

Disambiguating Natural Language via Aligning Meaningful Descriptions

Final Dissertation Defense

Yida Xin

27 June 2022

Acknowledgement

This work is made possible by Prof.
Peter Chin, Prof. Steve Homer, Prof. Derry Wijaya, Dr.
Henry Lieberman, Pedro Colon-Hernandez, my
BULISP colleagues, and my MIT Genesis colleagues.



What can I help you with?



Language + Commonsense

- Ambiguities are the core difficulty for computers ...
- ... because computers lack commonsense
- Ambiguities accumulate from sentences to discourses
- Ambiguities should be resolved in transparent ways

Commonsense Disambiguation Hypothesis

- Disambiguation is the application of commonsense inference to language understanding.
- Disambiguation at all levels would solve most, if not all, of language understanding.

Research Strategy

- Simple disambiguation for sentences
- More sophisticated reasoning for sentences
- Simple disambiguation for discourses
- More sophisticated reasoning for discourses

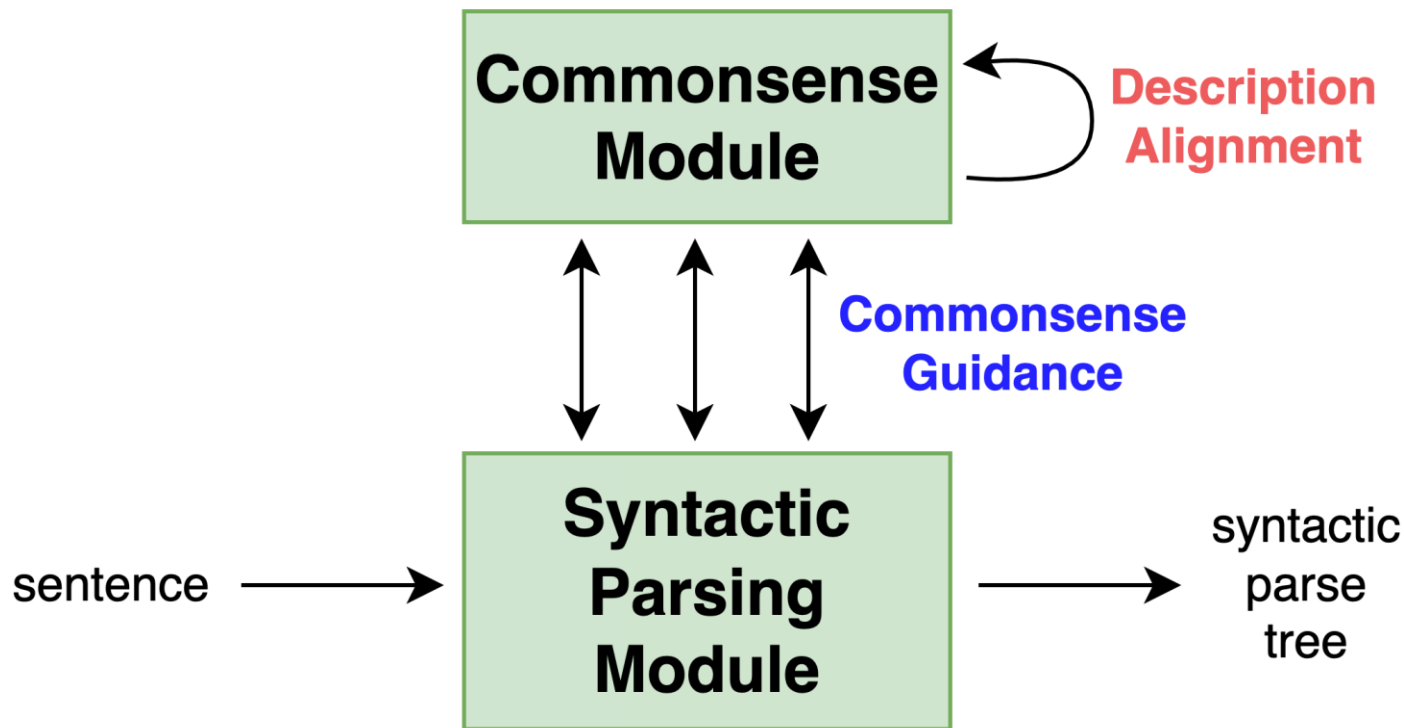
Early On I Discovered That...

- Many sentence-level ambiguities are context-independent
- Most discourse-level ambiguities are context-dependent

- O** ■ PatchComm: Sentences, Context-independent Disambiguation
- U** ■ Simple disambiguation, using ConceptNet
- T** ■ Sophisticated inferences, using RetroGAN-DRD
- L** ■ ProGeneXP: Implicit contexts -> Transparent descriptions
- I** ■ DialComm: Discourses, Context-dependent Disambiguation
- N** ■ Simple disambiguation, via alignment
- E** ■ Sophisticated inferences, using LM-GAN

- O** ▪ **PatchComm: Sentences, Context-independent Disambiguation**
- U** ▪ **Simple disambiguation, using ConceptNet**
- T** ▪ **Sophisticated inferences, using RetroGAN-DRD**
- L** ▪ **ProGeneXP: Implicit contexts -> Transparent descriptions**
- I** ▪ **DialComm: Discourses, Context-dependent Disambiguation**
- N** ▪ **Simple disambiguation, via alignment**
- E** ▪ **Sophisticated inferences, using LM-GAN**

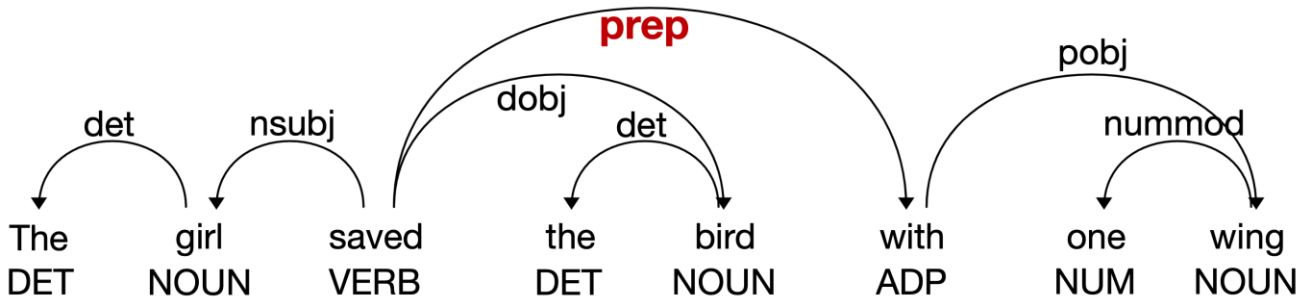
PatchComm: Context Indep Disambiguation



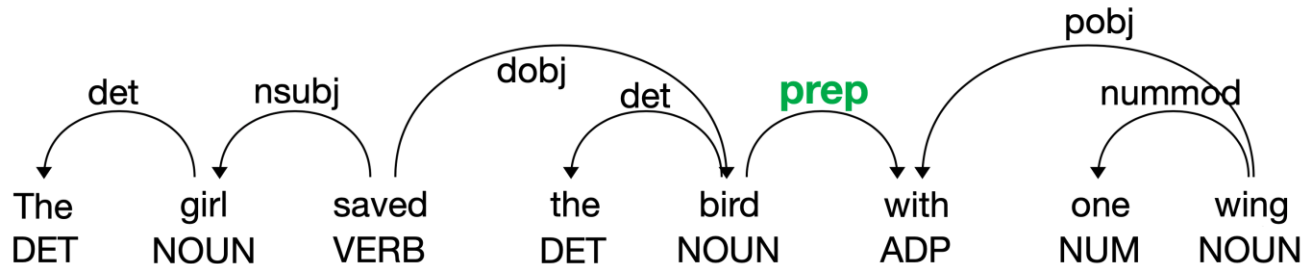
- Prepositional Phrase Attachment Ambiguity
- Pronoun Coreference Ambiguity

