# Model Debugging

- You’ve implemented a nice model (or replicated a SOTA model)
- Your accuracy on the test set is bad
- What do I do?
    - Training/Test stage

# Another Typical Situation

- You’ve implemented a nice model (or replicated a SOTA model)
- Your accuracy on the test set is good
- You want to know what your model is not good at?

# Model Diagnostic

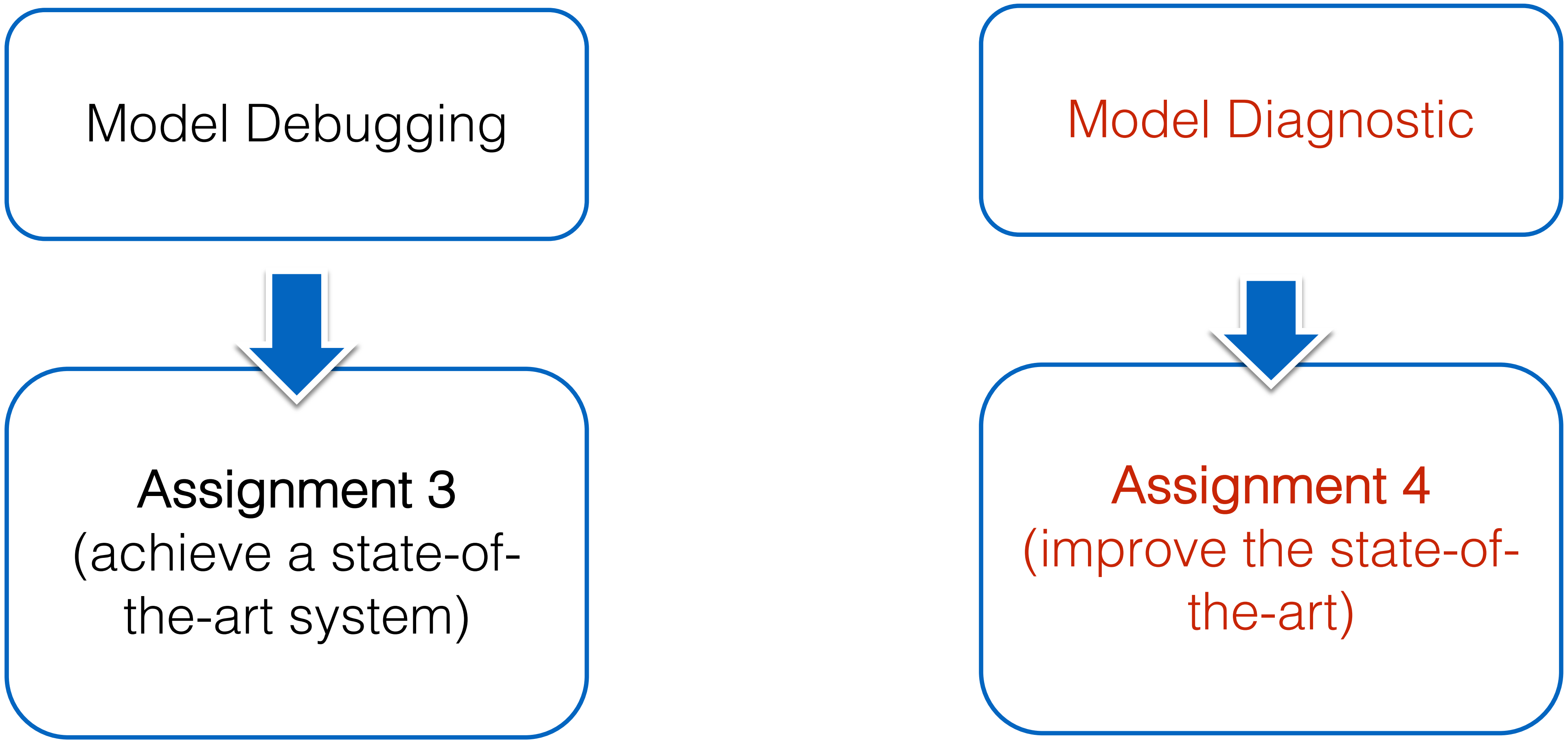
- What is “Model Diagnostic”?
  - Identify the weaknesses (strengths) of your models
- Why do we need “Model Diagnostic”?
  - What Works? (Interpretability)
  - What’s Next? (Next step)

# Model Diagnostic

How to further improve the performance?


Performance of many NLP tasks (i.e. NER) has **reached a plateau.**

# More Intuitively



# How to achieve this goal?

- Error Analysis
- Diagnostic Evaluation
- Interpretable Evaluation

# How to achieve this goal?

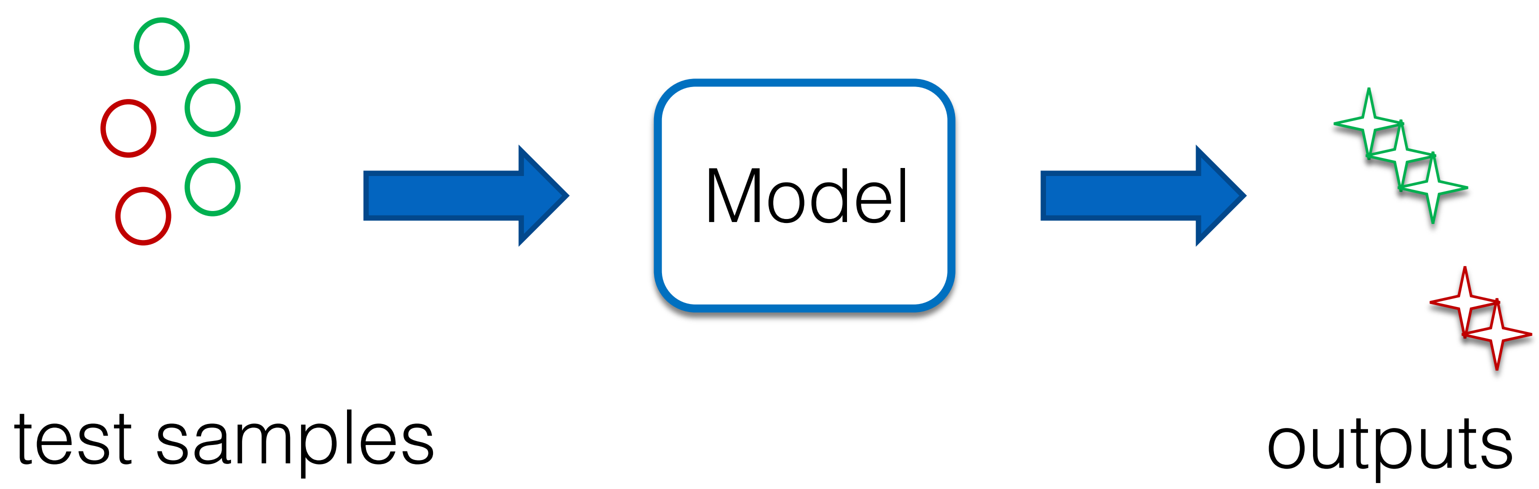
- Error Analysis  (four must-read papers)
- Diagnostic Evaluation (four must-read papers)
- Interpretable Evaluation (two must-read papers)

| Year | Conf. | Citation | Title |
|---|---|---|---|
| 2015 | arXiv | 916 | Visualizing and understanding recurrent networks<br>Andrej Karpathy, Justin Johnson, Li Fei-Fei |
| 2011 | CICLing | 498 | Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?<br>Christopher D. Manning |
| 2016 | ACL | 458 | A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task<br>Danqi Chen, Jason Bolton, Christopher D. Manning |
| 2012 | EMNLP | 99 | Parser Showdown at the Wall Street Corral: An Empirical Investigation of Error Ty<br>Jonathan K. Kummerfeld, David Hall, James R. Curran, Dan Klein |

