

Adapting Discriminative Reranking to Grounded Language Learning

- Joohyun Kim ,Raymond J.Mooney

Review By Sugam Anand

What is Grounded Language Learning ?

- It involves learning language by simply observing its usage in context in which the meaning of the sentence in natural language is grounded in perception or context.

“ Rahul dravid ducks as **ball** passes over his head”



“ ball touched the pad first”



Background

- Borschinger et al. (2011) reduced the task of grounded language learning to unsupervised induction of probabilistic context-free grammar (PCFG) and applied it to the task of sportcasting in simulated robot soccer games.
- Kim and Mooney applied this approach to the complex problem of learning to follow navigational instructions.
- Effective approach to improve performance of generative models with secondary discriminative model
- Authors introduced the novel idea of applying discriminative reranking to this problem. Reranking has been widely applied to different NLP tasks like NER, semantic parsing, syntactic parsing etc
- Conventional discriminative reranking techniques cannot be directly applied to navigation task. So paper describes how did they achieve this task.

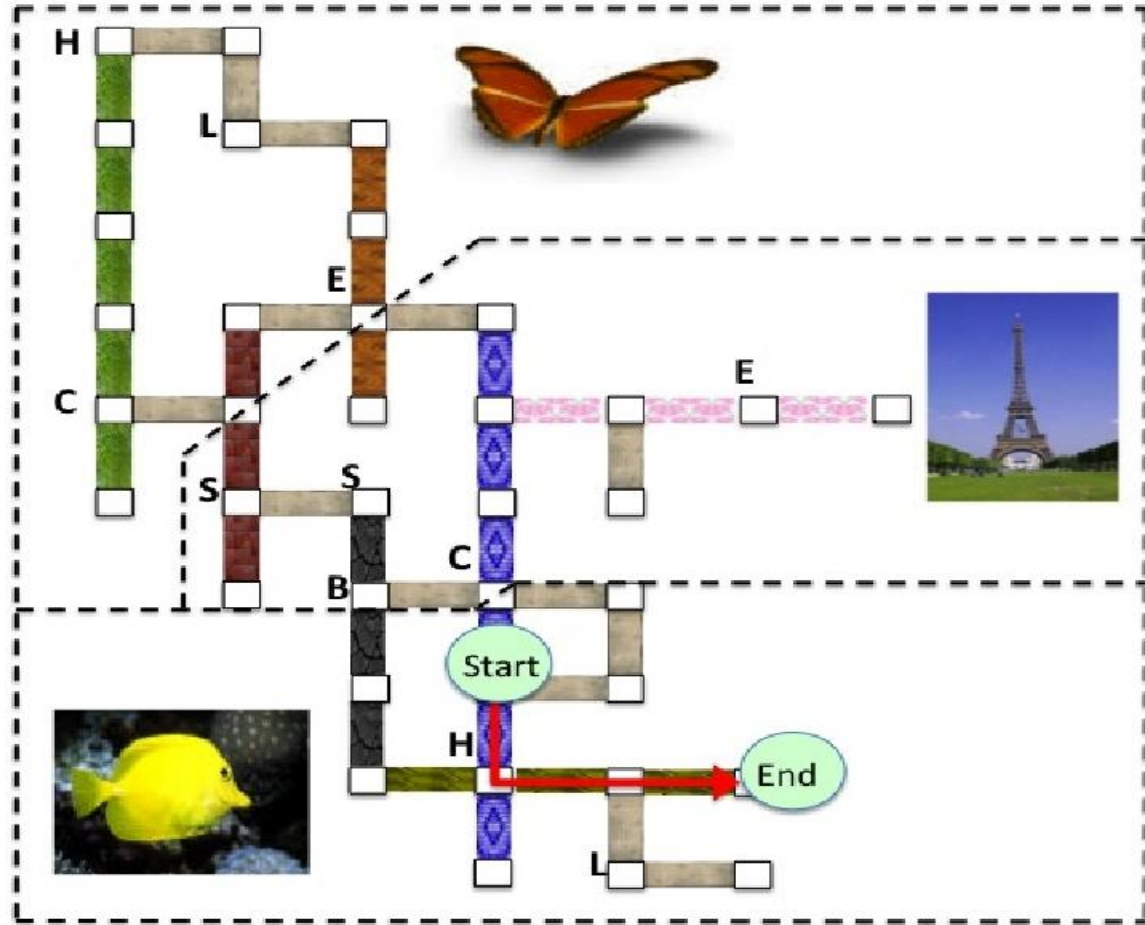
Task ...

