spring attention income getting reached

(Jia and Liang, 2017)

Adversaries: AddAny

- Step 2: Use local search to greedily change one word at a time to worsen the model's performance
- Search space: 20 randomly sampled common words and all words in the question
- Performance measure: **expected F1 score** over the model's output distribution
- **Requires several queries to the model and "grey-box" access to the output distribution**

(Jia and Liang, 2017)

AddAny: Example (d = 5)

Spring attention income getting reached

Spring attention income NBC reached

George attention income NBC reached

George handle handle NBC other

Model predicts: George

(Jia and Liang, 2017)

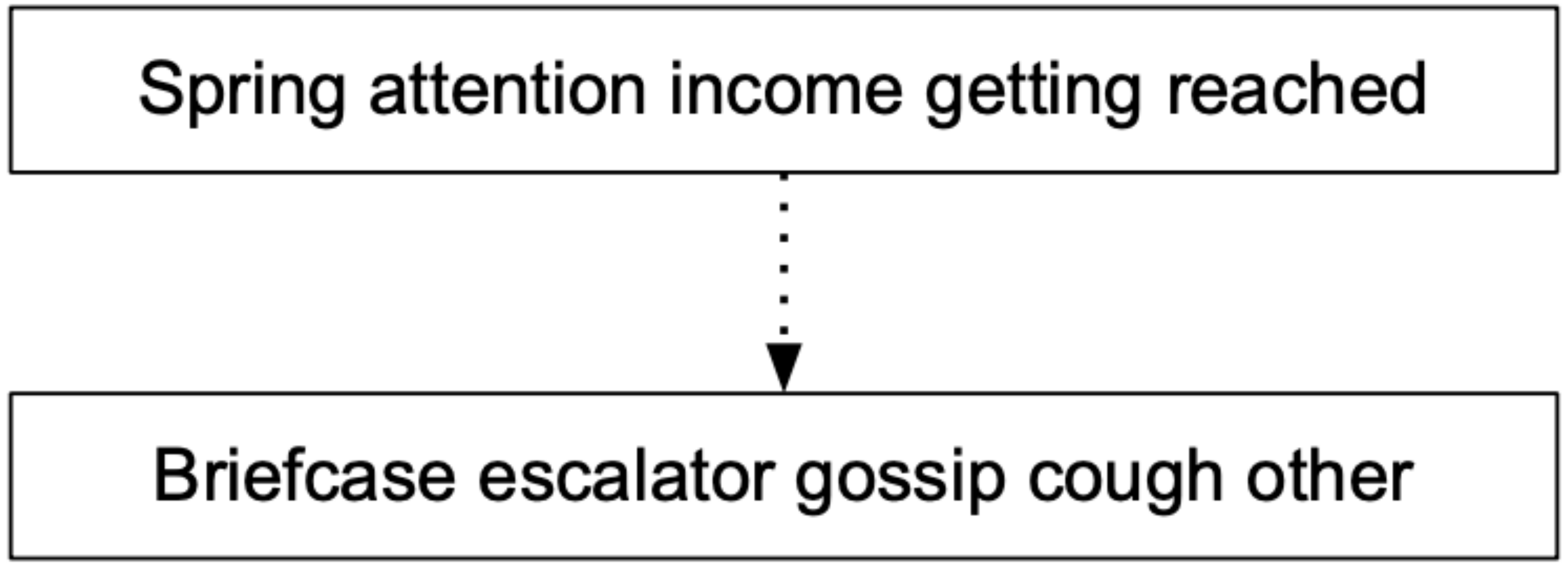
Adversarial Examples for Reading Comprehension

Adversaries: AddCommon

- All the adversaries so far rely in part on "baiting" the model with keywords from the question

- **Can we trick the model in a less straightforward way?**
  - Identify **subtler error patterns** of the model

- AddCommon: same as AddAny but the local search is restricted to common words



(Jia and Liang, 2017)

Adversaries: Overview

Adversary	Access to model	Appends sensible sentences	Uses words from question
AddSent	Black-box 5 queries/example	Y	Y
AddOneSent	Black-box Model-independent	Y	Y
AddAny	"Grey-box" 1000s of queries/example	N	Y
AddCommon	"Grey-box" 1000s of queries/example	N	N

(Jia and Liang, 2017)

Experiments: Setup

- Evaluate on 2 models during development
  - BiDAF (Seo et al, 2016)
  - Match-LSTM (Wang and Jiang, 2016)
  - Single and ensemble version for each

- Use 10 other models for validation as well

(Jia and Liang, 2017)

Results: Main Experiments

	Match Single	Match Ens.	BiDAF Single	BiDAF Ens.
Original	71.4	75.4	75.5	80.0
ADDSENT	27.3	29.4	34.3	34.2
ADDONESENT	39.0	41.8	45.7	46.9
ADDANY	7.6	11.7	4.8	2.7
ADDCOMMON	38.9	51.0	41.7	52.6