# Error Analysis

- **Manually** check test cases on which models make a wrong prediction (or unreasonable generation)

- Try to abstract **commonalities** of these error cases



# Error Analysis on Sentiment Classification Task

- The classifier will fail when …
  - Err-I: sentences with double negation
    - I **don't** think this movie is **not** interesting
  - Err-II: sentences with subjunctive mood
    - The movie **could have** been better.
  - Err-III: sentences with annotation errors
  - I like this movie -> negative

Long-term Dependency

# Error Analysis on Sentiment Classification Task

- The classifier will fail when …
  - Err-I: sentences with double negation
    - I **don't** think this movie is **not** interesting
  - Err-II: sentences with subjunctive mood
    - The movie **could have** been better.
  - Err-III: sentences with annotation errors
    - I like this movie -> negative

Reasoning

# Error Analysis on Sentiment Classification Task

- The classifier will fail when …
  - Err-I: sentences with double negation
    - I **don't** think this movie is **not** interesting
  - Err-II: sentences with subjunctive mood
    - The movie **could have** been better.
  - Err-III: sentences with annotation errors
    - I like this movie -> negative

De-noising

# In Summary

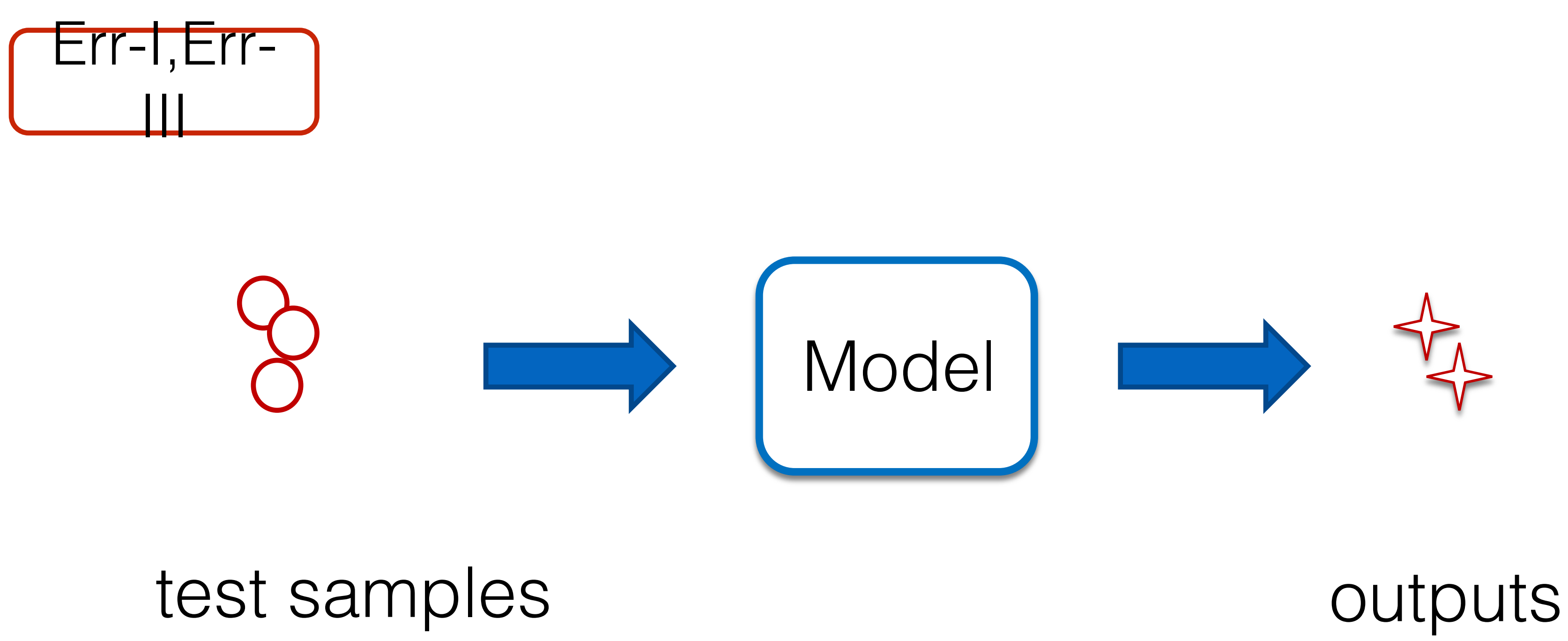
- Naïve but super useful method
- Learning to perform error analysis is a good research habit
  - Many solid ideas come from error analysis
- Improve yourself by error analysis
  - Zero-distance with the data, get more domain knowledge

# Blind Spots of Error Analysis

- Err-I: sentences with double negation
- Err-II: sentence with subjunctive mood
- Err-III: sentence with annotation errors

# Blind Spots of Error Analysis

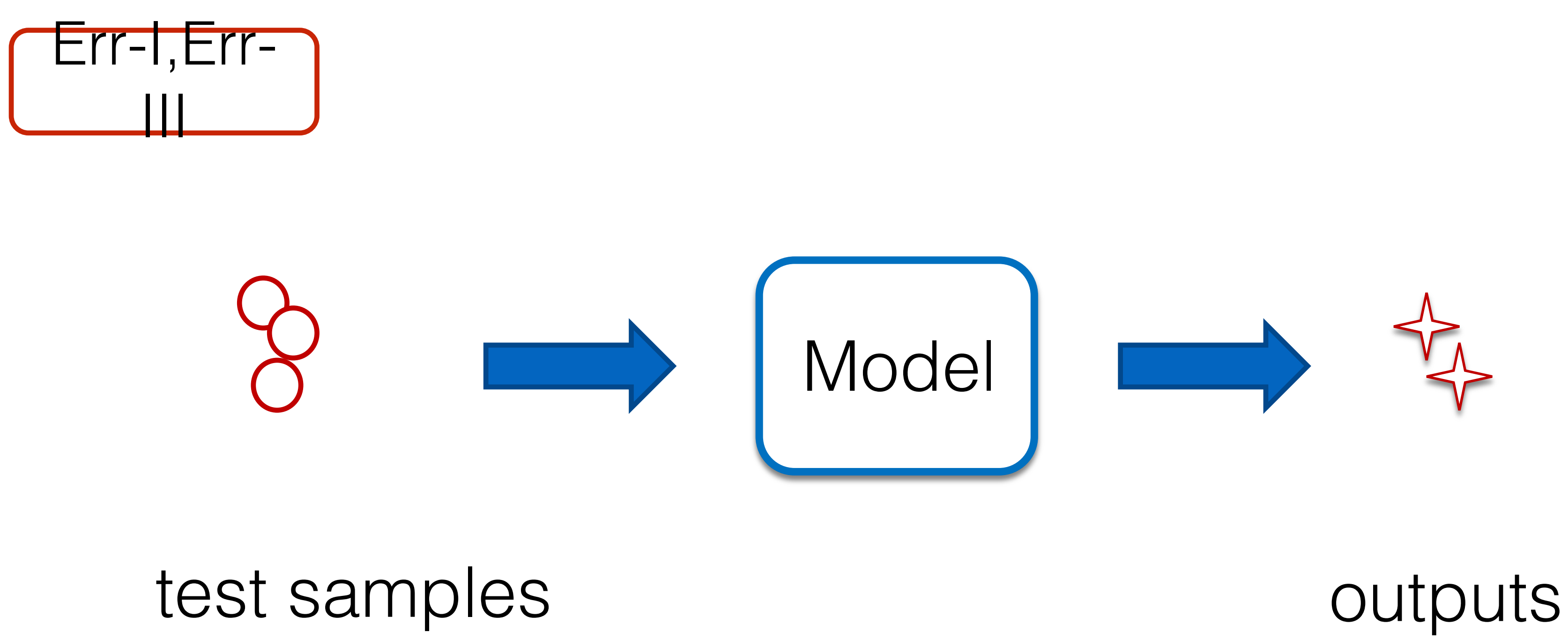
What if there is no Err-II samples in the test set