PatchComm Makes More Common Sense

spaCy

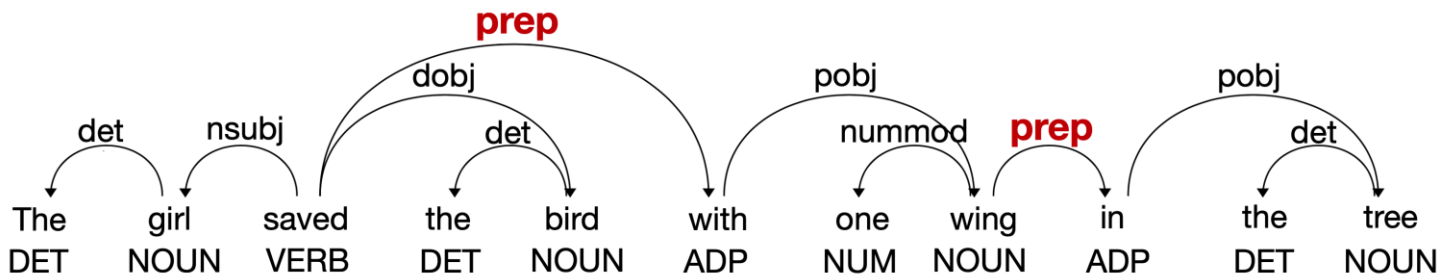


PatchComm

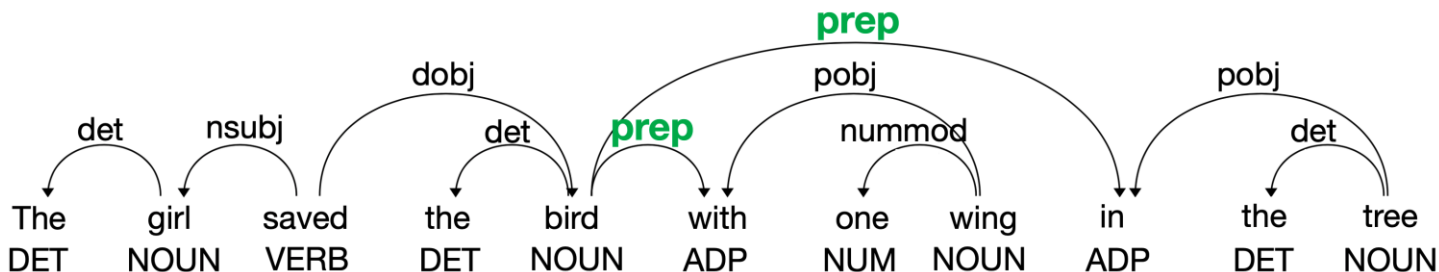


Disambiguate Multiple Prepositional Phrases

spaCy



PatchComm



Testing PatchComm with ConceptNet

Prepositional Phrase Attachment

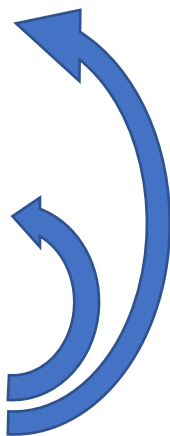
- Self-created dataset of 100 prepositional phrase sentences.
 - “The journalists **reported** the profits in the **newspapers**”
 - “The journalists reported the **profits** in the **commodities**”

	spaCy	spaCy + PatchComm w/ ConceptNet
Self-created	57.0%	67.0%

PatchComm: Pronoun Coreference

Query ConceptNet to Align Descriptions

The **knife** [ENT₁]
cuts through
the **butter** [ENT₂]
because **it** [PRON]
is **soft** [DESC]