Mnemonic Reader models long-range dependencies within the paragraph → can locate correct answer

Model	Original	ADDSENT	ADDONESENT
ReasoNet-E	**81.1**	39.4	49.8
SEDT-E	80.1	35.0	46.5
BiDAF-E	80.0	34.2	46.9
Mnemonic-E	79.1	**46.2**	**55.3**
Ruminating	78.8	37.4	47.7
jNet	78.6	37.9	47.0
Mnemonic-S	78.5	**46.6**	**56.0**
ReasoNet-S	78.2	39.4	50.3
MPCM-S	77.0	40.3	50.0
SEDT-S	76.9	33.9	44.8
RaSOR	76.2	39.5	49.5
BiDAF-S	75.5	34.3	45.7
Match-E	75.4	29.4	41.8
Match-S	71.4	27.3	39.0
DCR	69.3	37.8	45.1
Logistic	50.4	23.2	30.4

(Jia and Liang, 2017)

Results: Human Evaluation

- **This is important!** If humans are consistently getting adversarial examples "wrong" then the examples are not valid.
- AddSent < AddOneSent only because humans naturally make mistakes

	Human
Original	92.6
ADDSENT	79.5
ADDONESENT	89.2

(Jia and Liang, 2017)

Results: AddSent Error Analysis

Question: The number of Huguenot colonists declined after what year?

Paragraph: The largest portion of the **Huguenots** to **settle** in the Cape arrived between 1688 and 1689, in seven ships as part of the organised migration, but quite a few arrived as late as **1700**; **thereafter**, the **numbers declined**, and only small groups arrived at a time. The **number** of old Acadian **colonists declined** **after** the **year** of **1675**.

Correct answer: **1700**

Model predicts: **1675**

A partial match with the question is enough to distract the model.

(Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)

Results: AddAny Error Analysis

Question: What city did Tesla move to in 1880?

Paragraph: In January **1880**, two of **Tesla's** uncles put together enough money to help him leave Gospić for **Prague**… **what** 30 **city 1880 what move city city** medical **move.**

Attack draws heavily from question keywords/related words.

Correct answer: **Prague**

Model predicts: **medical**

(Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)

Results: AddCommon Error Analysis

Question: Where did he (Tesla) claim the blueprint was stored?

Paragraph: During the period in which the negotiations were being conducted... the **blueprint** for the teleforce weapon was all **in his mind**. Doubt was did about carried wasn't year 1961 near policy.

Correct answer: **in his mind**

Model predicts: **near policy**

Some strange attacks are beyond our intuition about keyword matching!

(Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)

Results: Transferability Across Models

- AddSent examples transfer well, AddAny examples do not
- Suggests that the attacks exploit general limitations of SQuAD rather than model-specific limitations

Targeted Model	Model under Evaluation ML Single	ML Ens.	BiDAF Single	BiDAF Ens.
ADDSENT				
ML Single	27.3	33.4	40.3	39.1
ML Ens.	31.6	29.4	40.2	38.7
BiDAF Single	32.7	34.8	34.3	37.4
BiDAF Ens.	32.7	34.2	38.3	34.2
ADDANY				
ML Single	7.6	54.1	57.1	60.9
ML Ens.	44.9	11.7	50.4	54.8
BiDAF Single	58.4	60.5	4.8	46.4
BiDAF Ens.	48.8	51.1	25.0	2.7

(Jia and Liang, 2017)

Results: Adversarial Training

- AddSentMod:
    - Use a different set of fake answers for each type e.g. Jeff Dean → Charles Babbage
    - Prepend (rather than append) the adversarial sentence to the paragraph
- Model overfits the adversary used for training

Test data	Training data: Original	Training data: Augmented
Original	75.8	75.1
AddSent	34.8	70.4
AddSentMod	34.3	$\boxed{39.2}$

(Jia and Liang, 2017)

Adversarial Examples for Reading Comprehension

Takeaways

- Adversarial examples can expose models that rely on shallow heuristics and provide insights into these heuristics
- They can also expose datasets that are simpler than they seem
- Just appending a sentence is effective as an attack
- **For future work**: Haven't successfully used adversarial examples to train robust models yet

Quiz time!

Discussion Question #2

Q: Both papers investigated the effect of training the model on these generated adversarial examples. Do you think this would eventually fix the problem or not?

Discussion Question #2

Q: Both papers investigated the effect of training the model on these generated adversarial examples. Do you think this would eventually fix the problem or not?

Answer:

- Adversarial training can help in some applications - notable success in computer vision.
- But this is harder in NLP's discrete space. Can help in improving the robustness of the model at test time, and also in reducing its likelihood to "break" at train time, but far from a solution.
- *Belinkov and Bisk, 2018*
    - some types of adversarial examples do not improve robustness as the model is incapable of learning any patterns.
    - training on a specific type of error/adv. example does not allow to generalise on other errors.