# Blind Spots of Error Analysis

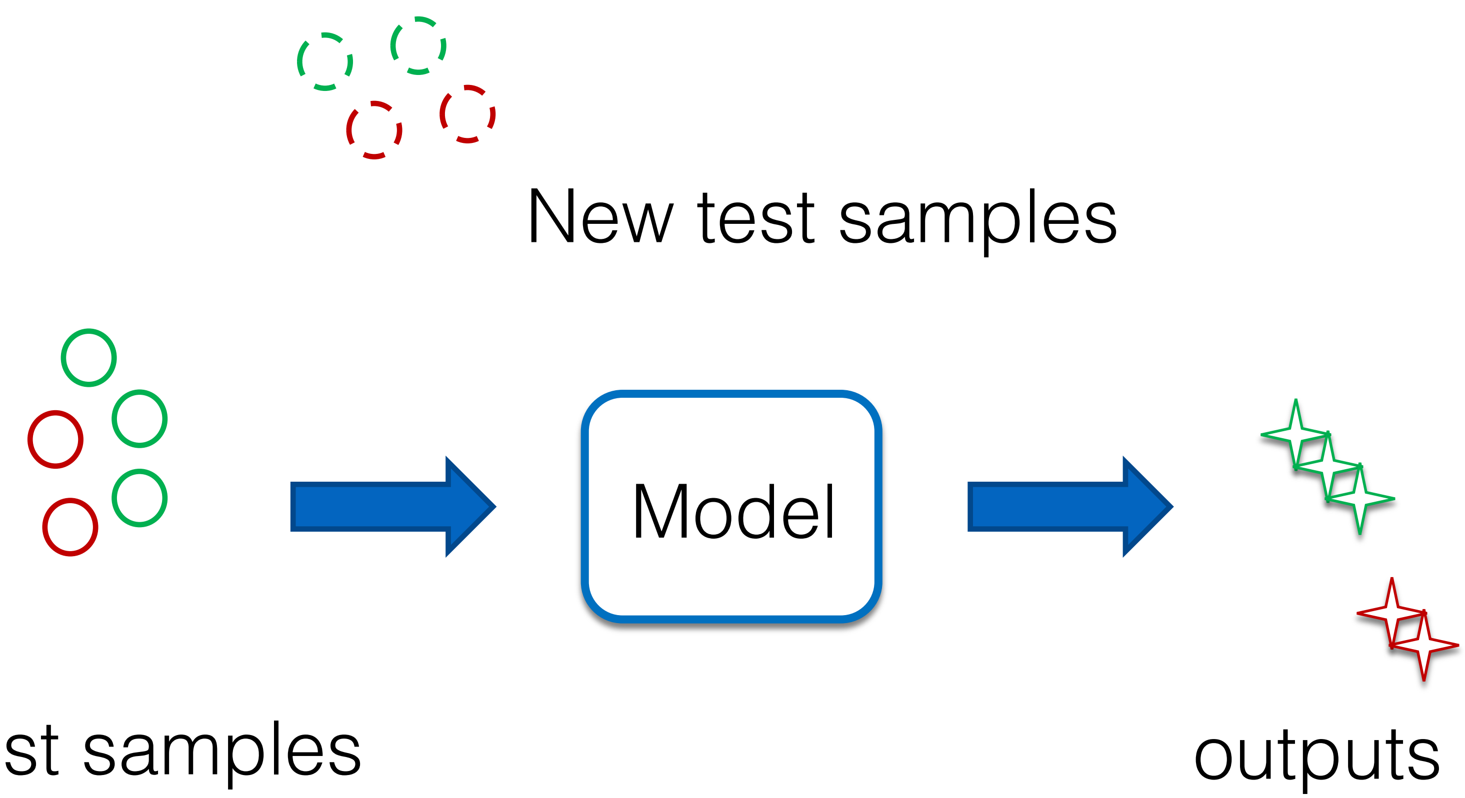
What if there is no Err-II samples in the test set

Construct!



# Diagnostic Evaluation

- **Automatically** construct a **new** set of test samples that current models will fail
- Re-evaluate models using the newly-constructed data

