Adversarial Examples for Evaluating Reading Comprehension Systems

Robin Jia, Percy Liang

Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, Cho-Jui Hsieh

Presented by Adithya Murali

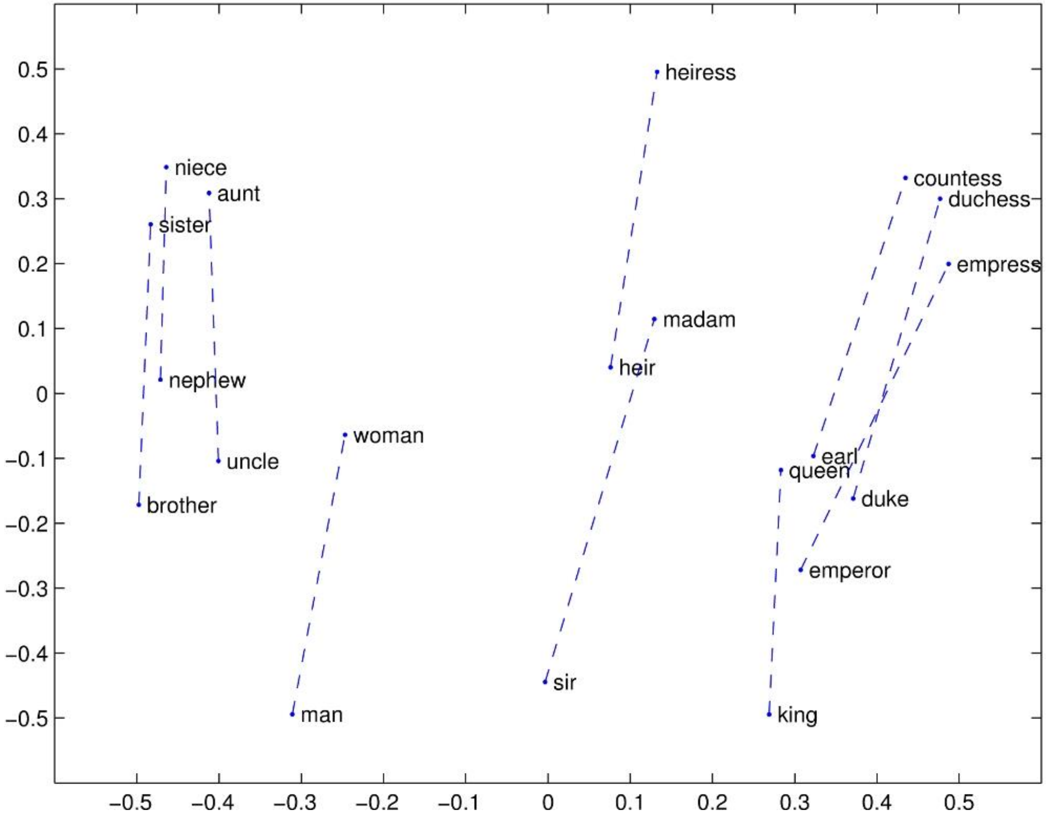
Building Adversaries for NLP Models

The police **helped** the protestors

The police **arrested** the protestors

Small perturbations -> Big changes

Discrete input and output space!



Summary

Adv for Comprehension	Seq2Sick
For reading comprehension systems: (p, q, a)	For seq-to-seq models (example: translation)
Blackbox; Untargeted	Whitebox; Targeted
Evaluates overstability	Evaluates oversensitivity
No input perturbation: additive method	Optimisation-based input perturbation
Human-in-the-loop	Fully mechanized

Adversarial Examples for Reading Comprehension

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

How is the problem formulated?		How is the solution formulated?
Be close to the original input	→	Add an irrelevant sentence to end of paragraph
Make sense to a human	→	Human-in-the-loop algorithm
Fools model	→	Make distractor sentence from question

Objective

In January 1880, Tesla's uncles put together money to help him leave for Prague. Unfortunately, he arrived too late.