Natural vs. Synthetic Noise: Success in Improving Robustness? - Belinkov and Bisk, 2018

Natural errors: collected from real examples at word level (e.g. Wikipedia edit histories, manually annotated essays written by non-native speakers, etc.), across 3 languages - German, French and Czech.

Synthetic erros: four types of noise

- Swap: e.g. *noise* → *nosie*
- Middle Random: e.g. *noise* → *nisoe*
- Fully Random: e.g. *noise* → *nisoe*
- Keyboard typo: e.g. *noise* → *noide*

Natural vs. Synthetic Noise: Success in Improving Robustness? - Belinkov and Bisk, 2018

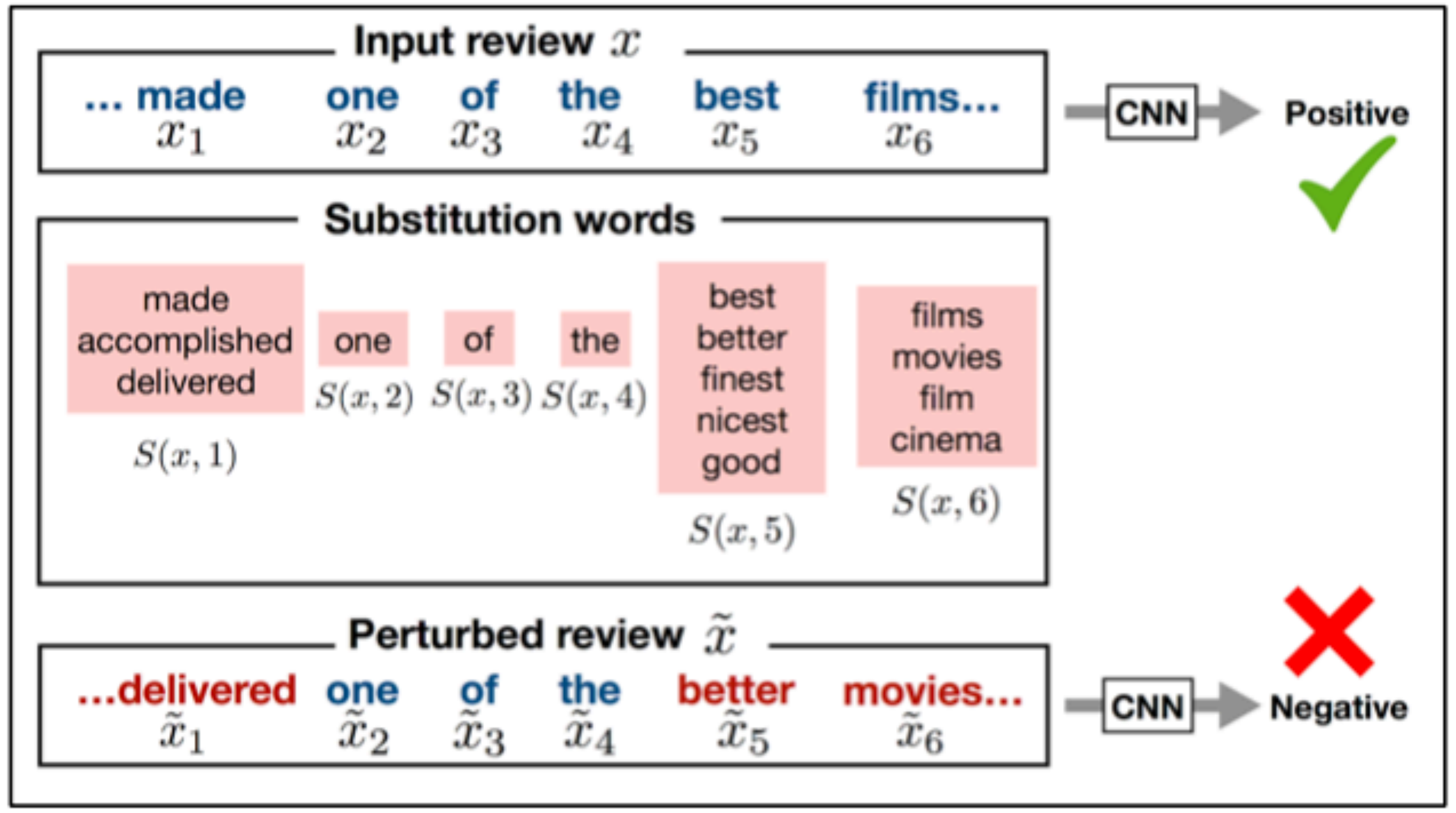
Results (BLEU Scores)