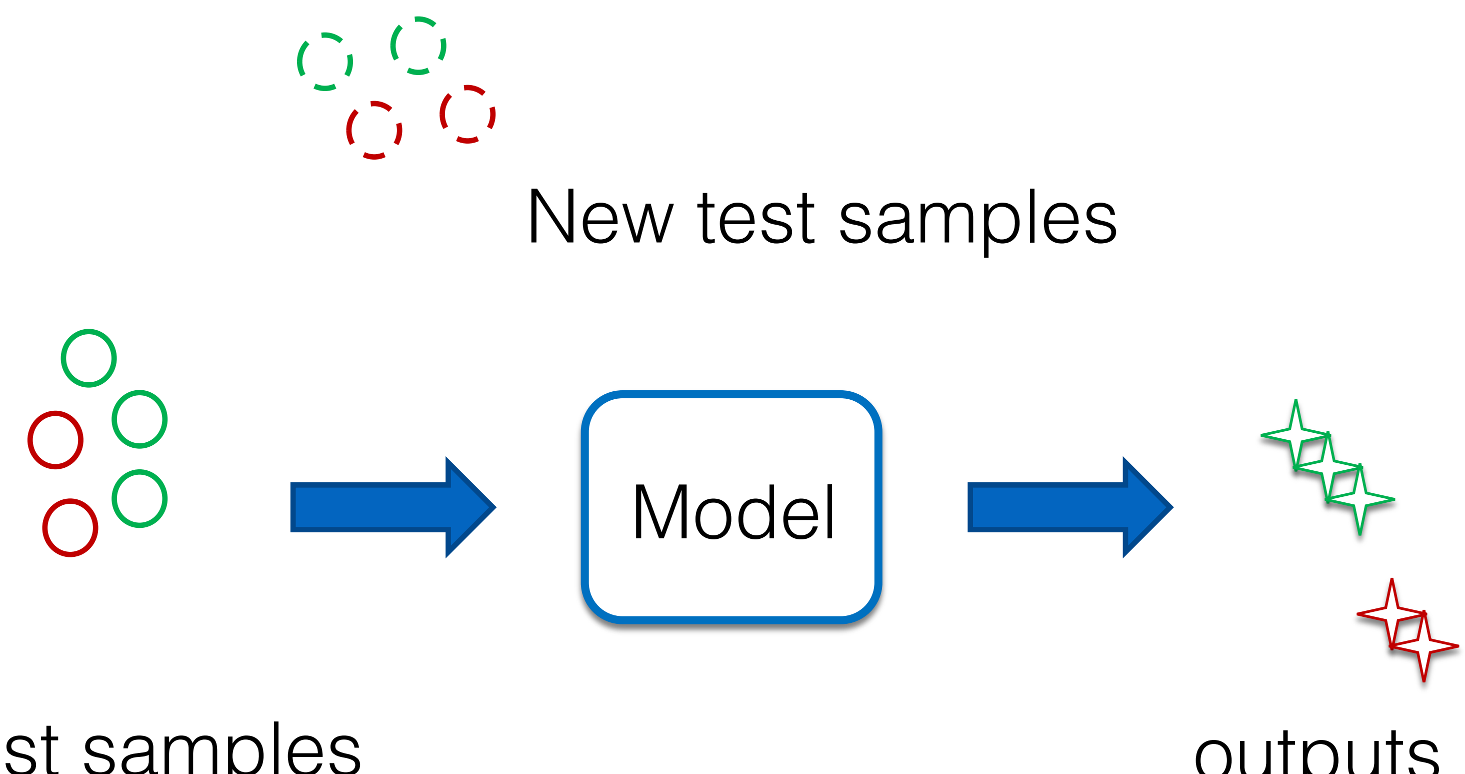


# Diagnostic Evaluation

Stress set
Contrastive set
Adversarial set
...

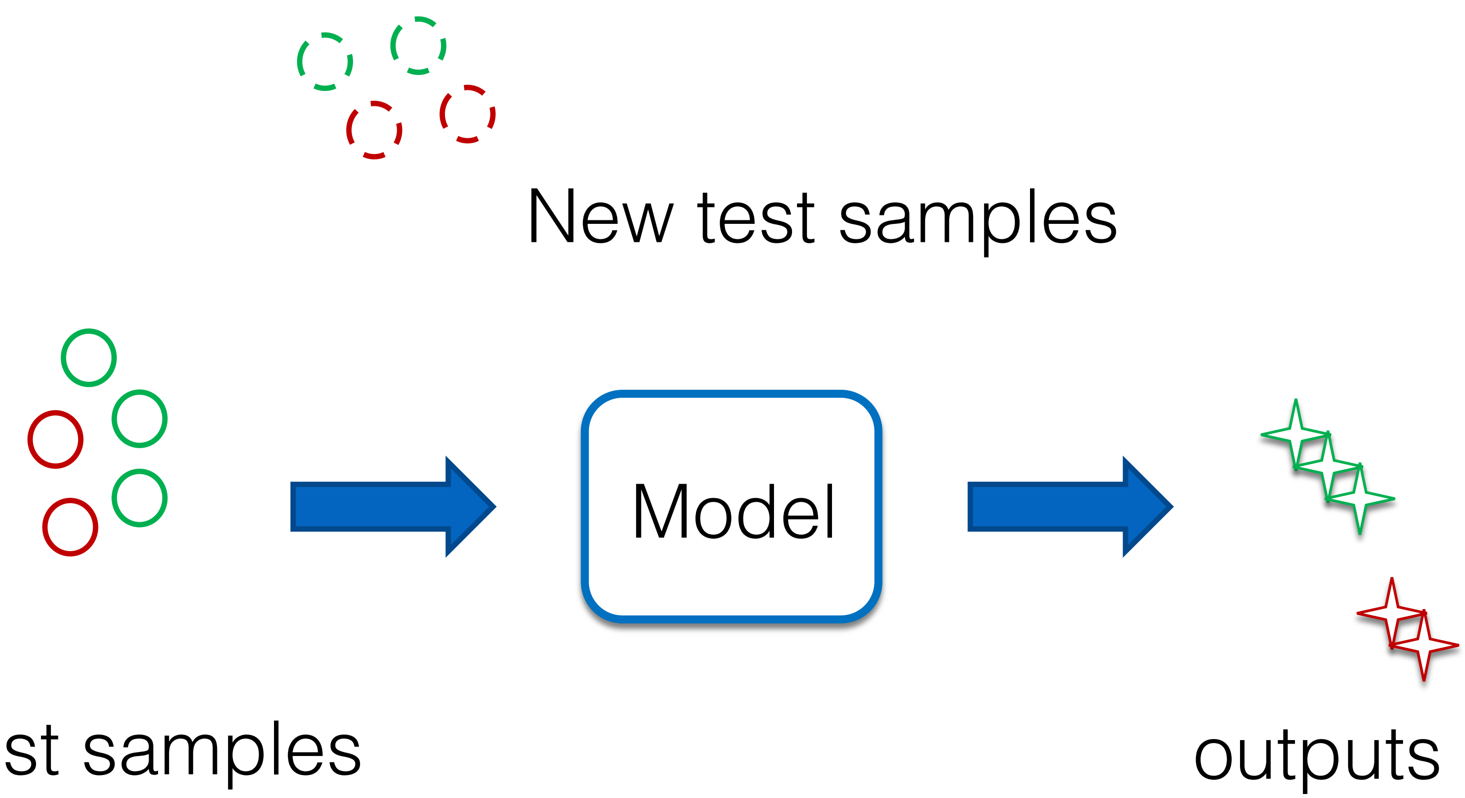
- **Automatically** construct a new set of test samples that current models will fail

- Re-evaluate models using the newly-constructed data



# Confirmation bias in Diagnostic Evaluation

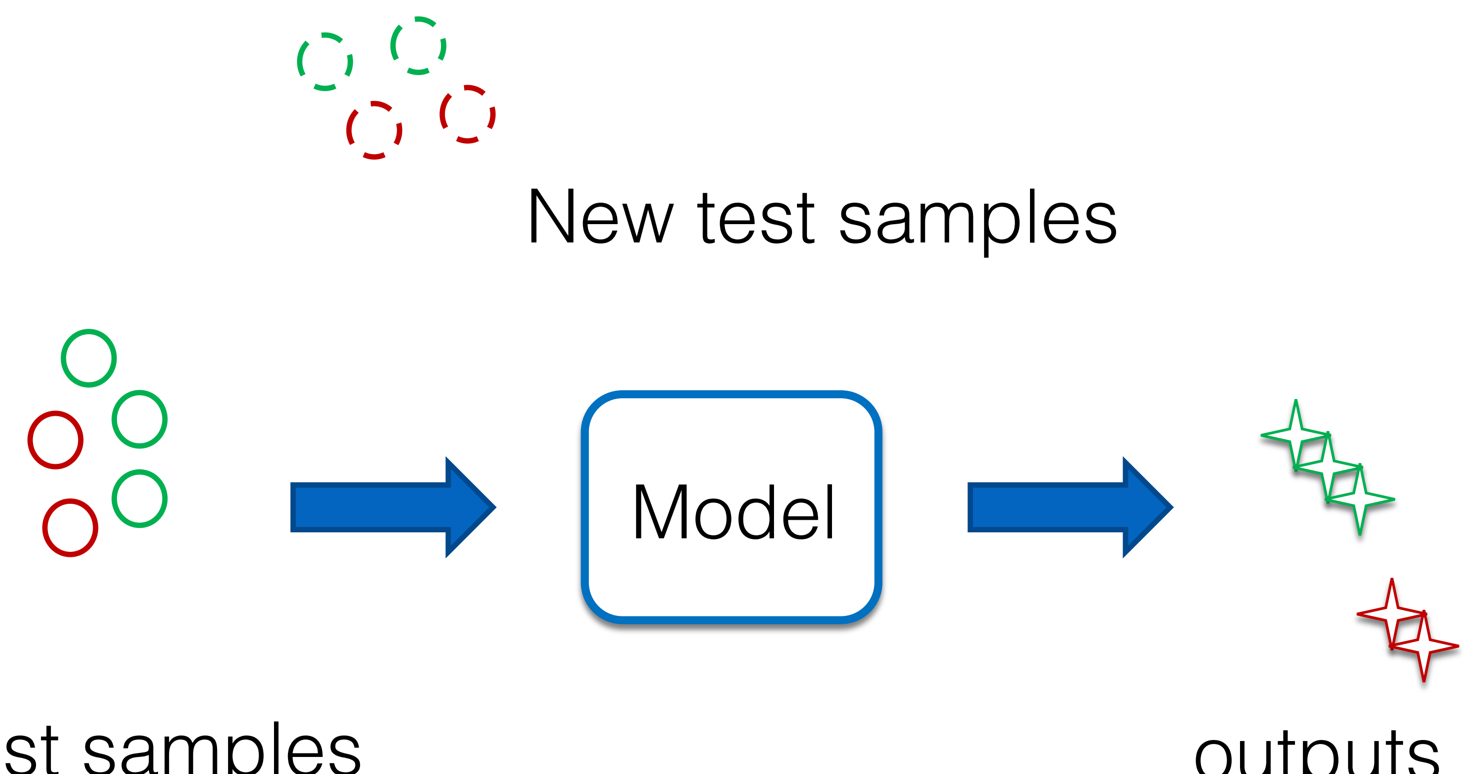
How do we know what types of samples to be constructed?