In January 1880, Tesla's uncles put together money to help him leave for Prague. Unfortunately, he arrived too late. **Tadakatsu moved to the city of Chicago in 1881.**

Paragraph, Question → Answer span

Against overstability, not oversensitivity

Average F1 score minimisation

Question: Which city did Tesla move to in 1880?

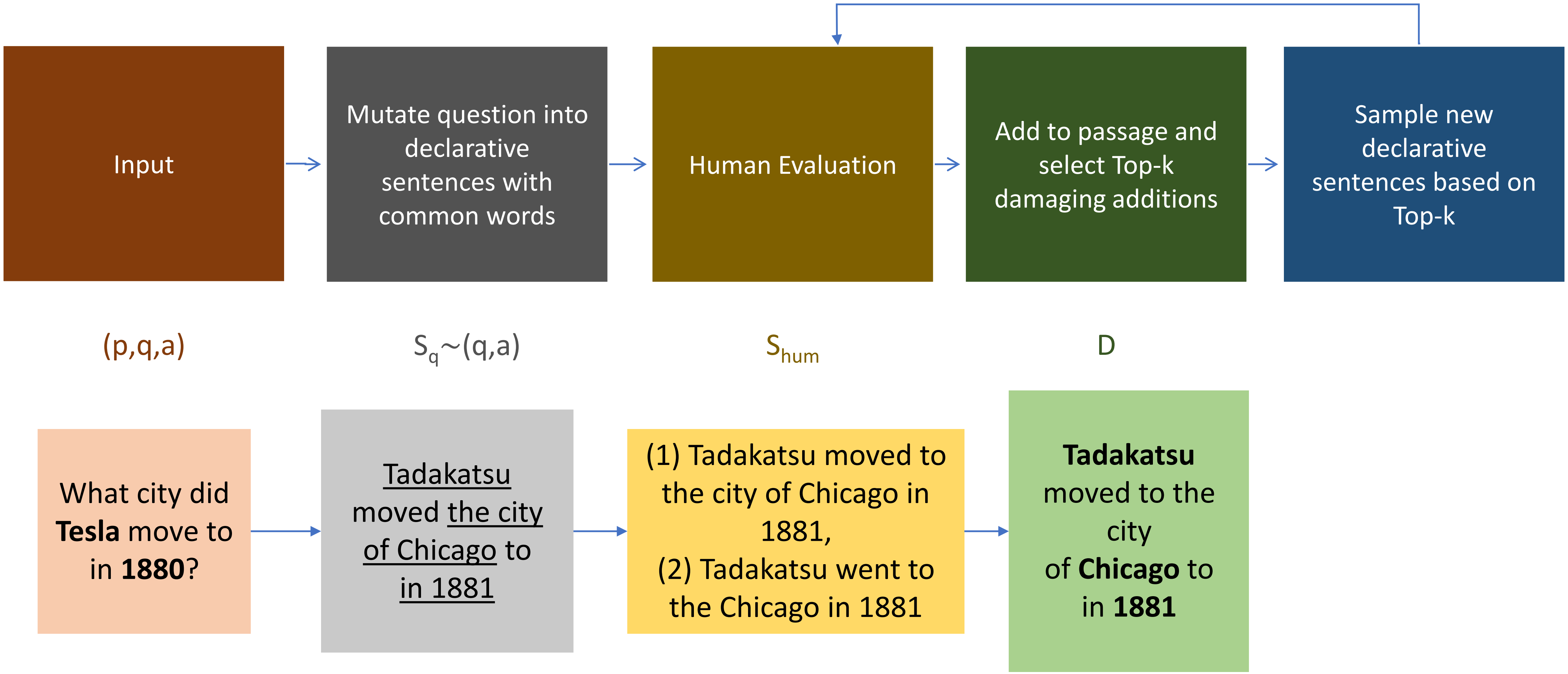
Answer: Prague

Answer: Chicago

Tesla moved to the city of Prague in 1880 ≠ Tadakatsu moved to the city of Chicago in 1881