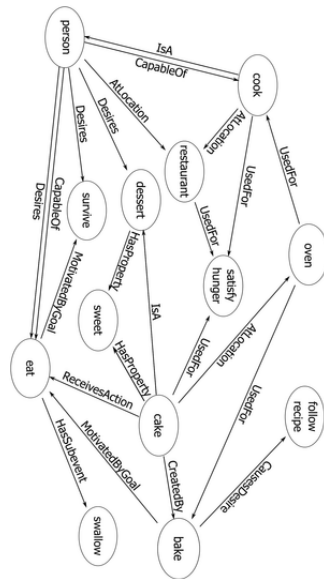
knife



butter



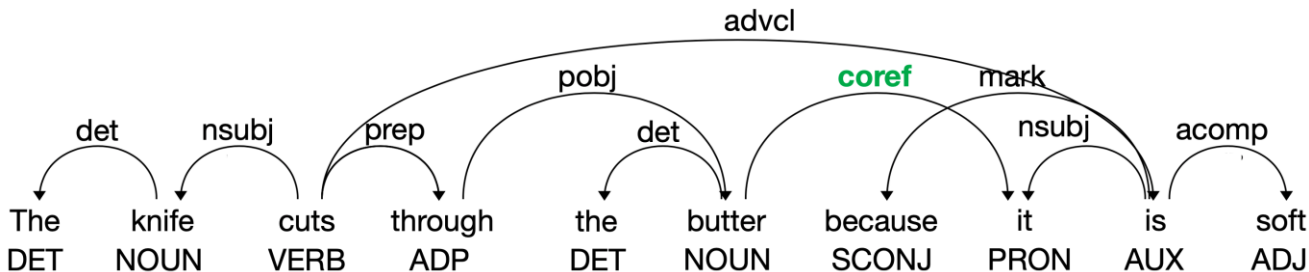
soft



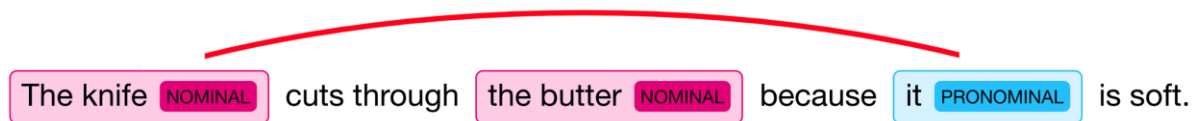
en butter — HasProperty → **en** soft

PatchComm Makes More Common Sense

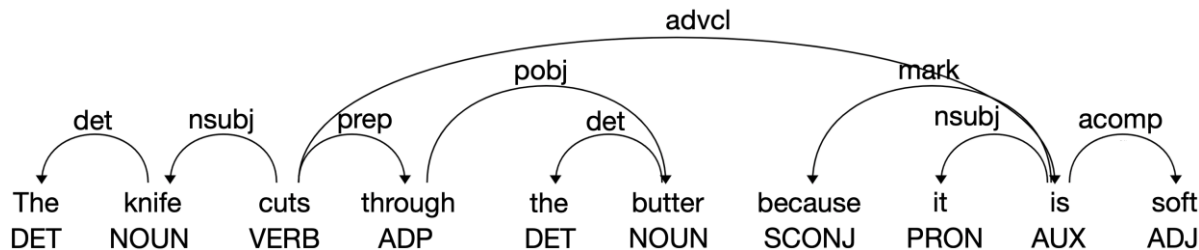
PatchComm



NeuralCoref



spaCy



Testing PatchComm with ConceptNet

Pronoun Coreference

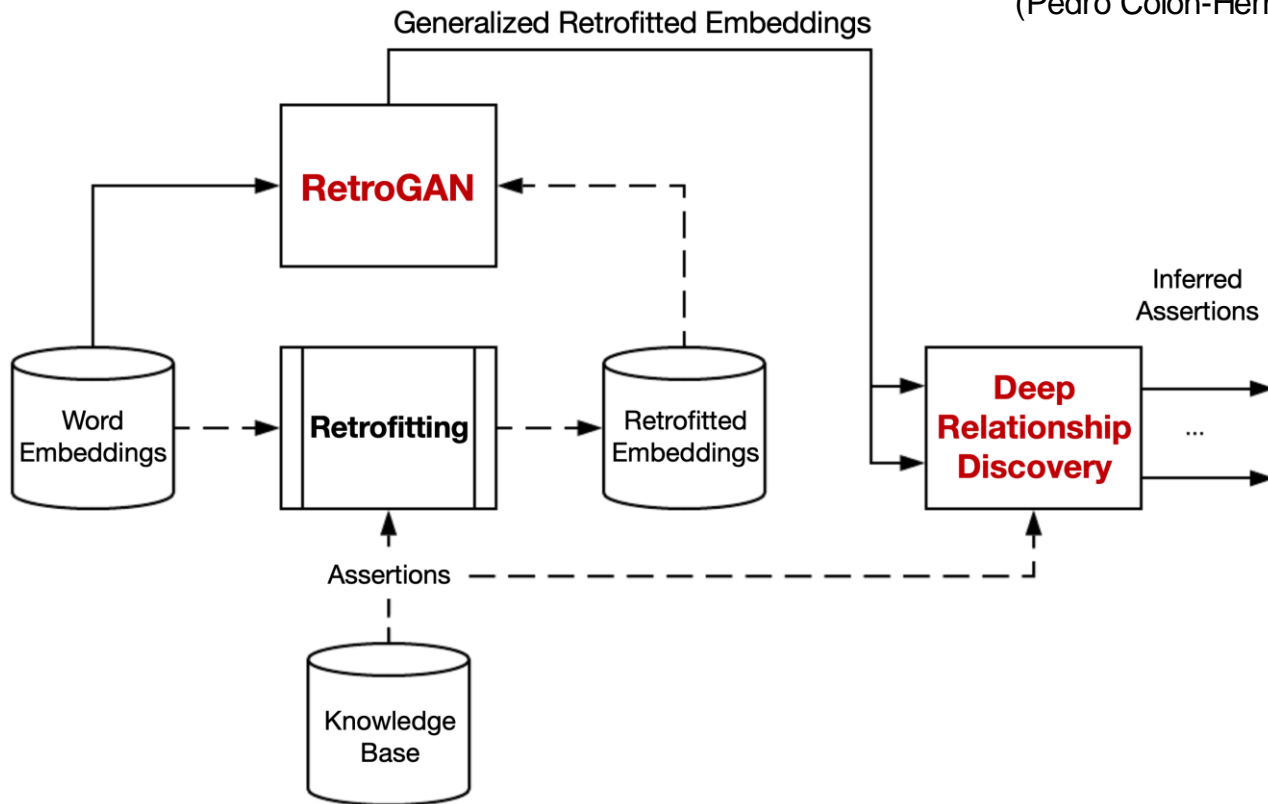
- Winograd Schema Challenge (WSC273) dataset.
 - “The **trophy** doesn't fit in the brown suitcase because **it**'s too **big**.”
 - “The trophy doesn't fit in the brown **suitcase** because **it**'s too **small**.”

	NeuralCoref	NeuralCoref + PatchComm w/ ConceptNet
WSC273	31.1%	39.9%

- O** ▪ **PatchComm: Sentences, Context-independent Disambiguation**
- U** ▪ Simple disambiguation, using ConceptNet
- T** ▪ **Sophisticated inferences, using RetroGAN-DRD**
- L** ▪ ProGeneXP: Implicit contexts -> Transparent descriptions
- I** ▪ DialComm: Discourses, Context-dependent Disambiguation
- N** ▪ Simple disambiguation, via alignment
- E** ▪ Sophisticated inferences, using LM-GAN

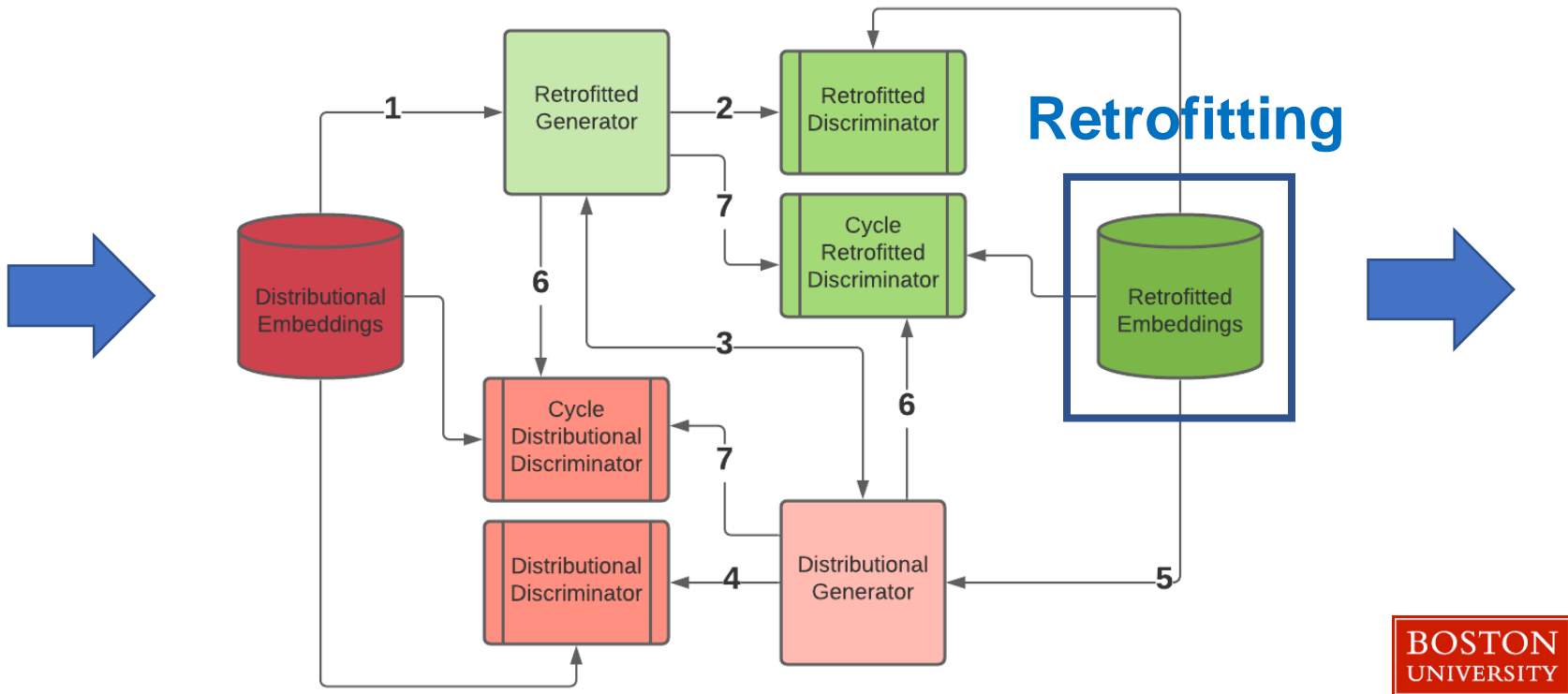
RetroGAN-DRD: Improving the Inferences

(Pedro Colon-Hernandez et al.)