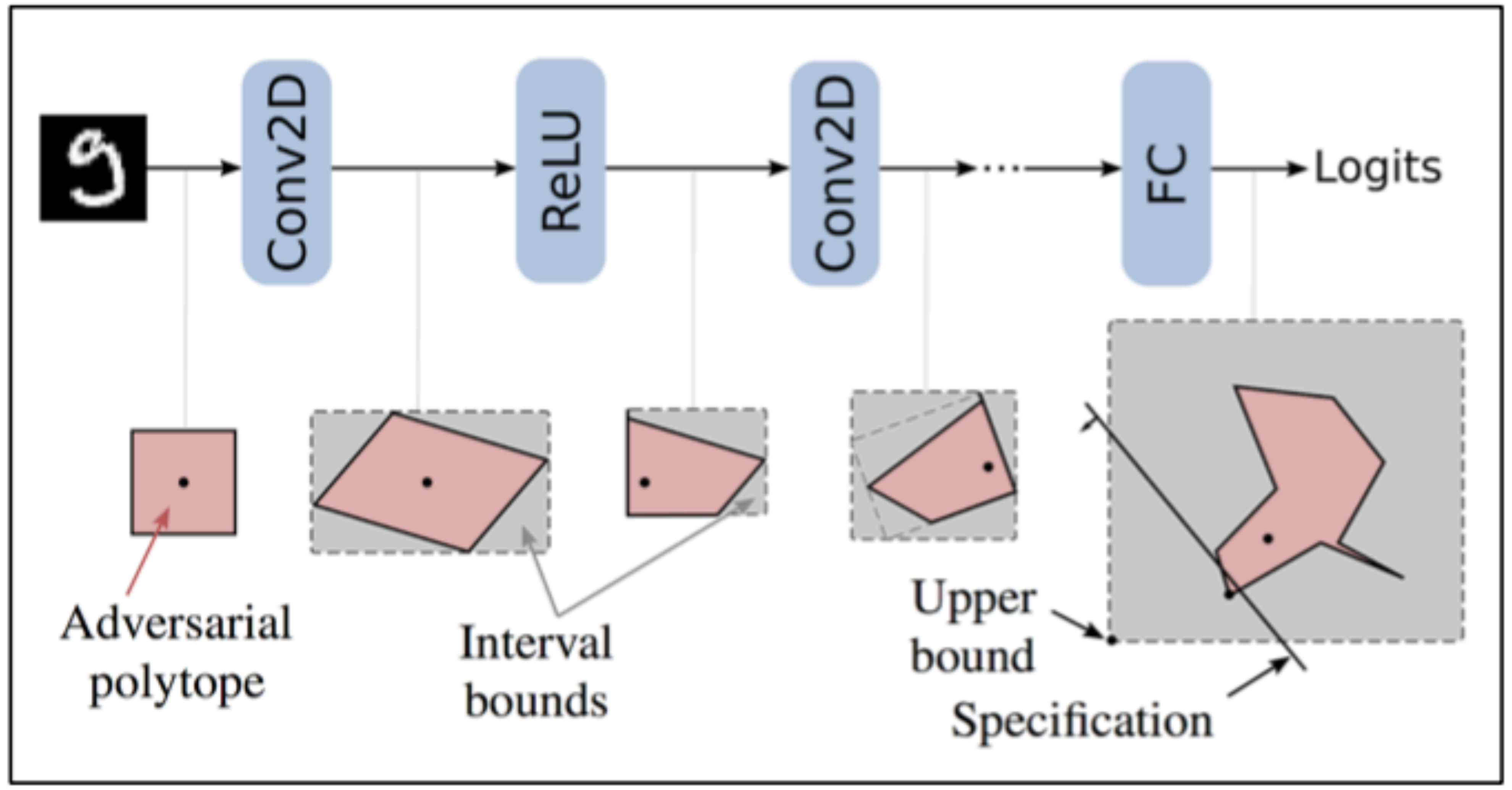
		Vanilla	Synthetic Swap	Synthetic Mid	Synthetic Rand	Synthetic Key	Nat
French	`charCNN`	42.54	10.52	9.71	1.71	8.26	17.42
German	`charCNN`	34.79	9.25	8.37	1.02	6.40	14.02
	`char2char`	29.97	5.68	5.46	0.28	2.96	12.68
	`Nematus`	34.22	3.39	5.16	0.29	0.61	10.68
Czech	`charCNN`	25.99	6.56	6.67	1.50	7.13	10.20
	`char2char`	25.71	3.90	4.24	0.25	2.88	11.42
	`Nematus`	29.65	2.94	4.09	0.66	1.41	11.88

- **Significant drop in BLEU when evaluated on noisy texts** → the more the noise the worse.
- Worst results on **languages with complex structures** (Czech).
- Other results: training on a **specific type of noise** makes the model more robust to that type of noise, but not to others (except random which never improves robustness).

Certified Robustness to Word-Level Attacks!

- Interval Bound Propagation → upper bound on the model's loss for **any combination of these substitutions**
- Optimise this upper bound directly!