# Confirmation bias in Diagnostic Evaluation

How do we know what types of samples to be constructed?

Assume that our model will struggle at samples with some patters

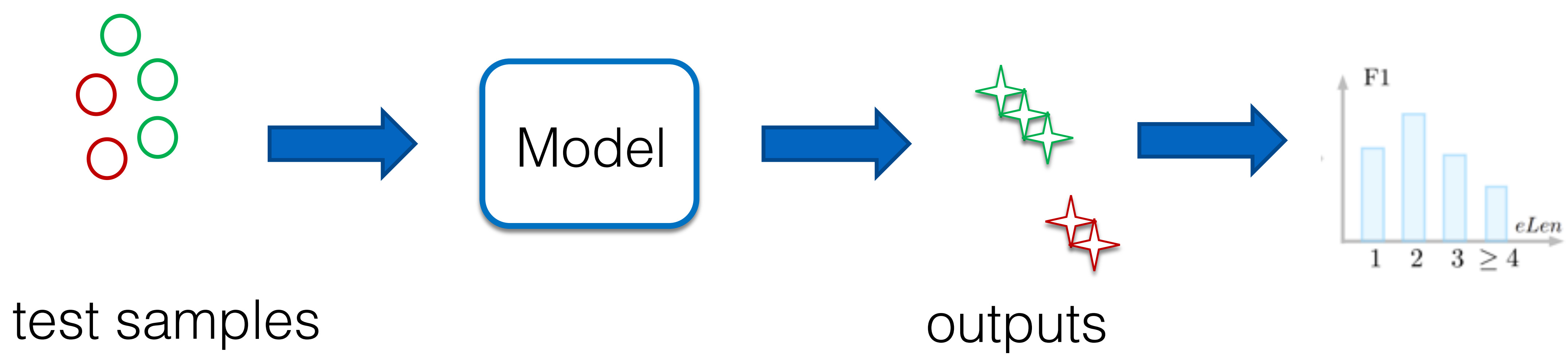


# Interpretable Evaluation

- Motivation: a good evaluation metric can
  - not only rank different systems
  - but also tell their *relative advantages* (*<u>strengths and weaknesses</u>*) of them.

# How to achieve it?

- One sentence to summarize
  - By partitioning the performance of **test set** into different interpretable groups based on a pre-defined attribute



# How to achieve it?

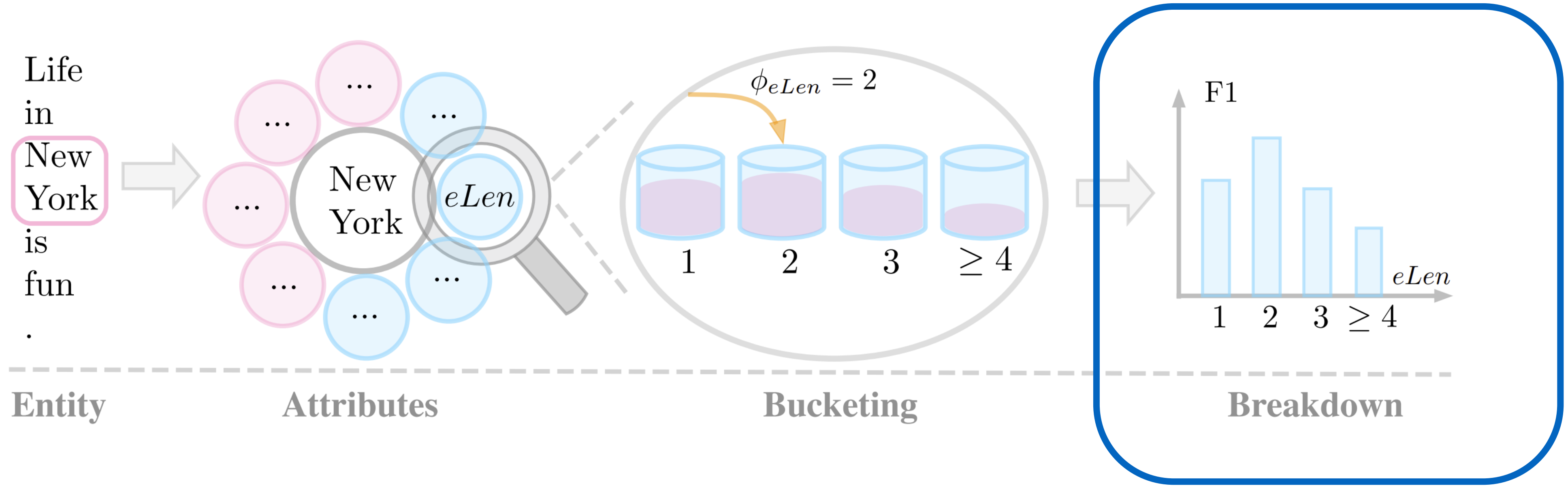
- One sentence to summarize
  - By partitioning the performance of test set into different interpretable groups based on a pre-defined attribute

- Define Attributes
- Partition Test Samples
- Breakdown Performance

# Methodology

- Define attributes (e.g., entity length: *eLen*)
- Partition test samples
- Breakdown performance