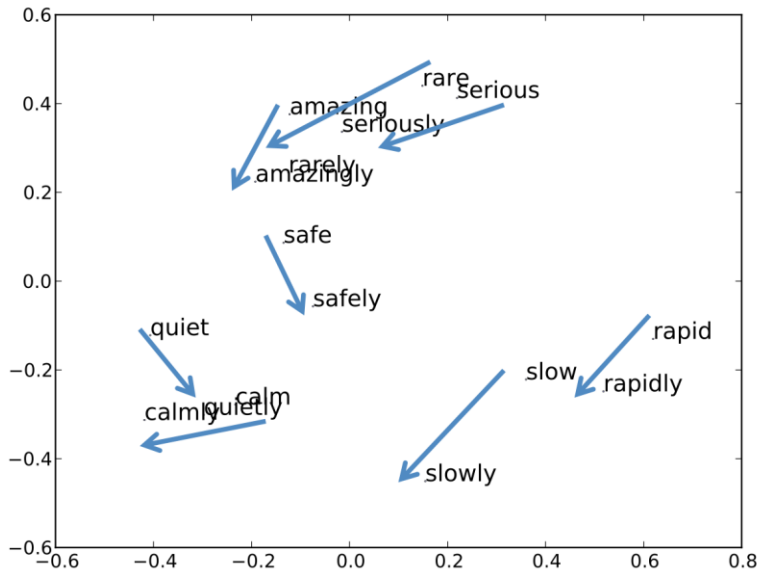
Architecture of RetroGAN

(Collaboration with Pedro Colon-Hernandez et al.)