(Jia et al., 2019; Gowal et al., 2018)

Comparison with Data Augmentation

System	Genetic attack (Upper bound)	IBP-certified (Lower bound)
Standard training		
BoW	9.6	0.8
CNN	7.9	0.1
LSTM	6.9	0.0
Robust training		
BoW	70.5	68.9
CNN	**75.0**	**74.2**
LSTM	64.7	63.0
Data augmentation		
BoW	34.6	3.5
CNN	35.2	0.3
LSTM	33.0	0.0

Sentiment Analysis on IMDB

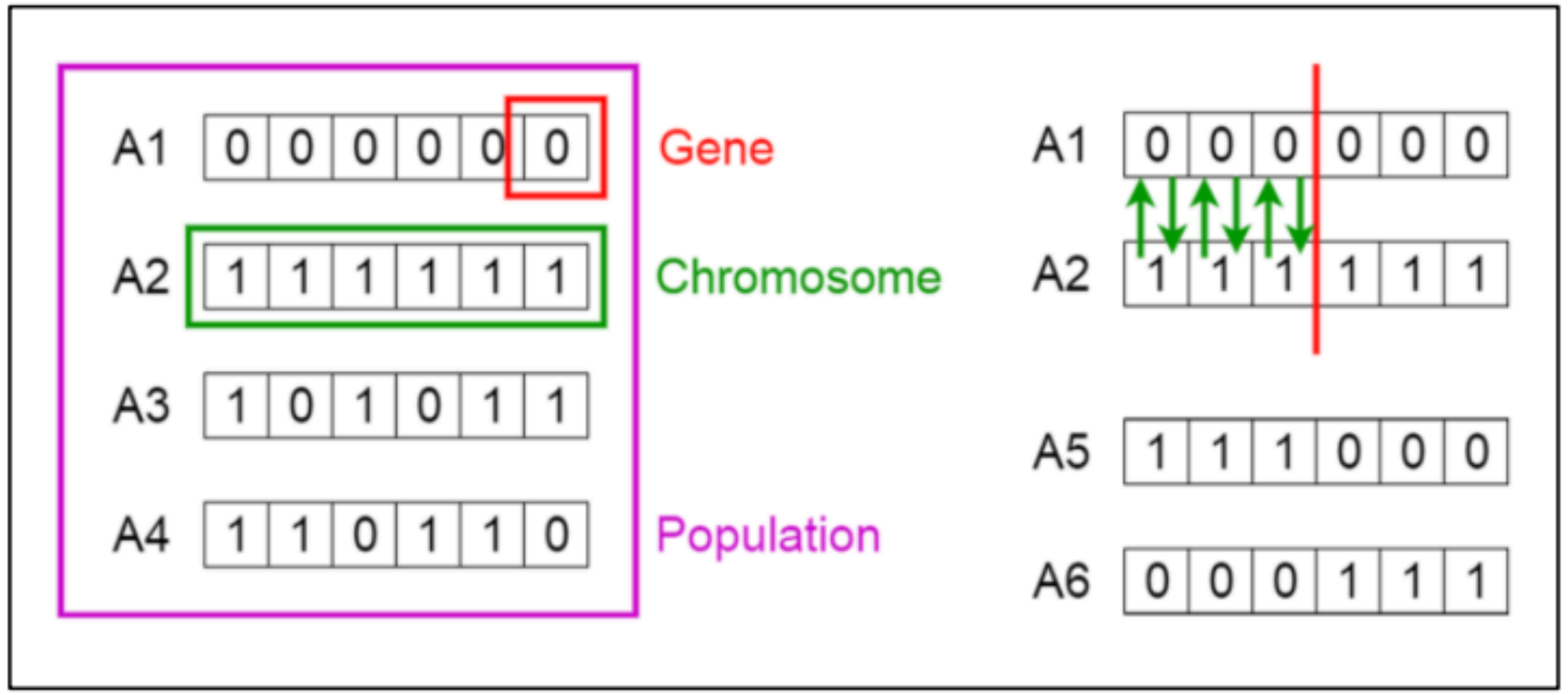
System	Genetic attack (Upper bound)	IBP-certified (Lower bound)
Normal training		
BoW	40.5	2.3
DecompAttn	40.3	1.4
Robust training		
BoW	**75.0**	**72.7**
DecompAttn	73.7	72.4
Data augmentation		
BoW	68.5	7.7
DecompAttn	70.8	1.4

Textual Entailment on SNLI

(Jia et al., 2019)

Genetic Algorithms to Generate Examples

- **Semantics-preserving**
- **Word-level** perturbations
- **Grey-box** (access to output probabilities)



Original Text Prediction = **Negative**. (Confidence = 78.0%)

This movie had terrible acting, terrible plot, and terrible choice of actors. (Leslie Nielsen ...come on!!!) the one part I considered slightly funny was the battling FBI/CIA agents, but because the audience was mainly kids they didn't understand that theme.

Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)

*This movie had **horrific** acting, **horrific** plot, and **horrifying** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **youngsters** they didn't understand that theme.*

Original Text Prediction: **Entailment** (Confidence = 86%)

Premise: *A runner wearing purple strives for the finish line.*

Hypothesis: *A runner wants to head for the finish line.*

Adversarial Text Prediction: **Contradiction** (Confidence = 43%)

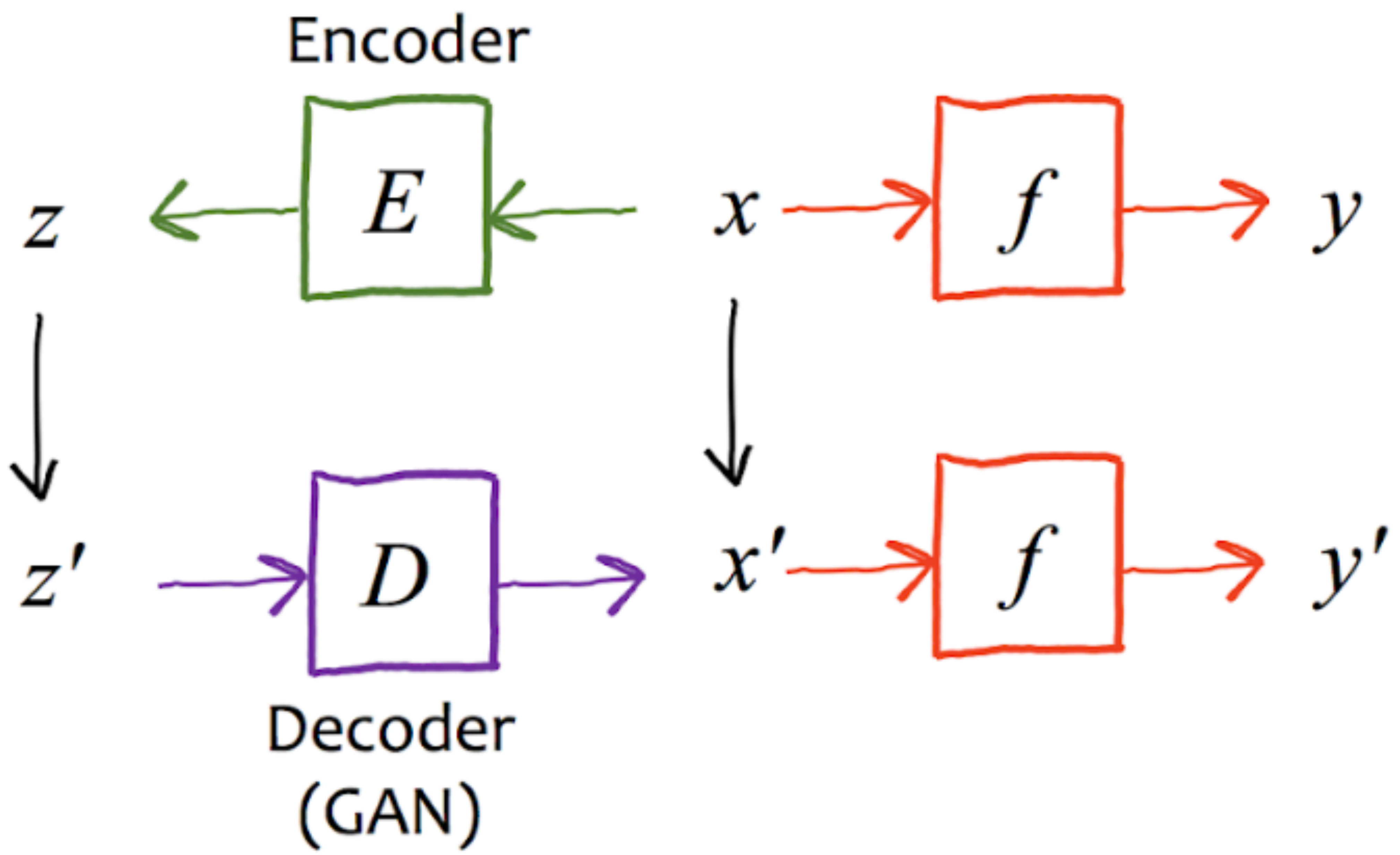
Premise: *A runner wearing purple strives for the finish line.*

Hypothesis: *A **racer** wants to head for the finish line.*

(Alzantot et al., 2018; Mallawaarachchi, 2017)

Generating Natural Adversarial Examples

- Search in **continuous space** via **sentence embeddings**
- **Black-box, sentence-level** perturbations
- Applied to computer vision as well



$$\min_{x'} \|z - z'\| \quad \text{s.t. } f(x') \neq f(x)$$

(Zhao et al., 2018; Singh, 2019)