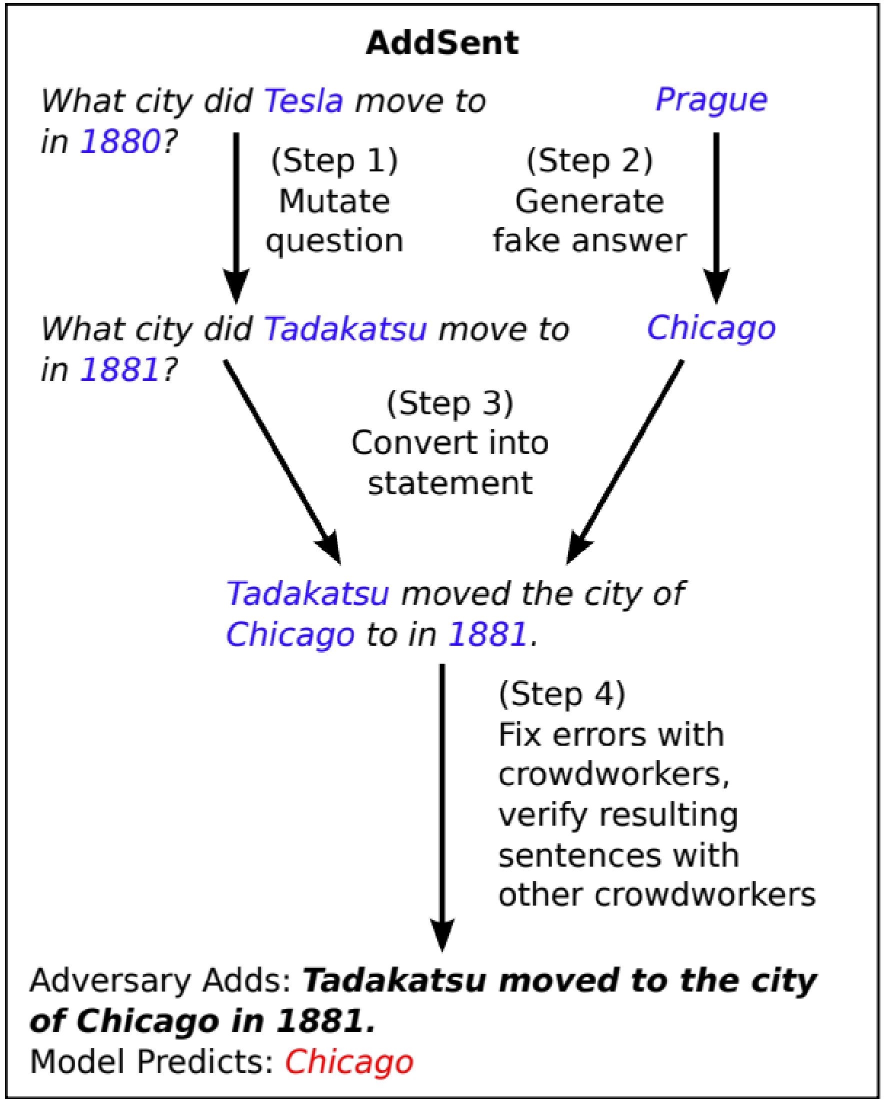
$$AdvAccuracy(AdvModel, D) = \frac{1}{|D|} \sum_{(p,q,a) \in D} F1(AdvModel(p, q, a, OrigModel), a)$$