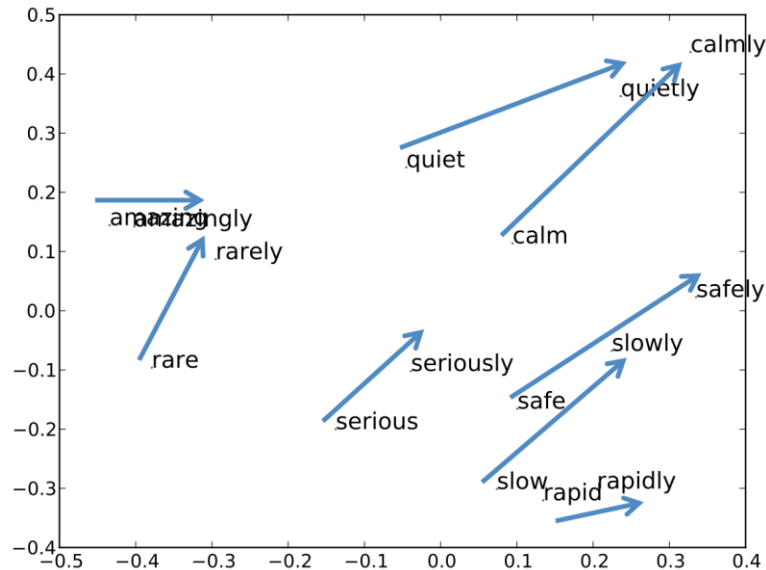


Retrofitting Distributional Word Embeddings with Knowledge Embeddings

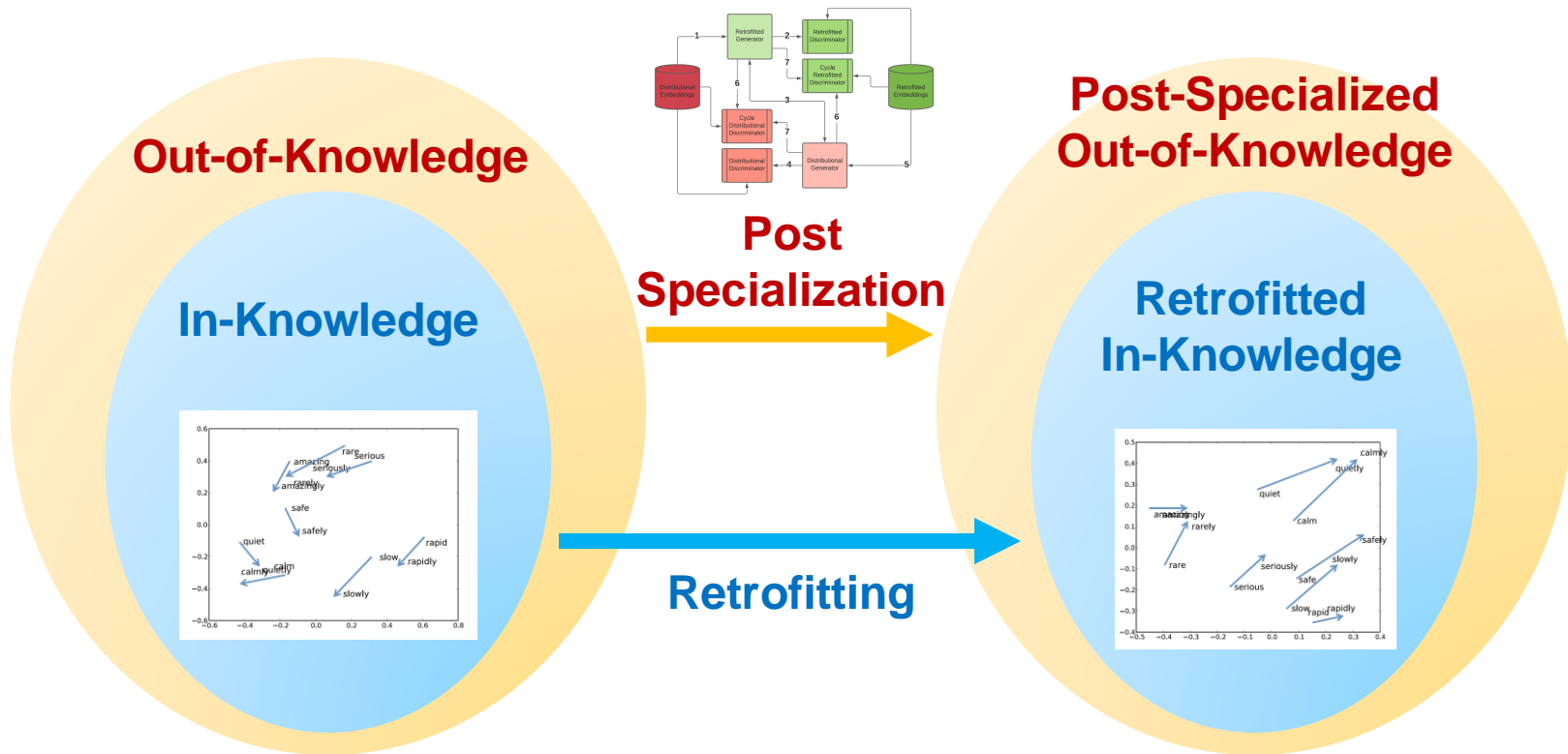
Distributional



Distributional + Knowledge

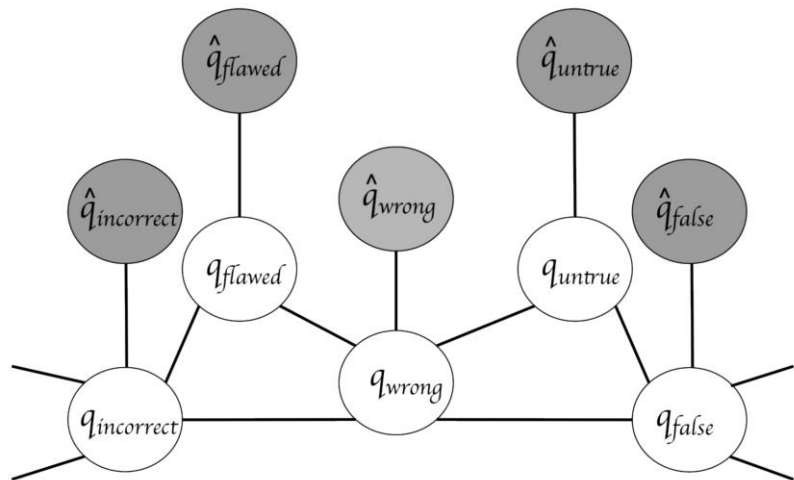


Post-Specialization for Out-of-Knowledge



RetroGAN: (1) Specialization

Attract (synonymy) only



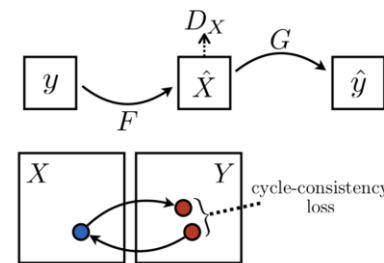
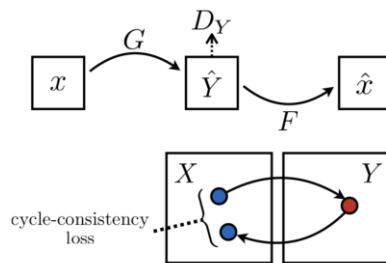
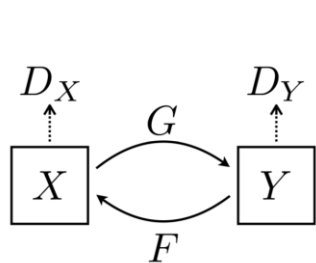
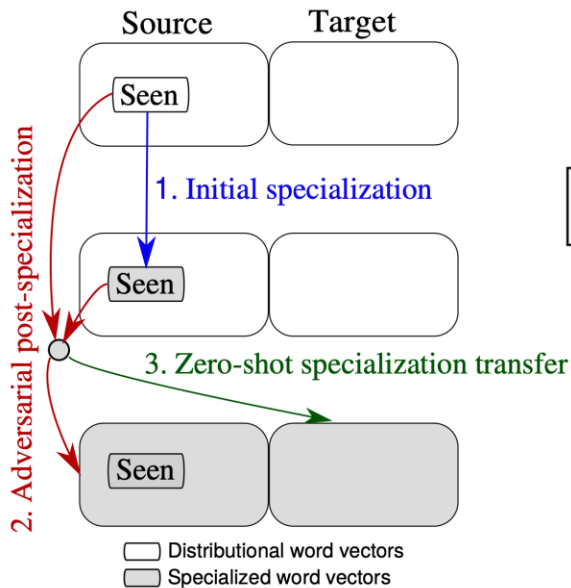
$$\Psi(Q) = \sum_{i=1}^n \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \beta_{ij} \|q_i - q_j\|^2 \right]$$

Attract (synonymy) & Repel (antonymy)

$$S(\mathcal{B}_S) = \sum_{(x_l, x_r) \in \mathcal{B}_S} \left[\tau (\delta_{syn} + \mathbf{x}_l \mathbf{t}_l - \mathbf{x}_l \mathbf{x}_r) + \tau (\delta_{syn} + \mathbf{x}_r \mathbf{t}_r - \mathbf{x}_l \mathbf{x}_r) \right]$$

$$A(\mathcal{B}_A) = \sum_{(x_l, x_r) \in \mathcal{B}_A} \left[\tau (\delta_{ant} + \mathbf{x}_l \mathbf{x}_r - \mathbf{x}_l \mathbf{t}_l) + \tau (\delta_{ant} + \mathbf{x}_l \mathbf{x}_r - \mathbf{x}_r \mathbf{t}_r) \right]$$

RetroGAN: (2) Post-Spec with CycleGAN

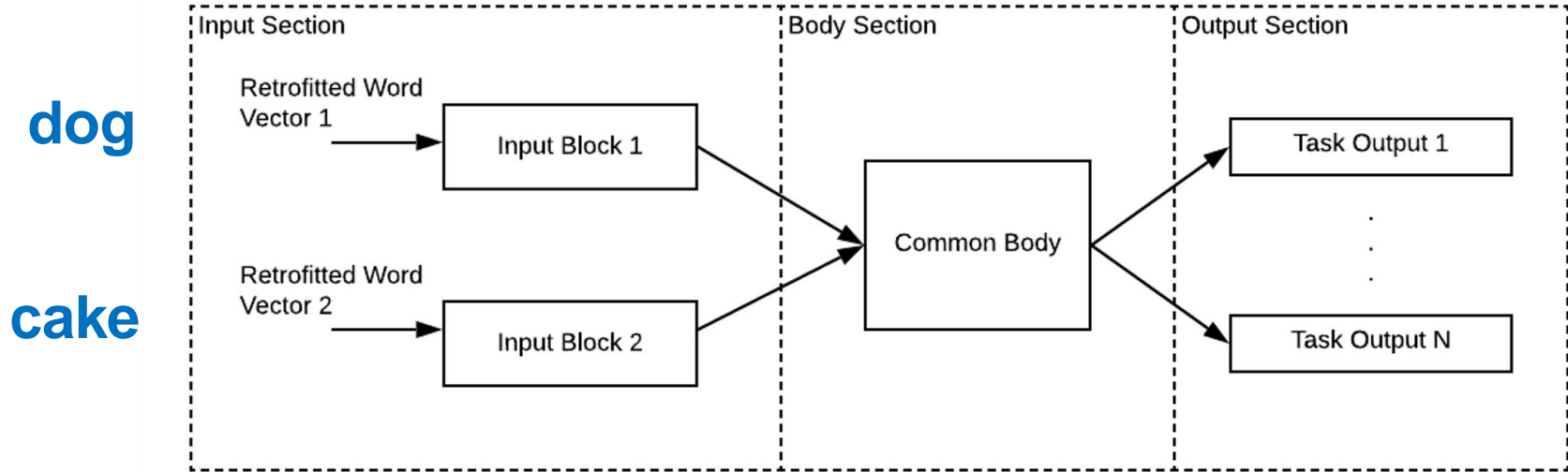


$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, X, Y) + \lambda L_{CYC}(G, F) + \gamma L_{ID}(G, F, X, Y) + L_{MM}(G, F, X, Y) + \varsigma L_{cCYC}(G, F, D_{cX}, D_{cY}, X, Y)$$

Testing RetroGAN for Out-of-Knowledge

	5%			10%			25%		
Models	SL	SV	C660	SL	SV	C660	SL	SV	C660
AuxGAN	0.615	0.510	0.453	0.667	0.569	0.470	0.679	0.581	0.475
RetroGAN	0.624	0.538	0.489	0.701	0.652	0.493	0.738	0.690	0.502
	50%			75%			100%		
Models	SL	SV	C660	SL	SV	C660	SL	SV	C660
AuxGAN	0.685	0.600	0.490	0.688	0.597	0.480	0.690	0.601	0.486
RetroGAN	0.755	0.716	0.511	0.763	0.721	0.507	0.762	0.715	0.509