Performance Histogram

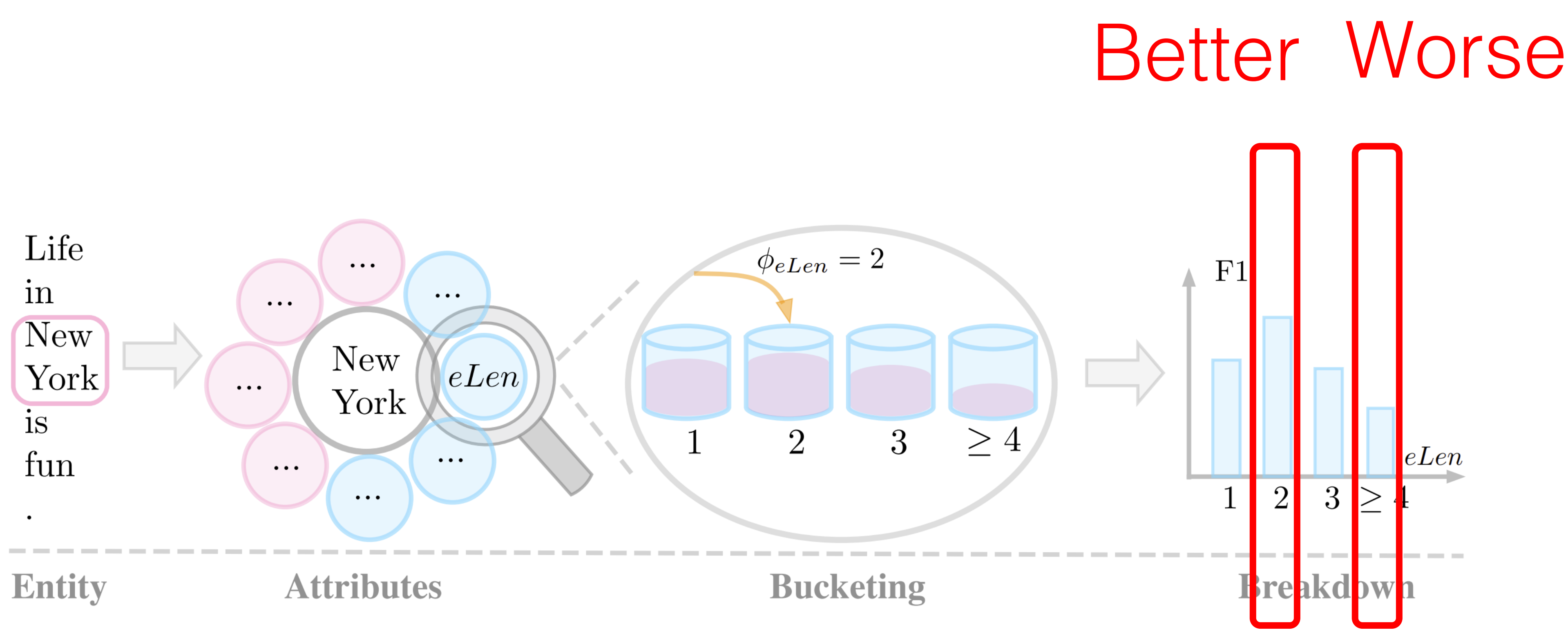


# Attributes

- Different tasks could have different attributes
- Token-level, span-level, sentence-level
  - Token-level: part-of-speech tag
  - Span-level: span length
  - Sentence-level: sentence length

# Performance Histogram

- *Diagnostic for single system*

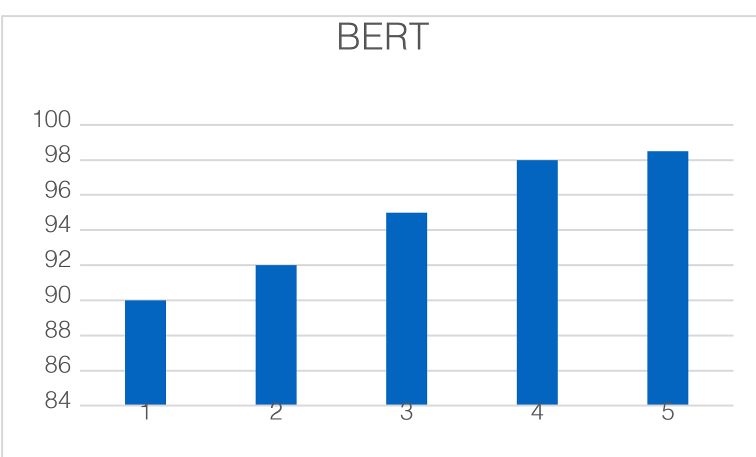


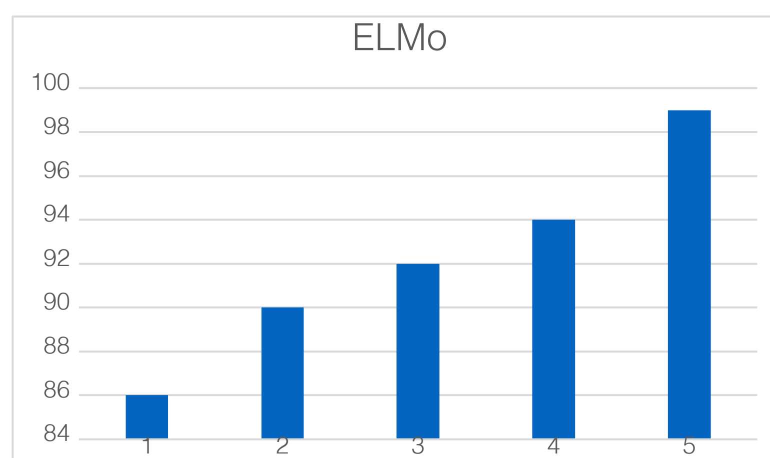
# Performance Histogram

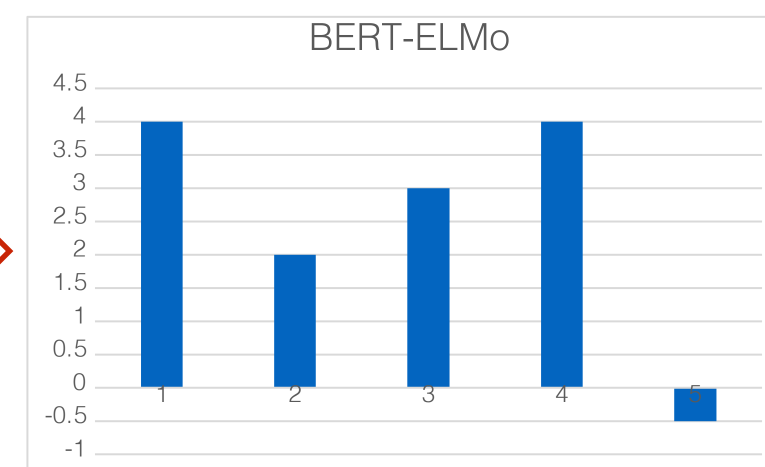
- *Diagnostic for two systems*

BERT v.s. ELMo

Performance Gap Histogram



# In Summary

- No need to construct new samples
- No need to think about potential error types
- But… need “attributes”

# Model Diagnostic: Comparison

| Methodology | Stage | Human effort | Additional test set |
|---|---|---|---|
| Error Analysis | test | ⭐⭐⭐ | × |
| Diagnostic Evaluation | test | ⭐⭐ | √ |
| Interpretable Evaluation | test | ⭐ | × |