General Algorithm



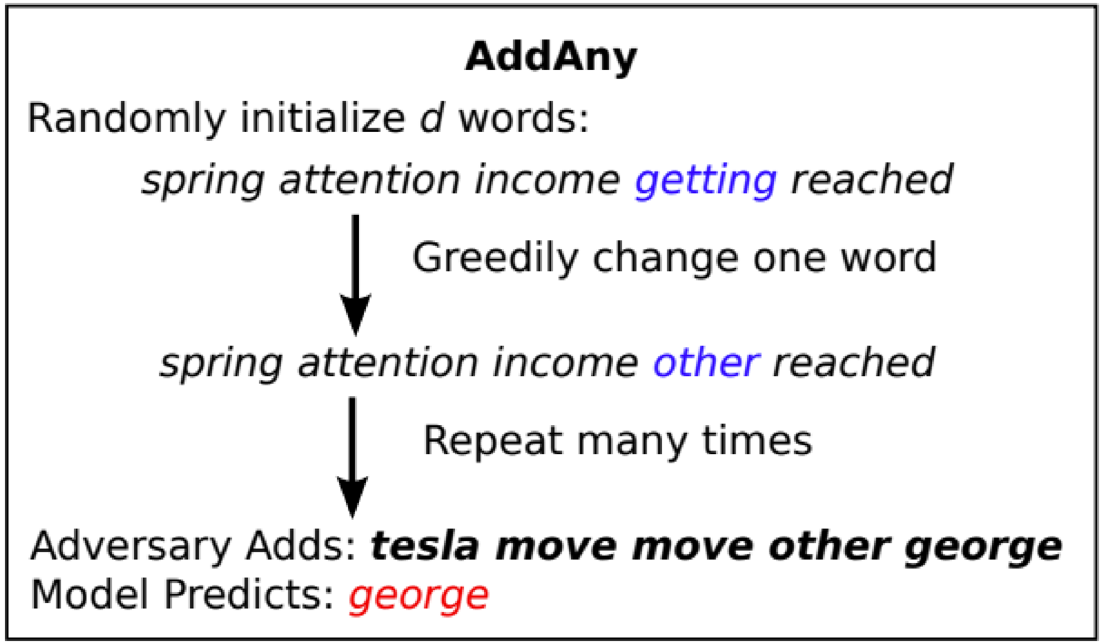
Mutation Method: ADDSENT

- One pass, no loop
- Modify q to *q'*
  - Nouns, Adjectives -> antonyms
  - Numbers, Named Entities -> nearest GloVe word with the same POS tag
- Modify *a* to *a'*
  - predefined choice with the same POS and NER tags
- Make declarative sentence to state *a'* satisfies *q'*
  - Trivial example: It is the case that *a'* is the answer to the question *q'*
  - General case: rules based on constituency parse
- Variation ADDONESENT: Randomly sample one from S_{hum} instead of Top-k
  - Model independent



Mutation Method: ADDANY

- Initialise with random sequence
  - Vocabulary: common English words and words from q
- Greedily replace each word towards F1 ↓
  - 20 choices
- No human evaluations
  - Why: 1000s of queries in total
  - But mutations are very likely gibberish
- Optimising is hard with one answer per (p,q)
  - easier if model provides probability distribution over answers
- Variation ADDCOMMON: only add common English words