- Learning to follow navigational instructions in natural language.

Environment

There are three virtual indoor environments in total. Each environment consists of interconnecting hallways with objects placed at various intersections.

There are several different floor patterns as well as wall painting which were used in conjunction with the objects for giving directions.

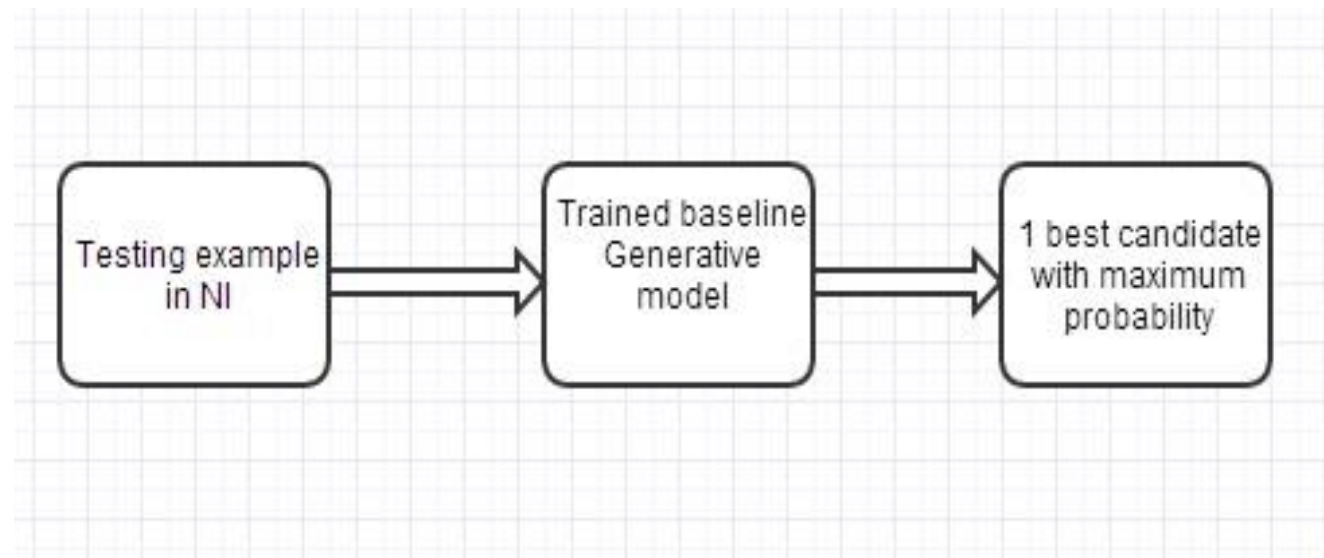


Challenge

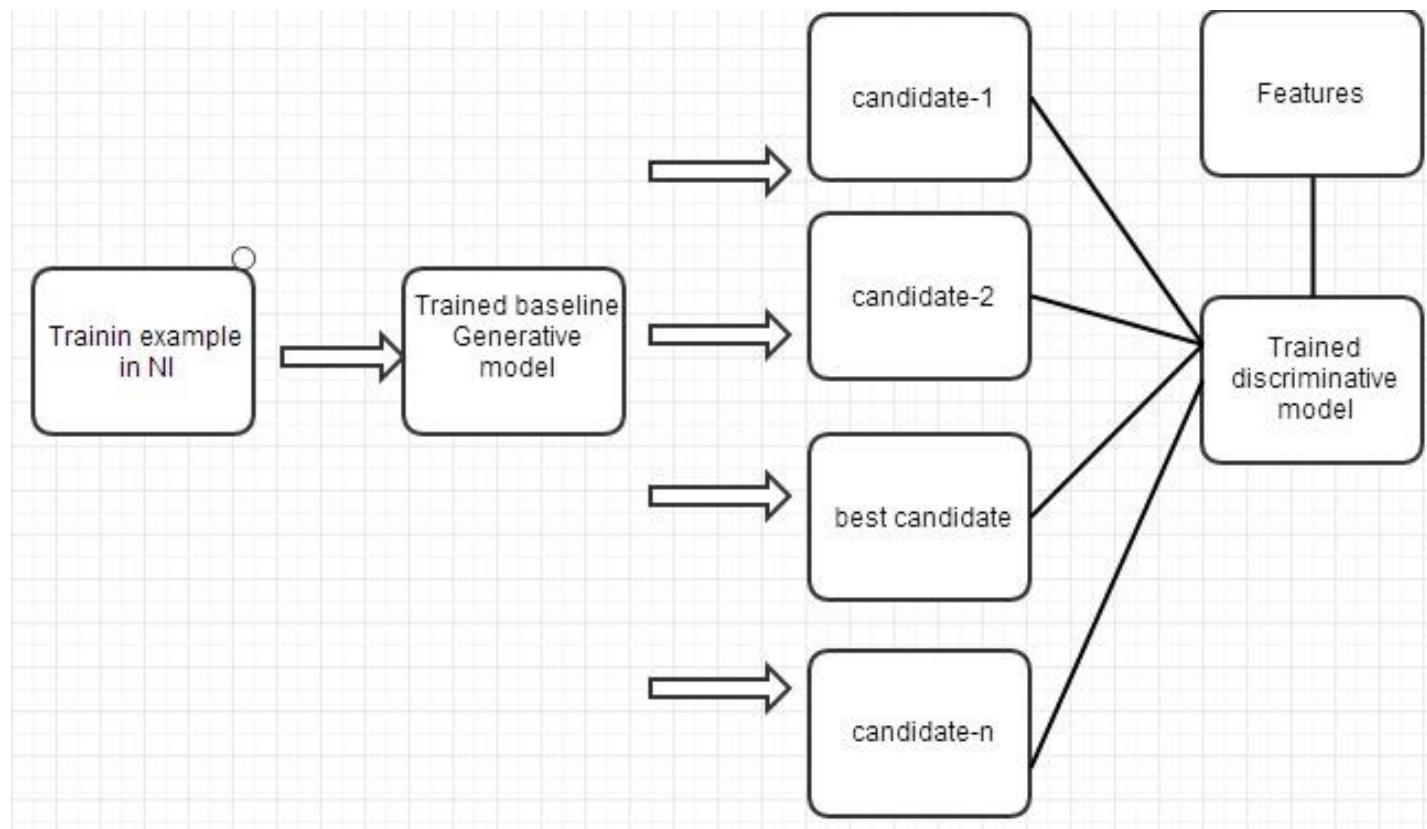
Instruction: "at the easel, go left and then take a right onto the blue path at the corner"

Landmarks plan: **Travel** (steps: 1) ,
Verify (**at: EASEL** , side: CONCRETE HALLWAY) ,
Turn (LEFT) ,
Verify (front: CONCRETE HALLWAY) ,
Travel (steps: 1) ,
Verify (side: BLUE HALLWAY , **front: WALL**) ,
Turn (RIGHT) ,
Verify (back: WALL , **front: BLUE HALLWAY** , front: CHAIR ,
front: HATRACK , left: WALL , right: EASEL)

How it works?