# Can we automate System Diagnostic?

- Require human efforts (more or less)
- Task-dependent

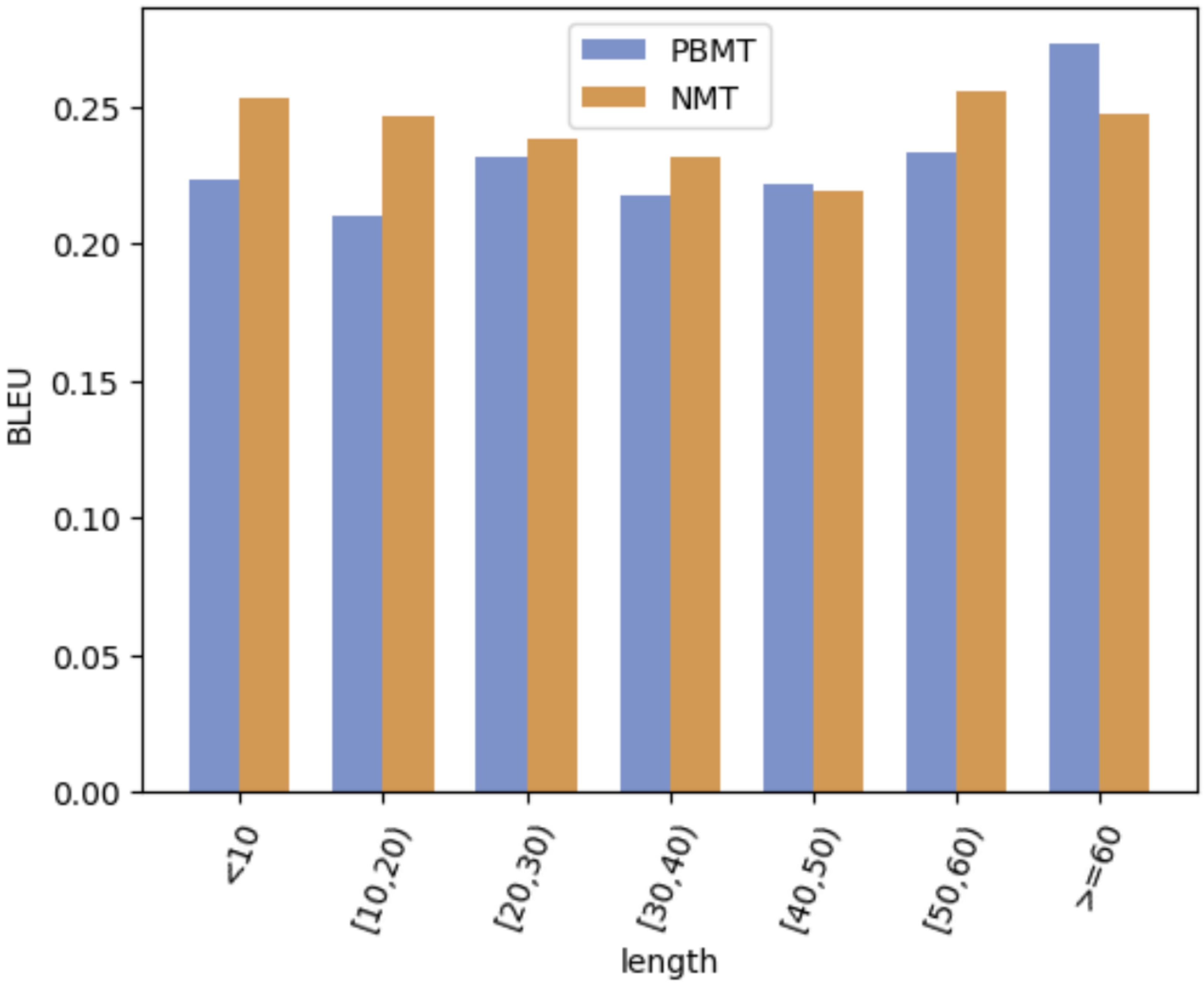
# Can we automate System Diagnostic?

| Methodology | Stage | Human effort | Additional test set |
|---|---|---|---|
| Error Analysis | test | ⭐⭐⭐ | × |
| Diagnostic Evaluation | test | ⭐⭐ | √ |
| | | | |
| Interpretable Evaluation | test | ⭐ | × |

# *Compare-mt*

- A diagnostic analysis toolkit for *machine translation*
- Calculates **aggregate statistics about accuracy of particular types of words or sentences**, finds salient test examples
- An example of this for quantitative analysis of language generation results (https://github.com/neulab/compare-mt)

# PBMT v.s. NMT




Tips: **phrase-based** machine translation and **neural network-based** machine translation systems are two major paradigms over the past 20 years.

# *ExplainaBoard*

- Next Generation of Leaderboard
    - *Track NLP progress*
    - *Help researchers diagnose NLP systems*