Evaluation

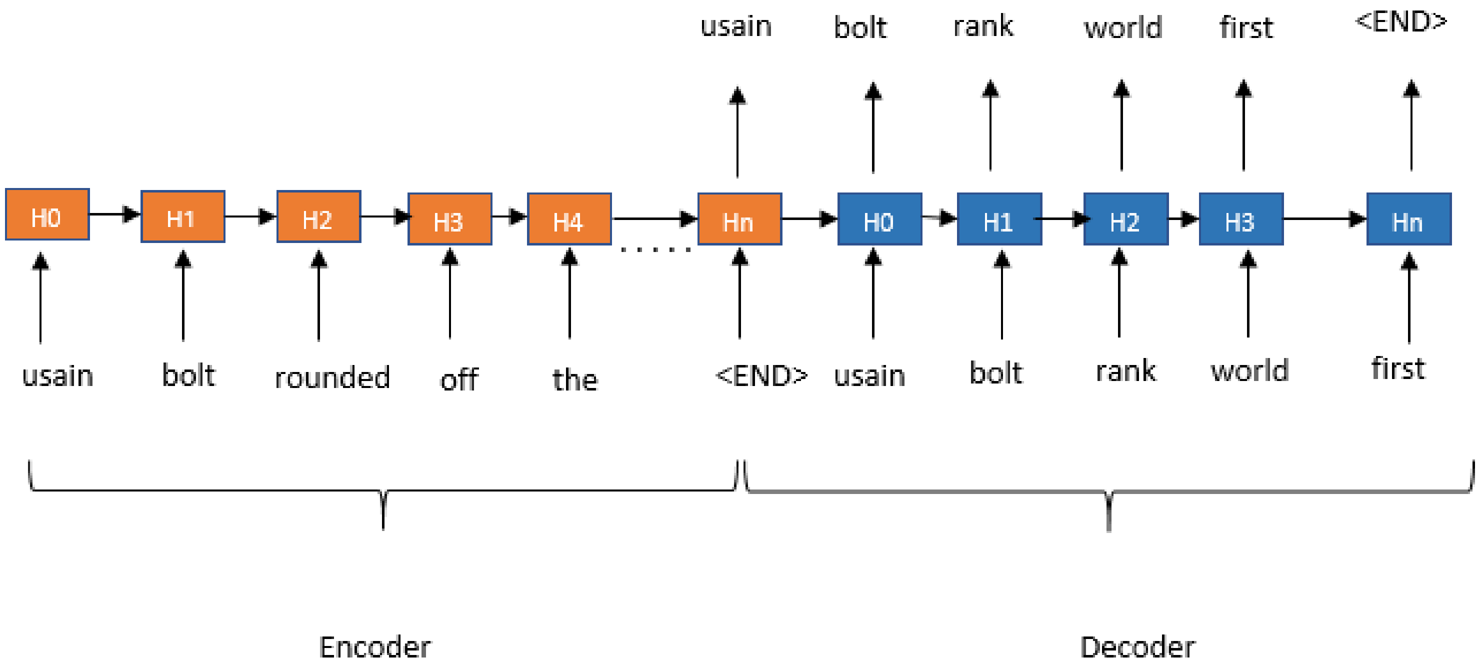
Evaluated on 1000 randomly sampled instances from SQuAD dataset

- Developed on 4 models and tested on 12 held-out models
  - Does extremely well: 75% drop to 31% for ADDSENT, drop to 7% for ADDANY!
- How well do the variants perform?
  - ADDONESENT similar to ADDSENT, although it is model independent!
  - ADDCOMMON drops score to 46%
- Humans are not fooled (mostly) by adversarial examples
  - Adversarial sentences do not contradict the information in the passage for the true answer

Evaluation

- Does the model take the adversarial bait?
    - 96.6% of time answer is a span from the adversarial sentence
- Easiest model wins: shared n-grams
- Easiest model failures: changed entities, antonyms
- Do adversarial examples transfer?
    - ADDANY does not
    - ADDSENT does
    - This is similar to vision models: one has to deploy a definite strategy to fool the models

Seq2Sick: Adversarial Examples for Seq2Seq



Have: Semantics altering/eliminating perturbations

Want: small changes → BIG effects

Want: Targeted attack

Infinite discrete space to optimise over!

Targeted attack is extremely hard!

Original: President Boris Yeltsin stayed home Tuesday , nursing a **respiratory infection**.
Summary: Yeltsin stays home after **illness**

Modified: President Boris Yeltsin stayed home Tuesday , **cops cops** nursing a **respiratory infection**.
Summary: Yeltsin stays home after **police arrest**

Problem Formulation

$$\min_{\delta} L(X + \delta) + \lambda \cdot R(\delta)$$

Loss function for targeted attack

Regularisation

Optimisation over discrete space

Relax the wants!