Discriminative Reranking

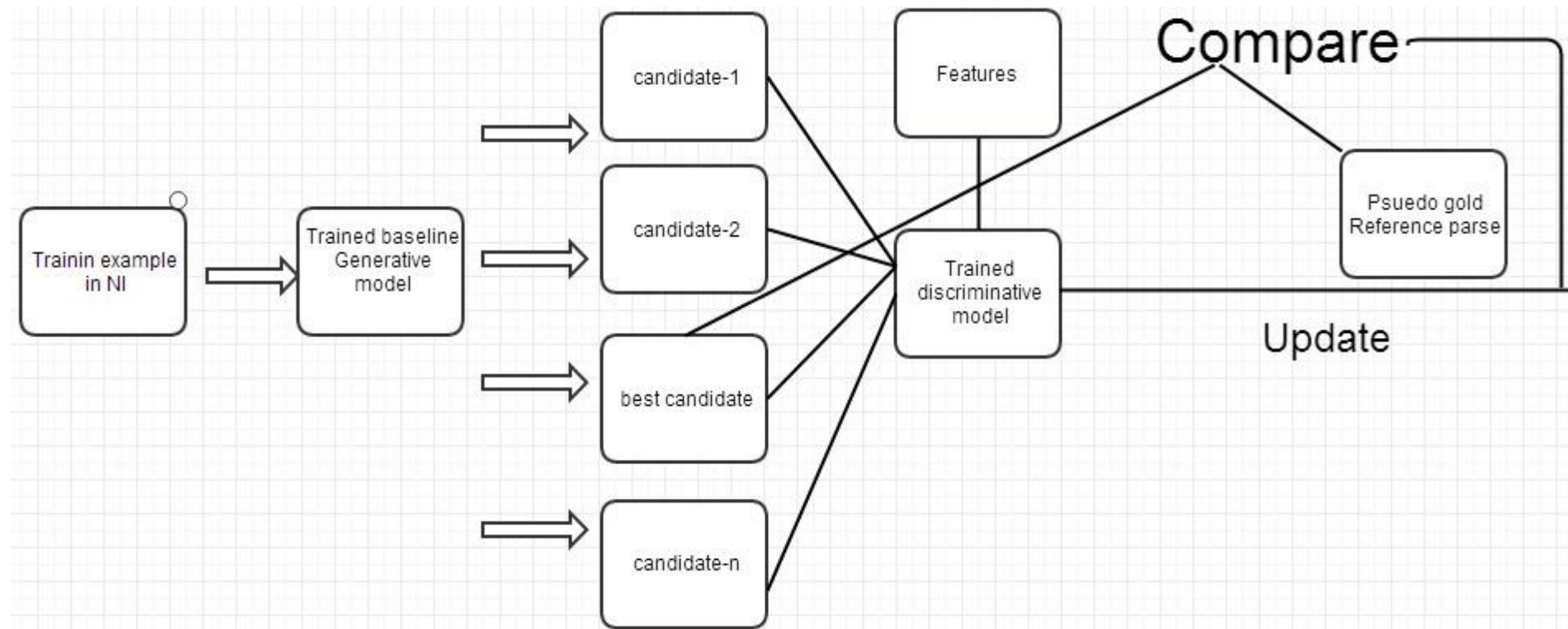


How does it work ??

- Generative model uses the unsupervised PCFG induction on the ideas of borchinger. The approach generates grammar-formulation rules which construct CFG productions that form a grammar that effectively maps NL sentences to formal meaning representations (MRs) encoded in its nonterminals.
- Now while training, for each sentence in Natural language, the generative model is first trained and then produces the set of candidate outputs for it.
- Next, a second conditional model is trained which uses global features to rescore these candidates.
- Generally for reranking averaged perceptron is used but authors used a variant of this algorithm suited to the task of navigation.

Discriminative Reranking

- Discriminative model parameter is updated with comparison between the best predicated candidate and the gold standard



Updating Weights

Weight Update with Single Parse

- The gold standard parse for each training example is not available .
- Pick a pseudo-gold parse out of all candidates. Choose the candidate with maximum probability of getting to target.

Weight Update with Multiple Parses

- Other than pseudo gold parses can also be useful for reranking
- Update with feature difference is weighted with difference between execution rates

Reranking Features

- For reranking the parses generated from baseline generative model , certain features are used by the discriminative model like:
- 1.Base features
- 2.Predicate-only Features
- 3.Descend Action features.

Results

	Parse Accuracy	Execution accuracy	Execution accuracy
	F-1	Single	Paragraph
Baseline	74.81	57.22	20.17
Gold -standard	78.26	52.57	19.33
Responses based	73.32	59.65	22.62

Results

- 1.Paragraph level accuracies is low compared to sentence level .
- 2.The execution accuracy of response based feedback is greater than gold-standard based ,for parsing accuracy situation is reversed.But the response based feedback outperforms it on final task.
- 3.Using multiple reference parse trees gives best results in all task.(like execution)
- 4.Also they showed that descend action features are most effective for weight updation but taking combination of all features gives best results.

Thanks
Questions ?