Commonsense Inference via Deep Relationship Discovery



Commonsense Inference via Deep Relationship Discovery

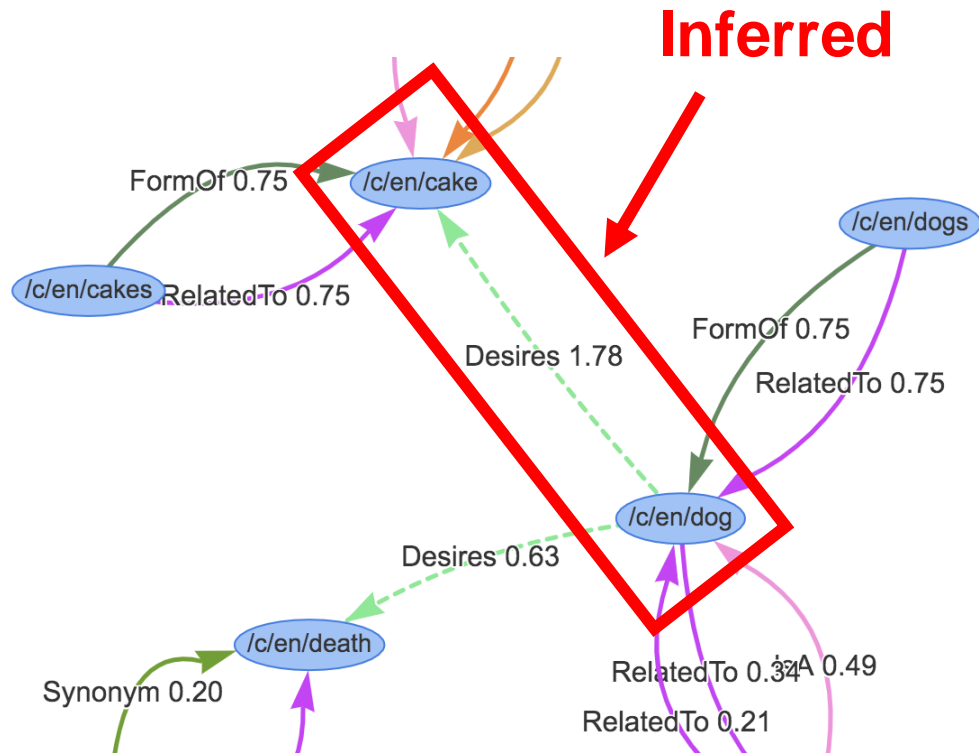
en a dog — Desires → en food

en cake — IsA → en food

???

~~en a dog — Desires → en cake~~

Boston University Department of Computer Science



Testing PatchComm with RetroGAN-DRD

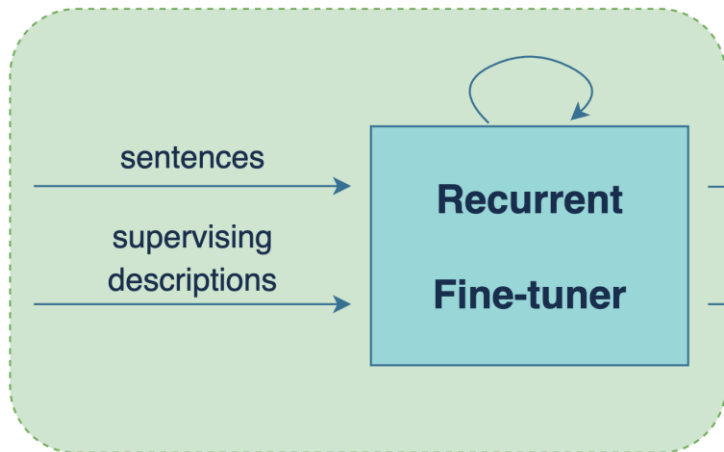
- Prepositional Phrase Attachment
 - “The journalists **reported** the **profits** in the **newspapers/commodities**”
- Pronoun Coreference
 - “The **trophy** doesn't fit in the brown **suitcase** because it's too **big/small**.”

	Baseline	Baseline + PatchComm w/ConceptNet	Baseline + PatchComm w/ RetroGAN-DRD
Self-created	57.0%	67.0%	80.0%
WSC273	31.1%	39.9%	67.4%

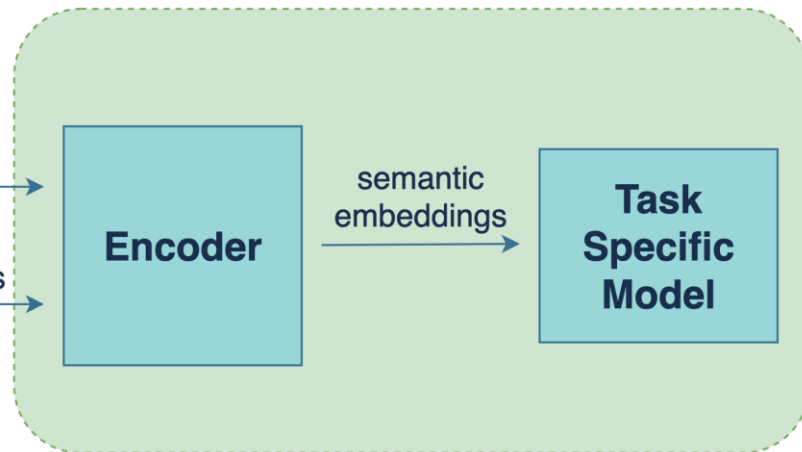
- O**
- PatchComm: Context Independent Disambiguation
- U**
- Simple disambiguation for sentences, using ConceptNet
 - Sophisticated reasoning for sentences, using RetroGAN-DRD
- T**
- L**
- **ProGeneXP: Implicit contexts -> Transparent descriptions**
- I**
- DialComm: Context Dependent Disambiguation
- N**
- Simple disambiguation for discourses, via alignment
 - Sophisticated reasoning for discourses, using LM-GAN
- E**

ProGeneXP: Generating Descriptions to Make Implicit Contexts Transparent

Stage 1: Recurrent Fine-tuning



Stage 2: Task Specialization



Recurrent Fine-tuner Targets Transparency

Initial
Super-
vision

<i>Sentence: The infection spread throughout the building because [it] was airborne.</i>	
Human	Airborne viruses can spread quickly in ventilated in-door areas.
Plain model	The infection spread throughout the building because it was airborne.
RF model	Airborne viruses can spread quickly in a building.
<i>Sentence: The infection spread throughout the building because [it] was ventilated.</i>	
Human	Airborne viruses can spread quickly in ventilated in-door areas.
Plain model	Infection spread because the building was not ventilated.
RF model	Ventilation can spread infection throughout a building.

Manual
Test

<i>Sentence: The smoke spread through the building because [it] was caught on fire.</i>	
Human	—
Plain model	Smoke spread through the building because it was on fire.
RF model	Smoke is created when a building is on fire
<i>Sentence: The smoke spread through the building because [it] came from the fire.</i>	
Human	—
Plain model	Smoke spread through the building because it came from the fire
RF model	Smoke from a fire can spread quickly in a building.