Non-overlapping attack

Floods on Yangtze river continue → Flooding in water recedes in river

Targeted keyword attack

Yeltsin stays home after illness → Yeltsin stays home after police arrest

Loss Functions

Idea: encode objective directly

Given sequence $S = s_1, s_2, \dots s_N$

Non-overlapping condition: $z_t^{(s_t)} < \min_{w \in W} z_t^{(w)} \quad \forall\, 1 \le t \le N$

$$L_{\text{non-overlapping}} = \sum_{t=1}^{M} \max\{-\epsilon,\ z_t^{(s_t)} - \max_{y \neq s_t}\{z_t^{(y)}\}\}$$

Can be encoded as a hinge-like loss!

Similarly for targeted keyword condition

$$L_{\text{keywords}} = \sum_{i=1}^{|K|} \min_{t \in [M]} \{\max\{-\epsilon, \max_{y \neq k_i}\{z_t^{(y)}\} - z_t^{(k_i)}\}\}$$

What happens when keywords compete?

Use mask to block off solved word positions!

Regularisation

l_2 distance is bad

- If the gradient on a word is non-zero (always happens) it will be changed
- Result: adversarial sequence completely different from input
- Hard to obtain convergence

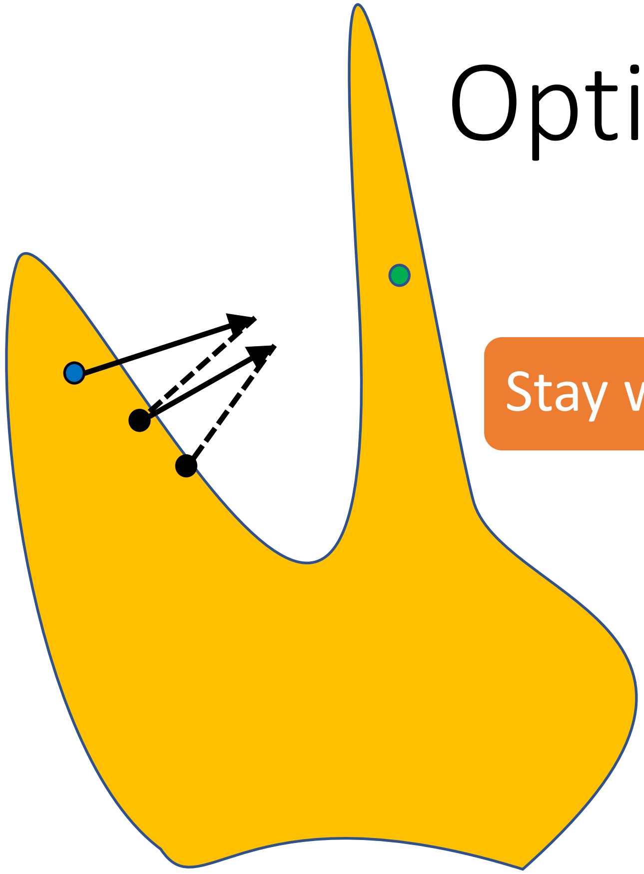
$$R(\delta) = \|\delta\|_2$$

Fix: enforce that most words have to remain the same

- Design the metric to aggregate distances over each word position
- Mathematically, lasso over word positions!