Semantically Equivalent Adversarial Rules

- Extract general patterns from backtranslation attacks

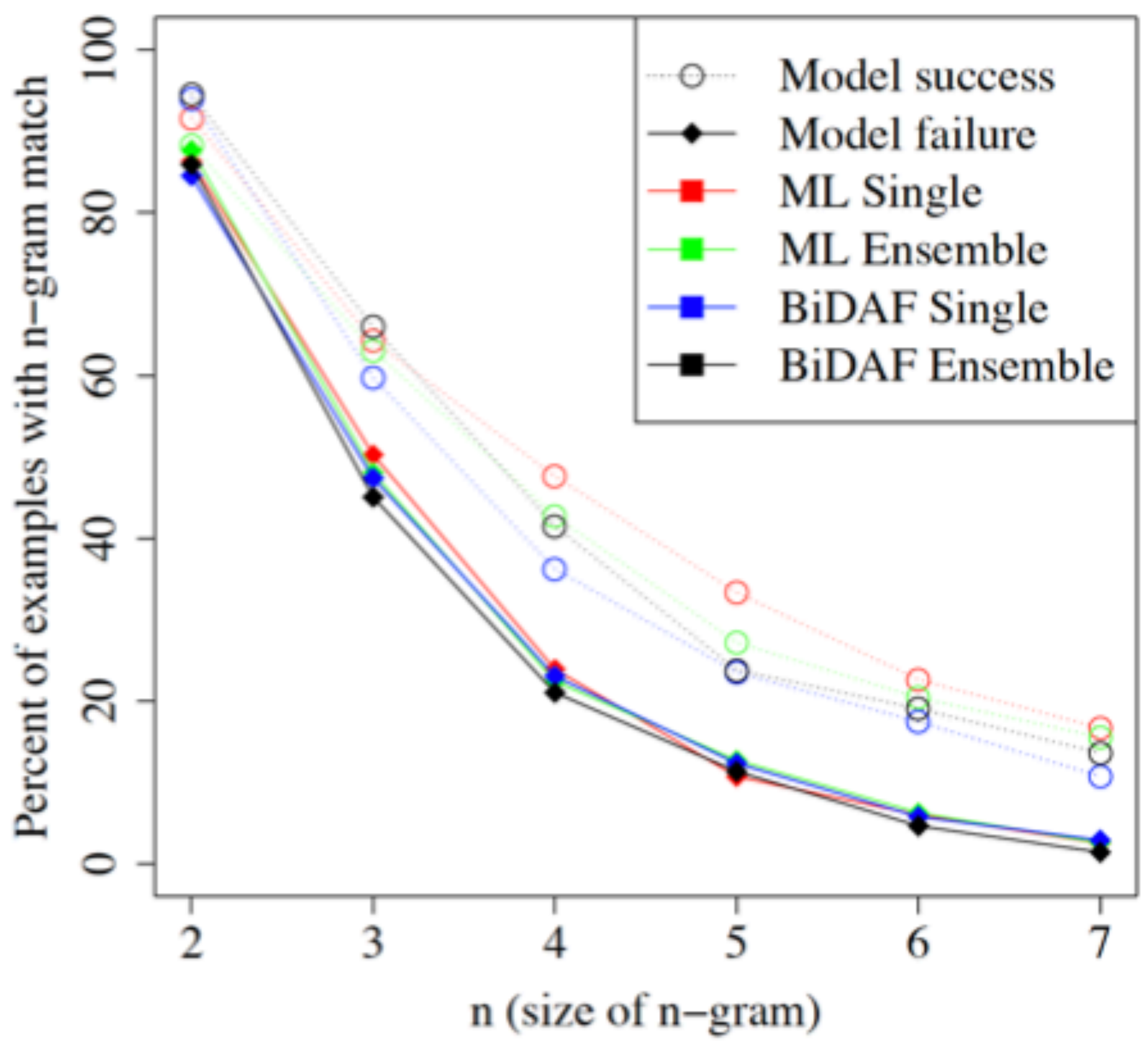
SEAR	Questions / SEAs	f(x)	Flips
What VBZ → What's	~~What is~~ What's the NASUWT?	~~Trade union~~ Teachers in Wales	2%
What NOUN → Which NOUN	~~What resource~~ Which resource was mined in the Newcastle area?	~~coal~~ wool	1%
What VERB → So what VERB	~~What was~~ So what was Ghandi's work called?	~~Satyagraha~~ Civil Disobedience	2%
What VBD→ And what VBD	~~What was~~ And what was Kenneth Swezey's job?	~~journalist~~ sleep	2%

(Ribeiro et al., 2018; Singh, 2019)