Preliminary Testing of ProGeneXP on Coreference

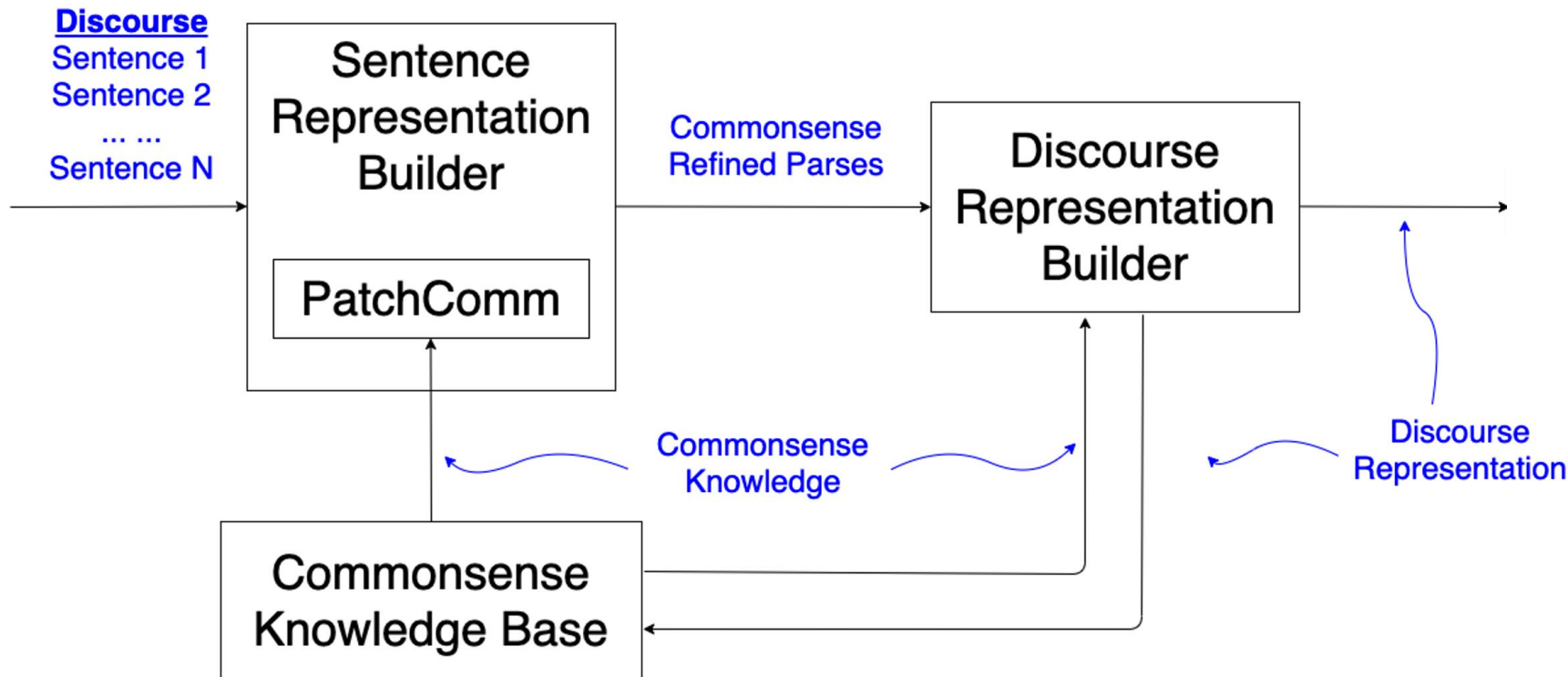
- Dataset: WinoGrande (WSC273-like sentences)

Train \ Test	BERT (cased)		BERT (uncased)		RoBERTa	
	–	with descs	–	with descs	–	with descs
WG-valid	0.867	0.859	0.863	0.852	0.855	0.856
WG-valid with descs	0.840	0.867	0.828	0.858	0.844	0.862

- Ongoing Work:
 - Using ProGeneXP to find better descriptions for PatchComm

- O** ■ PatchComm: Context Independent Disambiguation
- U** ■ Simple disambiguation for sentences, using ConceptNet
- T** ■ Sophisticated reasoning for sentences, using RetroGAN-DRD
- L** ■ ProGeneXP: Implicit contexts -> Transparent descriptions
- I** ■ **DialComm: Context Dependent Disambiguation**
- N** ■ **Simple disambiguation for discourses, via alignment**
- E** ■ Sophisticated reasoning for discourses, using LM-GAN

DialComm: Context Dep Disambiguation

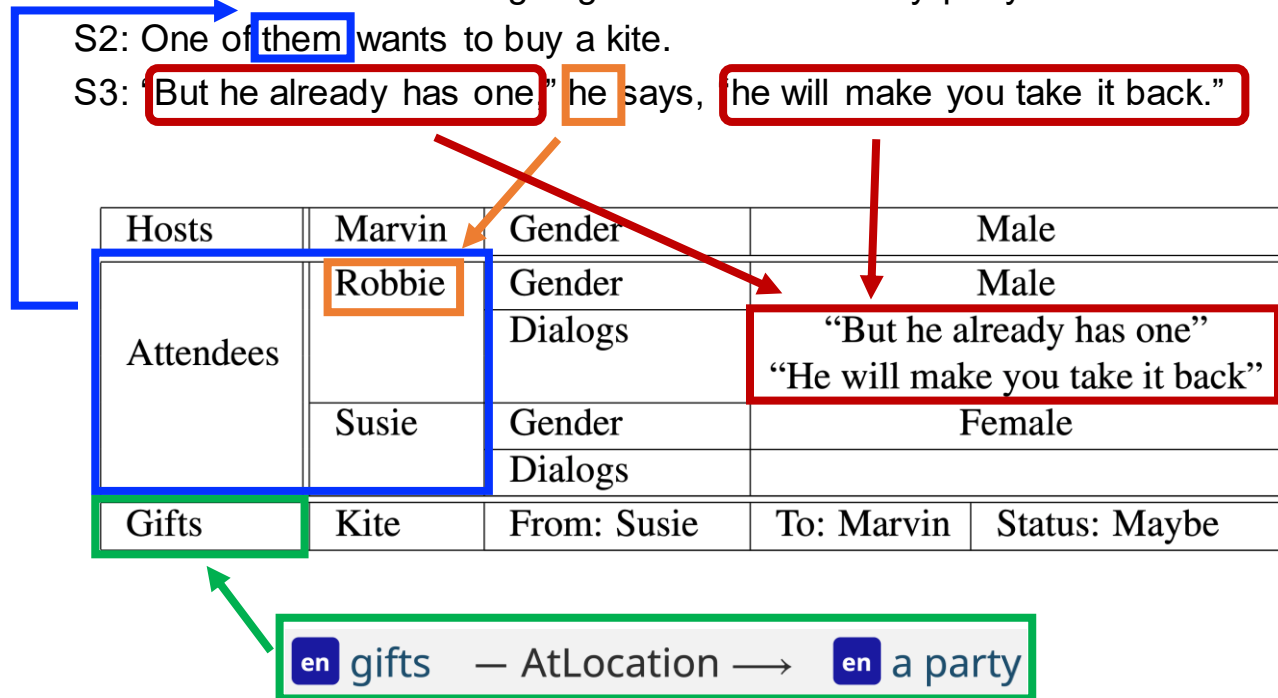


Aligning Descriptions into Frame

S1: Robbie and Susie are going to Marvin's birthday party.

S2: One of them wants to buy a kite.

S3: "But he already has one" he says, "he will make you take it back."



DialComm for Interactive End-User Programming in Natural Language

Boston University Department of Computer Science

74 Metafor Interactive Natural Language Programming (hugo@media.mit.edu)

File

```
[user] (01:28:32) Pacman is a character who loves to run through a maze and eat dots.
[agent] (01:28:32) Ok. I created a new agent Pacman that is a kind of character agent. I added the ability for Pacman to run, which can be through a maze. I added the ability for Pacman to eat. A dot is something which can be eaten.
[user] (01:28:53) Whenever Pacman eats a dot, it disappears and he wins a point.
[agent] (01:28:53) Ok. Now we are going to think about Pacman eating... When Pacman eats a dot, the dot disappears and Pacman wins a point. ...Now that we are done thinking about Pacman eating, we are going to resume thinking about __main__.
```

When Pacman is running through the maze, if a ghost catches him, then he loses and the game is over.|

```
DEICTIC STACK: [('Pacman', ['he', 'him', 'his', 'her', 'she', 'hers', 'who']), ('character', ['it']), ('maze', ['it']), ('dots', ['they', 'them']), ('dot', ['it']), ('dot', ['it']), ('Pacman', ['he', 'him', 'his', 'her', 'she', 'hers', 'who']), ('dot', ['it']), ('point', ['it'])]

DIR: ['__main__.Pacman', '__main__.dot']

CODETREE: ['__main__', 'FunctionT

def __main__():
    class Pacman(character):
        def run(maze):
            pass

        def eat(dot):
            dot.disappear()
            Pacman.win(point)

        def win(point):
            pass

    class dot:
        def disappear():
            pass
```

I found a bar with a bartender who was making fancy drinks.