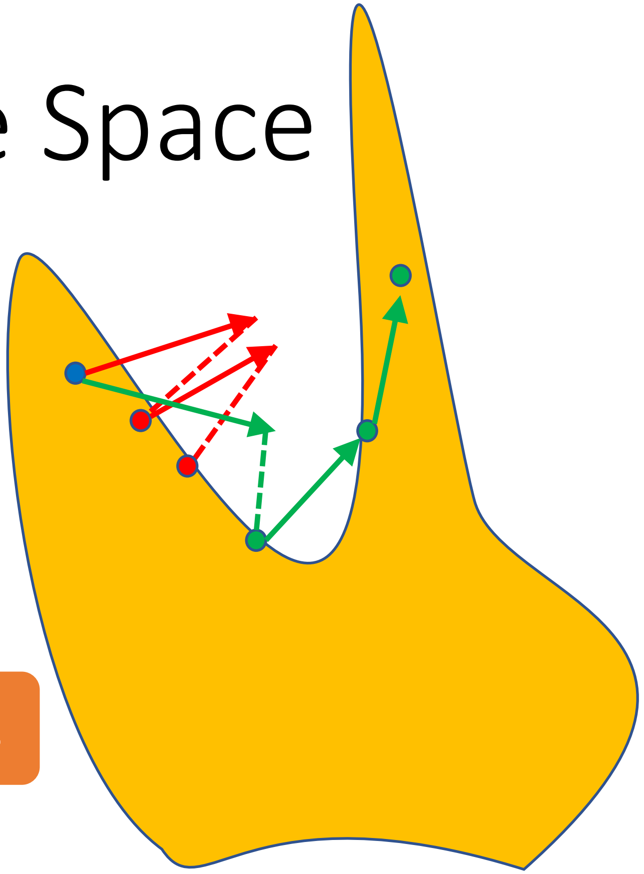
$$R(\delta) = \sum_{i=1}^{|\delta|} \|\delta_i\|_2$$

Optimising over Discrete Space



Stay within vocabulary!

Don’t go to lonely corners!



Projected Gradient Descent

$$\min_{\delta} L(X + \delta) + \lambda \cdot R(\delta)$$

s.t $x_i + \delta_i \in W \ \forall i = 1, 2, \dots |\delta|$

Gradient Regularisation

$$\sum_{i=1}^{N} \min_{w \in W} \|x_i + \delta_i - w\|_2$$

Evaluation

Evaluated on Text Summarisation (TS) and Machine Translation (MT)

- Changing just 2-3 words is extremely effective on TS
- 80-100% accuracy on any task in either setting
- MT adversaries give meaningless outputs
  - But *very close* to grammatically correct (anecdotal)

Dataset	Success%	BLEU	# changed
Gigaword	86.0%	0.828	2.17
DUC2003	85.2%	0.774	2.90
DUC2004	84.2%	0.816	2.50

Datasest	$\lvert K \rvert$	Success%	BLEU	# changed
Gigaword	1	99.8%	0.801	2.04
	2	96.5%	0.523	4.96
	3	43.0%	0.413	8.86
DUC2003	1	99.6%	0.782	2.25
	2	87.6%	0.457	5.57
	3	38.3%	0.376	9.35
DUC2004	1	99.6%	0.773	2.21
	2	87.8%	0.421	5.1
	3	37.4%	0.340	9.3

Method	Success%	BLEU	# changed
Non-overlap	89.4%	0.349	3.5
1-keyword	100.0%	0.705	1.8
2-keyword	91.0 %	0.303	4.0
3-keyword	69.6%	0.205	5.3

Evaluation

Are adversarial inputs <u>syntactically</u> similar to the original input?

Simple check: evaluate perplexity w.r.t the model

	DUC2003	DUC2004
Original	102.02	121.09
Non-overlap	114.02	149.15
1-keyword	159.54	199.01
2-keyword	352.12	384.80

Are adversarial inputs <u>semantically</u> similar to the original input?

Simple test: check if sentiments are preserved

Result: Only 2.2% not preserved!

Concrete adversarial examples in paper!

Conclusion

- Designing adversarial examples to NLP systems is hard
  - Discrete space
  - Small perturbations can change semantics
    - Can you tell apart a good adversarial example from a bad one?
  - Therefore, NLP systems are more robust, by and large!
    - Does this mean that we have achieved good NLP systems?
- But there are methods to get around it
  - Black box methods using ‘behavioural tests’
  - Gradient-based white box methods
  - Human evaluations are important!
- Is it easy to construct adversarial examples for NLP models?
  - Can be done even with random perturbation (unlike vision)
- Do adversarial examples transfer for NLP models?
  - Randomly generated ones do not (just like in vision)
  - But examples optimized w.r.t a particular model transfer much better