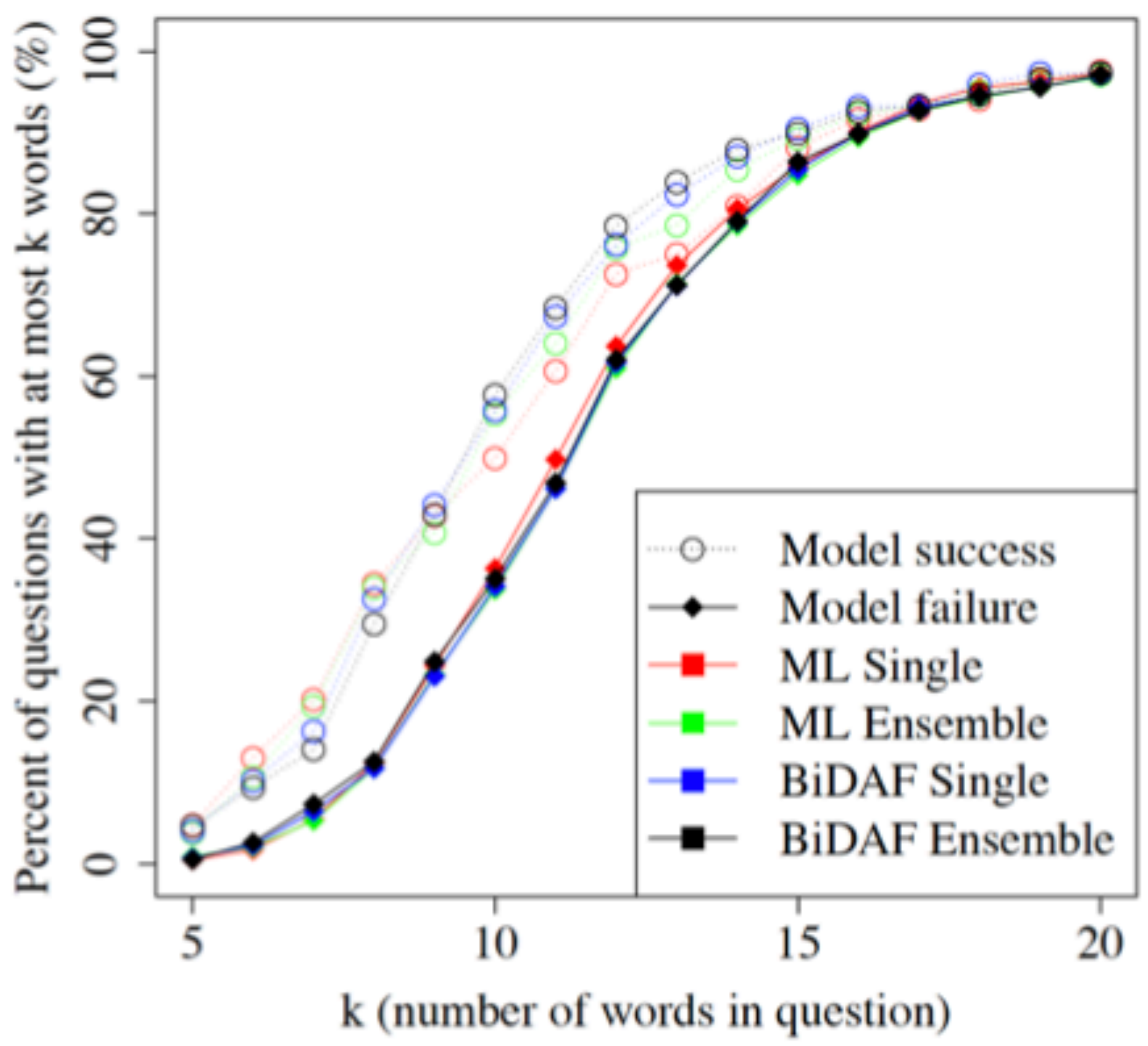
Bonus Slide and References

Bonus Slide!

Results: Reasons for Model Success



Models did better with an exact n-gram match between question and paragraph.



Models also did better with short questions.

(Jia and Liang, 2017)

References

- Belinkov and Bisk, 2018: https://arxiv.org/pdf/1711.02173.pdf
- Ebrahimi et al., 2017: https://arxiv.org/pdf/1712.06751.pdf
- Ebrahimi et al., 2018: https://arxiv.org/pdf/1806.09030.pdf
- Alzantot et al., 2018: https://arxiv.org/abs/1804.07998
- Goodfellow et al., 2015: https://arxiv.org/abs/1412.6572
- Gowal et al., 2018: https://arxiv.org/abs/1810.12715
- Jia, 2017: https://vimeo.com/238231419
- Jia and Liang, 2017: https://arxiv.org/abs/1707.07328
- Jia et al., 2019: https://arxiv.org/abs/1909.00986
- Mallawaarachchi, 2017: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
- Rajpurkar et al., 2016: https://arxiv.org/abs/1606.05250
- Ribeiro et al., 2018: https://www.aclweb.org/anthology/P18-1079.pdf
- Singh, 2019: http://sameersingh.org/files/ppts/naacl19-advnlp-part2-sameer-slides.pdf
- Zhao et al., 2018: https://arxiv.org/abs/1710.11342

Thank you

Any questions?