USER

DialComm

```
class Bar:
    def __init__(self):
        self.bartender = Bartender()
        self.properties = []

class Bartender:
    def __init__(self):
        self.properties = []

    def make(self, drink):
        pass

class Drink:
    def __init__(self):
        self.properties = ['fancy']
```

I found a bar with a bartender who was making fancy drinks. If a drink was in the menu, the bartender would make it.

```
class Bar:
    def __init__(self):
        self.bartender = Bartender()
        self.properties = []

class Bartender:
    def __init__(self):
        self.properties = []

    def make(self, drink):
        if drink in menu:
            pass

class Drink:
    def __init__(self):
        self.properties = ['fancy']

class Menu:
    def __init__(self):
        self.properties = []
```

I found a bar with a bartender who was making fancy drinks. If a drink was in the menu, the bartender would make it. One customer was rude, so the bartender threw away their drink.

```
class Bar:
    def __init__(self):
        self.bartender = Bartender()
        self.properties = []

class Bartender:
    def __init__(self):
        self.properties = []

    def make(self, drink):
        if drink in menu:
            pass

    def throw(self, drink):
        pass

class Drink:
    def __init__(self):
        self.properties = ['fancy']

class Menu:
    def __init__(self):
        self.properties = []

class Customer:
    def __init__(self):
        self.properties = ['rude']
```

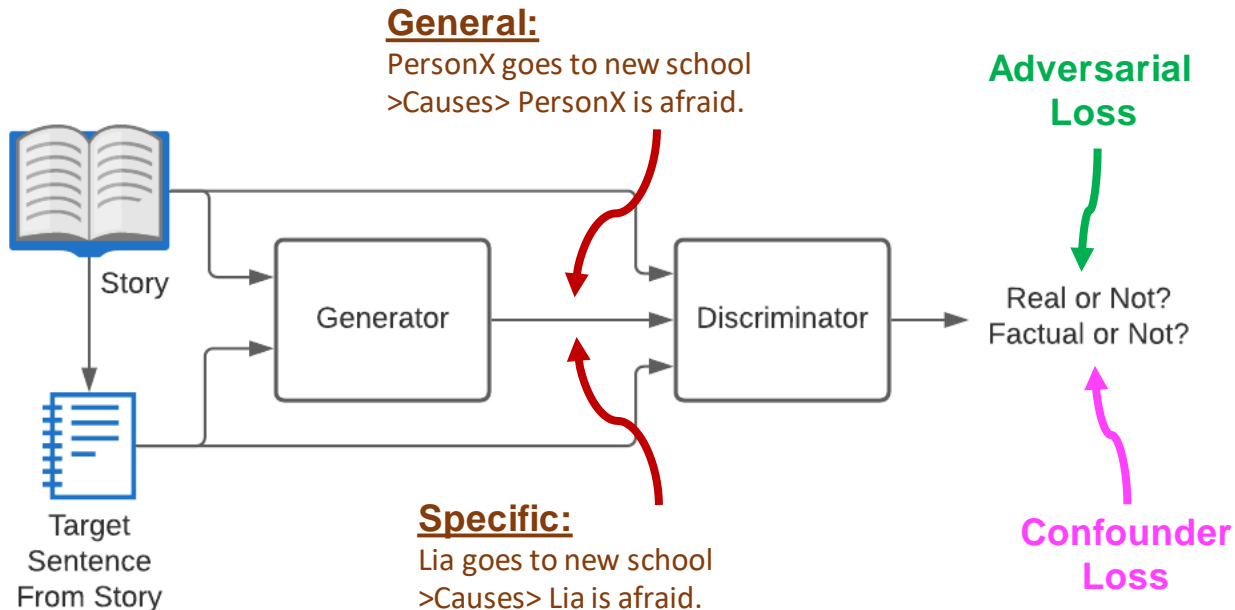
- O** ▪ PatchComm: Context Independent Disambiguation
- U** ▪ Simple disambiguation for sentences, using ConceptNet
- T** ▪ Sophisticated reasoning for sentences, using RetroGAN-DRD
- L** ▪ ProGeneXP: Implicit contexts -> Transparent descriptions
- I** ▪ **DialComm: Context Dependent Disambiguation**
- N** ▪ Simple disambiguation for discourses, via alignment
- E** ▪ **Sophisticated reasoning for discourses, using LM-GAN**

LM-GAN: Context-Dependent Inference via Hinting + Joint Inference + Adversarial Training

(Collaboration with Pedro and friends.)

Lia was starting high school at a new school.
She was afraid. She thought nobody would like her. But all the students were kind to her. Lia made many friends on her first day!

She was afraid.



Striking a balance between Recall and Precision

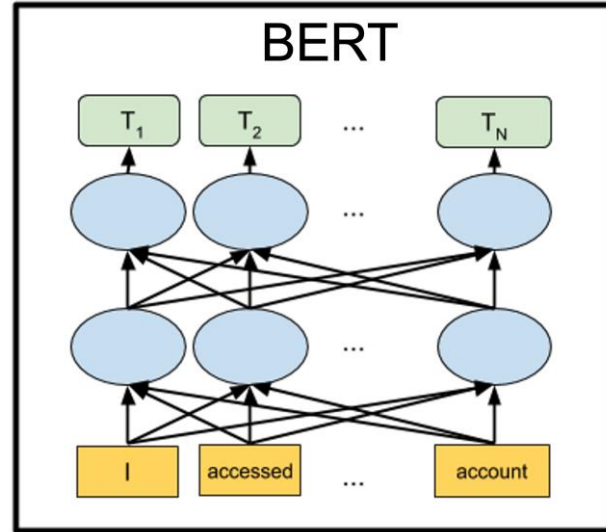
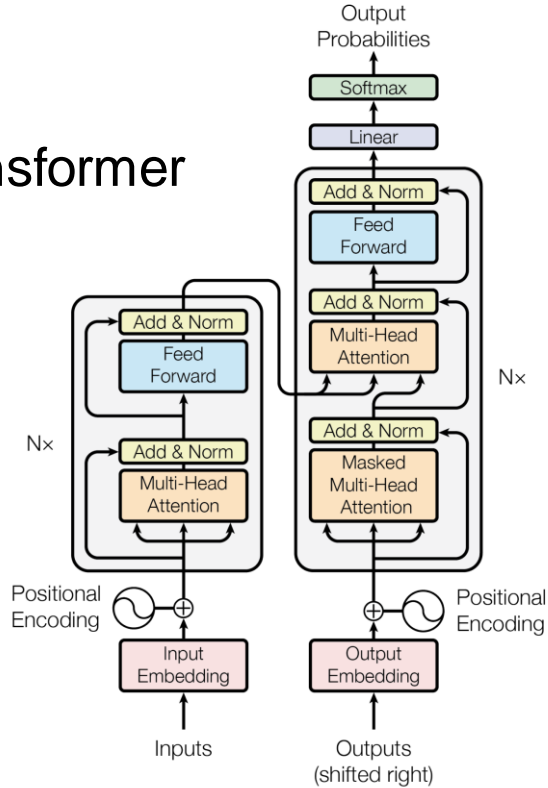
Recall

Precision

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L-SUM	BLEU	METEOR
+ ADV + CONF	43.656	10.544	40.380	40.379	31.335	61.683
+ ADV - CONF	43.747	10.559	40.530	40.531	31.279	61.623
- ADV + CONF	43.715	10.680	40.292	40.292	31.470	61.776

Honorable Mention...

Transformer



Contributions

- Tested PatchComm on disambiguating sentences
- Showcased ProGeneXP on bringing context into sentences
- Implemented DialComm to disambiguate discourses and enabled end-user programming in natural language
- Incorporated commonsense into language understanding
- Set the stage for further advances in NLU and Commonsense