# LeaderBoard v.s. ExplainaBoard

Other models · Models with highest F1

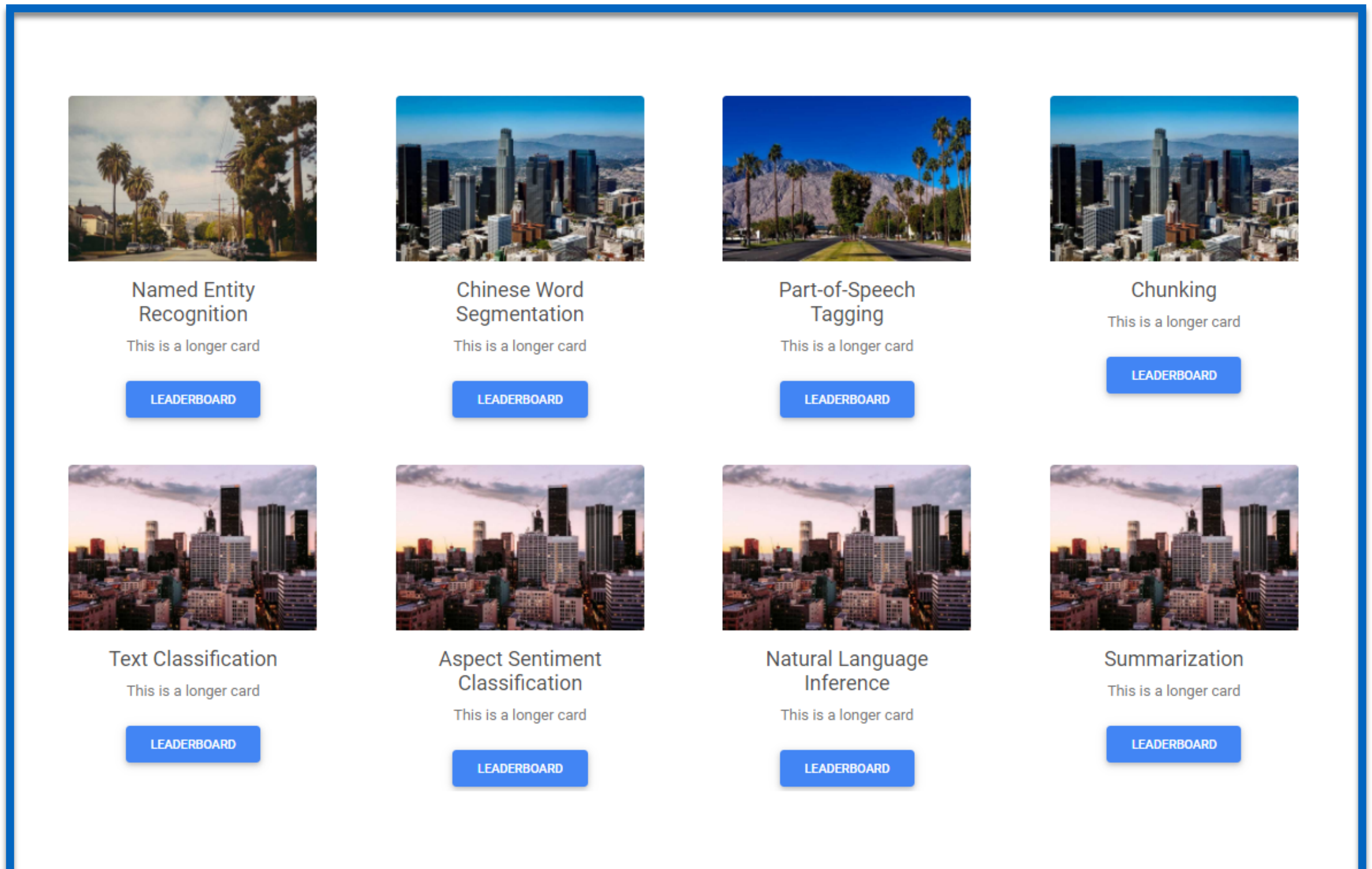
View F1 All models Edit

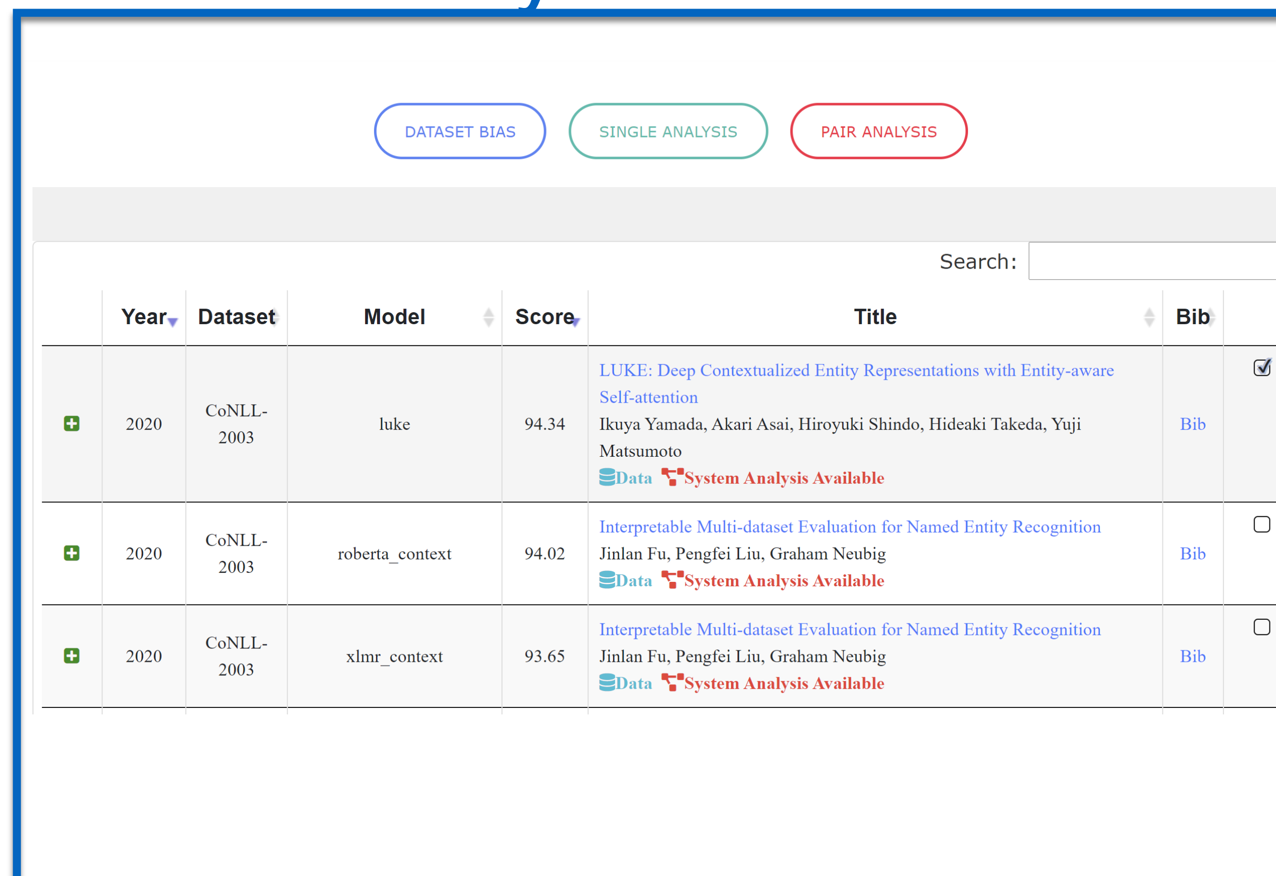
| RANK | MODEL | F1 | EXTRA TRAINING DATA | PAPER | CODE | RESULT | YEAR |
|---|---|---|---|---|---|---|---|
| 1 | LUKE | 94.3 | × | LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention | | | 2020 |
| 2 | ACE + document-context | 94.14 | × | Automated Concatenation of Embeddings for Structured Prediction | | | 2020 |
| 3 | Cross-sentence context (First) | 93.74 | × | Exploring Cross-sentence Contexts for Named Entity Recognition with BERT | | | 2020 |
| 4 | ACE | 93.64 | × | Automated Concatenation of Embeddings for Structured Prediction | | | 2020 |
| 5 | CNN Large + fine-tune | 93.5 | ✓ | Cloze-driven Pretraining of Self-attention Networks | | | 2019 |
| 6 | Biaffine-NER | 93.5 | × | Named Entity Recognition as Dependency Parsing | | | 2020 |
| 7 | GCDT + BERT-L | 93.47 | ✓ | GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling | | | 2019 |
| 8 | I-DARTS + Flair | 93.47 | ✓ | Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition | | | 2019 |
| 9 | CrossWeigh + Pooled Flair | 93.43 | × | CrossWeigh: Training Named Entity Tagger from Imperfect Annotations | | | 2019 |
| 10 | LSTM-CRF+ELMo+BERT+Flair | 93.38 | ✓ | Neural Architectures for Nested NER through Linearization | | | 2019 |

# LeaderBoard v.s. ExplainaBoard

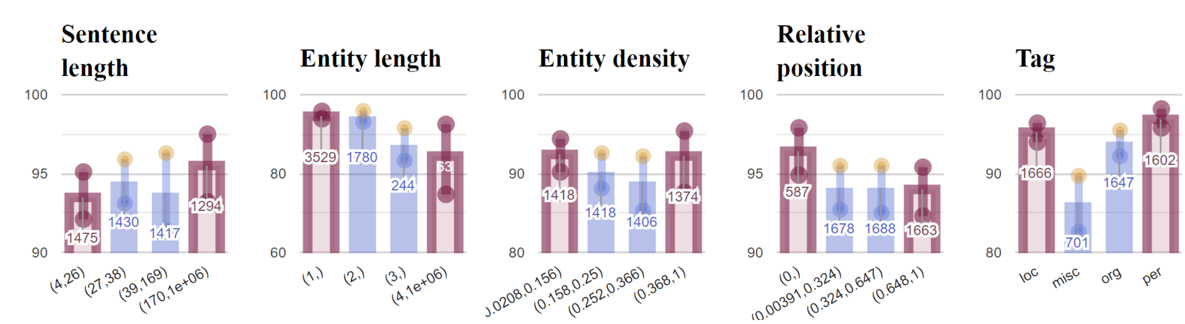
Multiple Tasks

Leaderboard and Analysis Buttons




Interpretable Evaluation Results



# *ExplainaBoard*

- Cover more tasks
- More functionalities
  - Interpretability: Single system diagnosis
  - Interactivity: System pair diagnosis
  - Reliability: confidence interval, calibration value
- Github: https://github.com/neulab/InterpretEval